OSLO

Optics Software for Layout and Optimization

Optics Reference

Version 6.1

Lambda Research Corporation
80 Taylor Street
P.O. Box 1400
Littleton, MA 01460

Tel: 978-486-0766
Fax: 978-486-0755

support@lambdares.com

COPYRIGHT

TRADEMARKS

Table of Contents

Chapter 1 Quick Start

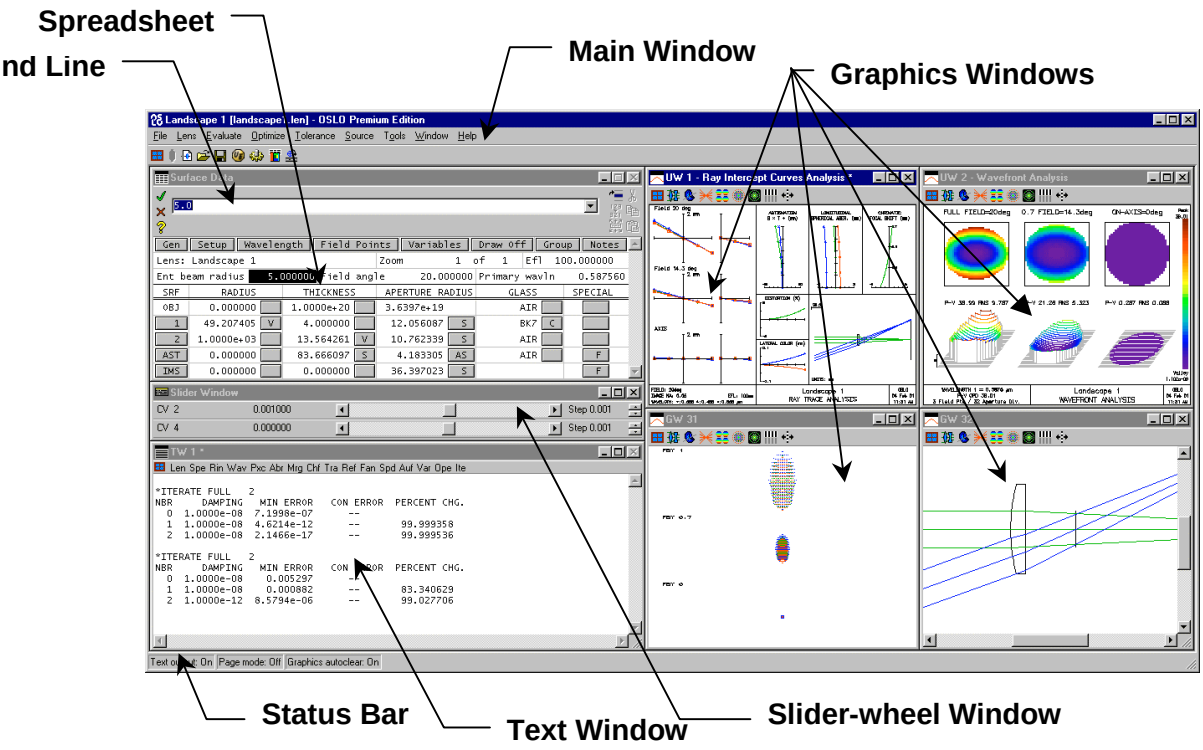
OSLO provides a computing environment for optical design. In addition to the usual functions that provide optimization and evaluation of optical systems, OSLO features a special windows interface that allows you to work interactively to probe the details of your optical system during the design process. OSLO accepts symbolic or numerical input using menus, toolbars, or commands; slider-wheel functions for real-time analysis; and automatic generation of dialog boxes and menus for custom program enhancements.

OSLO works similarly to other windows programs. If you are familiar with other windows software, you will be able to use OSLO without difficulty. However, the OSLO user interface does contain several unique features that make it an efficient and easy-to-use optical design program, and you can enhance your productivity by reading through this chapter.

The screen shot below shows a typical configuration of OSLO. Generally, you enter data either in a spreadsheet or in the command line. You can enter commands either directly in the command line or by clicking a menu or toolbar button. Commands and menus are completely integrated; there are no separate command and graphical input modes. Output from OSLO typically appears in a text or graphics window, according to the type of output to be displayed.

A unique feature of OSLO is its slider-wheel window, holding up to 32 graphical sliders, providing callbacks to default or user-supplied routines that perform evaluation or even full optimization iterations when a slider is moved.

Other special windows are provided for database functions, file handling, and text editing. A substantial portion of OSLO is written in CCL, a modern byte-code language similar to Java. Incremental compilers and a linker are seamlessly integrated with the program to provide byte-code efficiency with the ease of use of an interactive language.

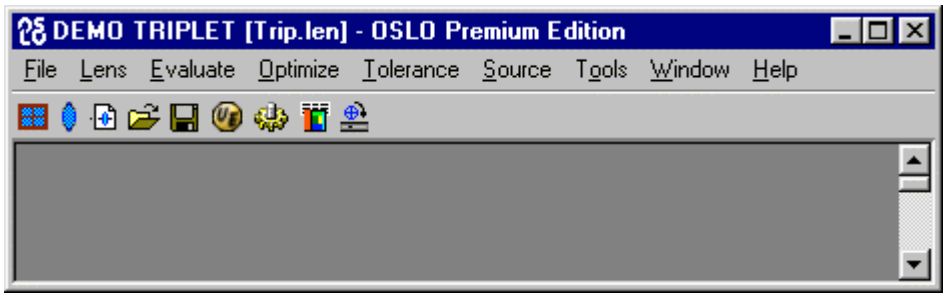


Main window

The main window is the central control point for most OSLO tasks. It contains the title bar, menu bar, tool bar, and a large work area that contains other OSLO windows. Usually, the main window should be maximized to make use of the full display area, allowing as much room for the other OSLO windows as possible.

Title bar

The **title bar** includes the Windows system button (displaying the OSLO icon), the ID of the current lens, the name of the current lens file (in brackets), and the name of the edition of OSLO that is running. At the right end of the main window title bar are minimize, maximize, and close buttons, which perform their usual functions for the main window.



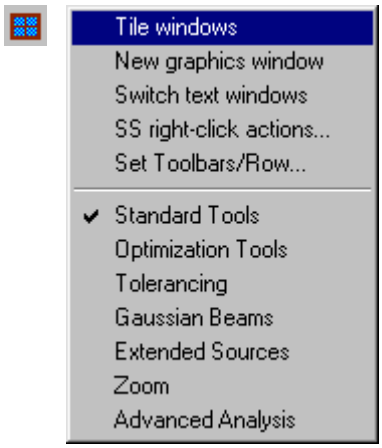
Menu bar

The **menu bar** displays a list of top-level items for the OSLO menu system. Each item contains a pull-down menu with several options. Some options contain pull-right menus with additional choices. The OSLO menu works like other Windows menus, but it can easily be enhanced or modified to include user-supplied commands. The menu that is displayed when OSLO is run is determined by the contents of a user-modifiable file called **a_menu.ccl**. The standard menu file provides the following options:

- **File** — Commands that open and save lens files, access lens and other databases, manage printing and plotting, and set program preferences. The last option on the File menu is Exit, which stops the program.
- **Lens** — Commands that enter or display lens and system data. These include access to OSLO spreadsheets, data output routines, and lens drawings.
- **Evaluate** — Commands that analyze or compute the performance of the system, such as aberration analysis, ray tracing, and image analysis.
- **Optimize** — Commands that set up and carry out optimization tasks.
- **Tolerance** — Commands that set up and carry out tolerancing tasks.
- **Source** — Commands that define or analyze the current system using special sources, such as extended or Gaussian beam sources.
- **Tools** — CCL compiler and special or demo commands supplied with the program.
- **Window** — Options to open, close, and update windows. Window also includes commands to copy text and graphical output to a file or the Windows clipboard.
- **Help** — Provides the main entry into the OSLO help system. You can also obtain context-sensitive help by clicking on the help button in a spreadsheet or dialog box.

Main tool bar

Although OSLO has only one menu bar, it has several tool bars. The Standard tools in the **main tool bar** include buttons that invoke basic file and data handling commands. Supplemental tools are grouped according to application area (optimization, tolerancing, sources, etc.). Like the main menu, the tools in the main toolbar are determined by the contents of a user-modifiable file **a_toolbar.h**, which is included in the a_menu.ccl file.



The main toolbar begins with the Window Setup button. This button is found in all OSLO toolbars, and is divided in two parts. The top part shows window setup commands, while the bottom part shows various toolbars appropriate for the window. The *Tile windows* item, common to all Window setup buttons, is a user-modifiable command that attempts to lay out the various spreadsheet, graphics, and text windows in a convenient pattern. *New graphics window* and *Switch text windows* create graphics and text windows. The *SS right-click actions* item pops up a menu of editing commands that pertain to the selected rows in the current spreadsheet (if any). The SS right-click actions menu also appears when you right-click in a spreadsheet. Finally, the *Set Toolbars/Row* item allows you to choose how many toolbars will be concatenated before starting a new row.

The lens spreadsheet button opens the main surface data spreadsheet, which in turn contains several buttons that open subsidiary spreadsheets for special and supplemental data.

These tools are used to create a new lens, open an existing lens, or save the current lens.

These tools invoke the editor and compile the code in the private CCL directory. The editor may be either the built-in OSLO editor, or an external editor such as Ultra-Edit (shown).

This button opens the current CDB database spreadsheet, which contains its own menu button and buttons for various column editing functions, as well as user-defined buttons for database callbacks.

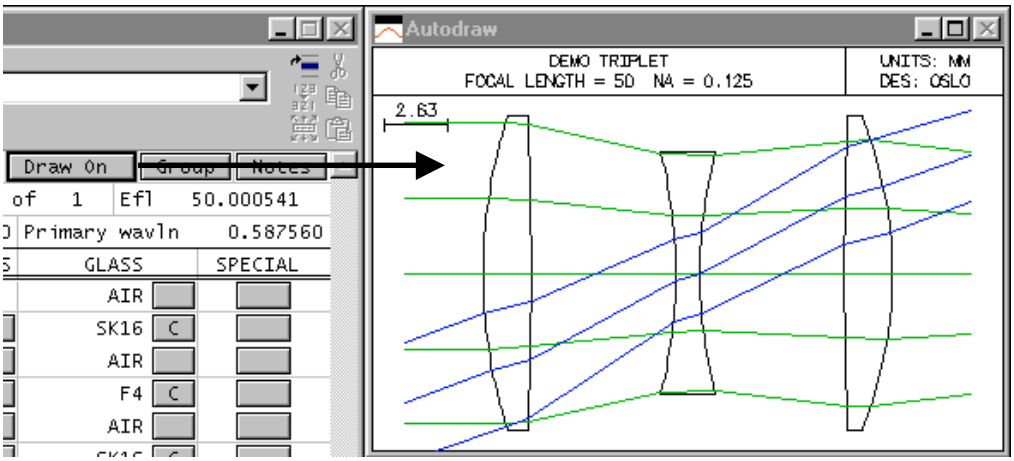
This button opens a spreadsheet used to set up a slider-wheel window. This window allows you to vary lens parameters during design or evaluation by dragging graphic sliders or rotating the mouse wheel.

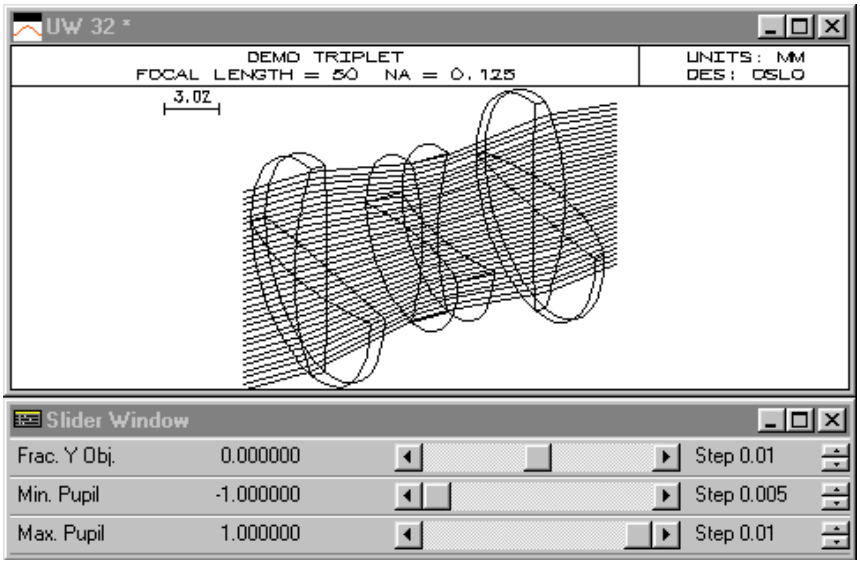
Graphics windows

OSLO's graphics windows help you see how your lens design performs. To avoid clutter, OSLO does not create a new window for each graphics plot. Instead, a new plot replaces the one in the current graphics window (marked with an asterisk). You make a window current by clicking in it. The Tile Windows command optimizes the layout of your windows by creating different tiling patterns, depending on which windows are active (minimized windows are ignored).

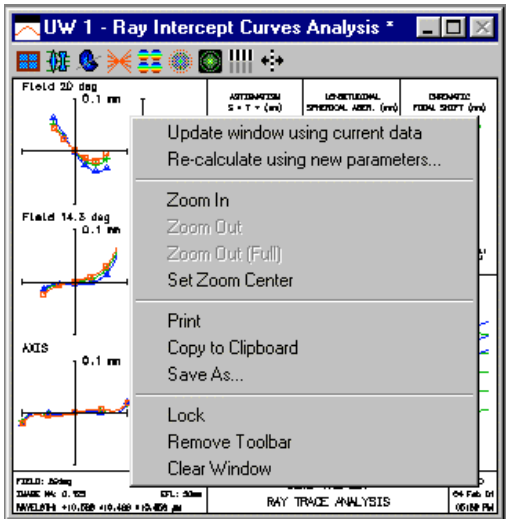
One graphics window is opened automatically when OSLO is started. To open or close additional graphics windows, click the Create Graphics Window item on the Setup Window menu (first toolbar button) or use the Open and Close options on the Window menu. Graphics windows are continuously resizable and zoomable, with content that resizes automatically. There are two types: *tow* (to window) windows have variable aspect ratio, and are used for most plots, while *iso* (isomorphic) windows are used for lens drawings, spot diagrams, etc. You can resize windows manually by dragging the window frame, or you can set window sizes and positions using CCL commands.

In addition to 32 standard graphics windows, OSLO has two special windows. The Autodraw window shows a plan view of the current lens, highlighting the surface currently selected in the lens spreadsheet. The Slider-wheel window holds up to 32 graphic sliders that allow you to change lens parameters interactively, while watching the effects (usually shown in graphics windows) in real time.





OSLO graphics windows are either static or updateable, indicated by GW or UW in the title bar. Updateable windows are very useful in developing a lens design. For example, you can put a plot in a graphics window using a standard tool bar icon. If you change the lens data, the plot will not change by itself. However, since the window is updateable, you can right-click in the window and select the "Update window using current data" item (or double-click in the window) to update the window. Also, you can re-plot a window using the same analysis but with different graphic scales by right-clicking and choosing the "Recalculate using new parameters" option. If you have several updateable graphics windows on the screen, you can update them all simultaneously using the Windows >> Graphics >> Update All command in the main menu.

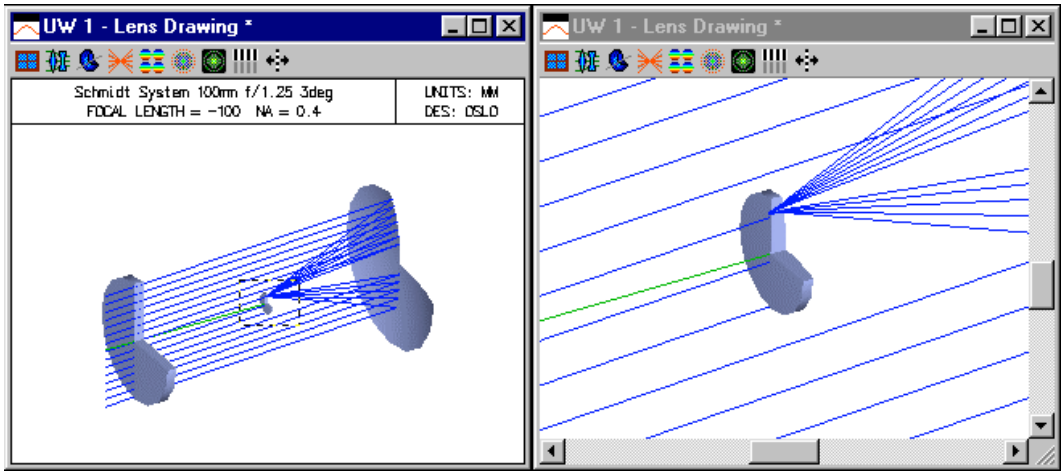


The current window can be copied to the clipboard by right clicking in the window and selecting the Copy to Clipboard item. Graphics output is placed in the clipboard in standard Windows metafile (vector) format.

You can save graphics windows to a file using the Save As command, which also pops up on the right-click menu. You can choose to save the window as a Windows metafile, bitmap, or hpgl file. Although metafile format is generally preferred for most line graphics output, some complex plots (notably spot diagrams) consume excessive file space and are better saved as bitmaps. In addition, shaded graphics plots must be saved as bitmap files.

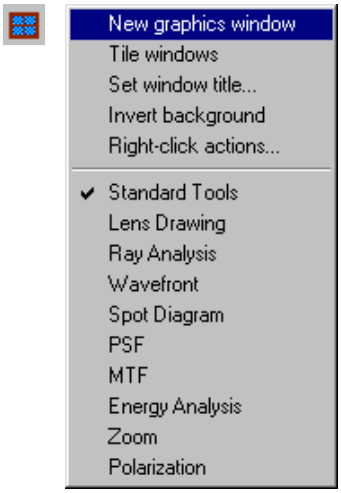
There are several ways to zoom an OSLO graphics window. Zooming is limited to 16x total magnification, but when combined with the maximize button, the total range is greater than this.

- You can zoom by holding down the left mouse button and dragging a marquee box around the desired zoom region.
- You can right click in the window and select a zoom option from the menu that pops up. You can then drag a scroll bar to pan.
- You can put your mouse cursor in the window and rotate the mouse wheel to zoom. If you place the mouse cursor over one of the scroll bars, the window will pan as you rotate the wheel.
- You can use CTRL+ARROW keys to zoom in or out, or SHIFT+ARROW keys to pan.



Graphics window tool bar

The Standard Tools for graphic windows are intended to provide general report-graphics plots. More detailed analyses can be produced using subsidiary toolbars selected from the Window Setup menu (1st toolbar button). The report graphics plots are made using CCL routines, so they can be modified to meet special requirements.



The Window Setup button in graphics windows contains commands for creating and managing additional graphics windows, as well as choosing toolbars. Only one toolbar is allowed in each graphics window; if you want several different toolbars simultaneously, you must create several windows to hold them. The *Set window title* item allows you to put an identifying phrase in the title bar of a graphics window. *Invert background* allows you to choose between black or white background colors, and *Right-click actions* produces the same menu as does right-clicking in the window.

The Standard Tools for graphics windows include basic lens drawing and optical evaluation features. The other items on the toolbar list provide expanded capabilities in particular areas.

The plan-view drawing button draws a 2D view of the current lens, with ray trajectories shown for the default rays according to the Lens drawing operating conditions. This drawing is made by neglecting any 3D information; it is not a section of a 3D drawing.

The shaded view drawing button draws a 3D view of the current lens, again using the Lens drawing operating conditions (Lens >> Lens Drawing Conditions). This drawing uses OpenGL graphics, instead of the normal OSLO vector graphics routines. To obtain hard copy, it is necessary to use a bitmap format (*.bmp, *.rle).

This button creates a Ray Analysis report graphic window, which is a single plot that contains several ray-based evaluations: Ray-intercept curves, Coddington astigmatism field curves, Longitudinal spherical aberration, chromatic focal shift, distortion, and lateral color. A plan view lens drawing is also shown in the lower right corner.

This button produces a wavefront report graphic, which shows false-color interferograms and sliced-contour representations of the image-space wavefront for each field point.

This button shows through-focus spot diagrams for each field point. Although this is a vector plot, it is preferable to produce hard copy in the form of a bitmap to save file space.

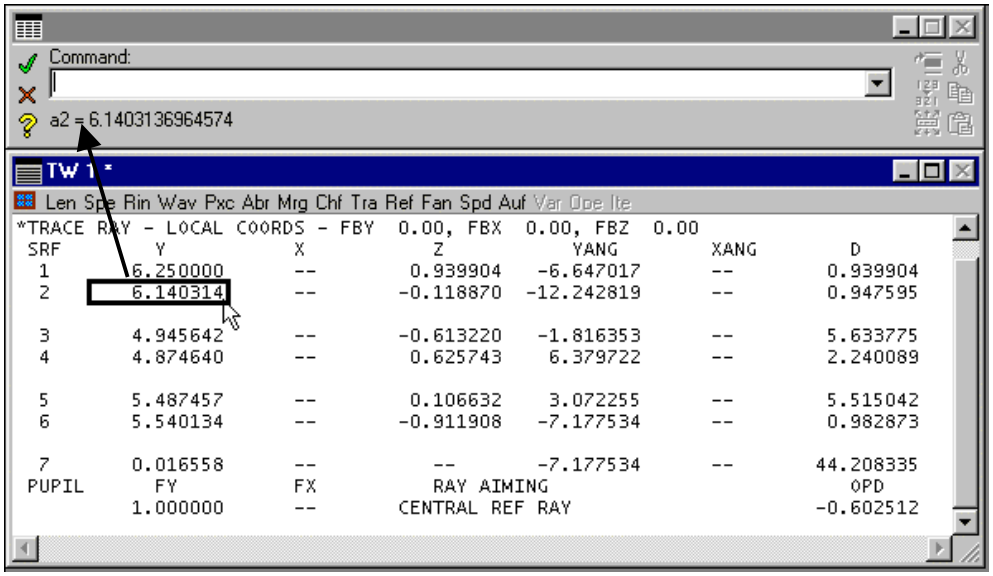
This report graphic shows (diffraction) point spread functions from the defined field points, together with x and y scans and encircled (and ensquared) energy distribution curves.

These buttons produce MTF vs frequency and MTF vs focus plots, respectively.

The Standard Tools for graphic windows are intended to provide general report-graphics plots. More detailed analyses can be produced using subsidiary toolbars selected from the Window Setup menu (1st toolbar button). The report graphics plots are made using CCL routines, so they can be modified to meet special requirements.

Text windows

Although much optical design is done using graphics plots, some aspects require (numerical) text output. OSLO text windows have a unique feature for these aspects: the Spreadsheet buffer. The Spreadsheet buffer is an array in memory that mirrors text output sent to the display. Each element in the Spreadsheet buffer can be accessed by clicking on the text window with the mouse, by typing its row-column designation in the command line, or by using a CCL function that retrieves its value for additional processing. Only real numbers are saved in the Spreadsheet buffer, but these numbers are saved with double precision.

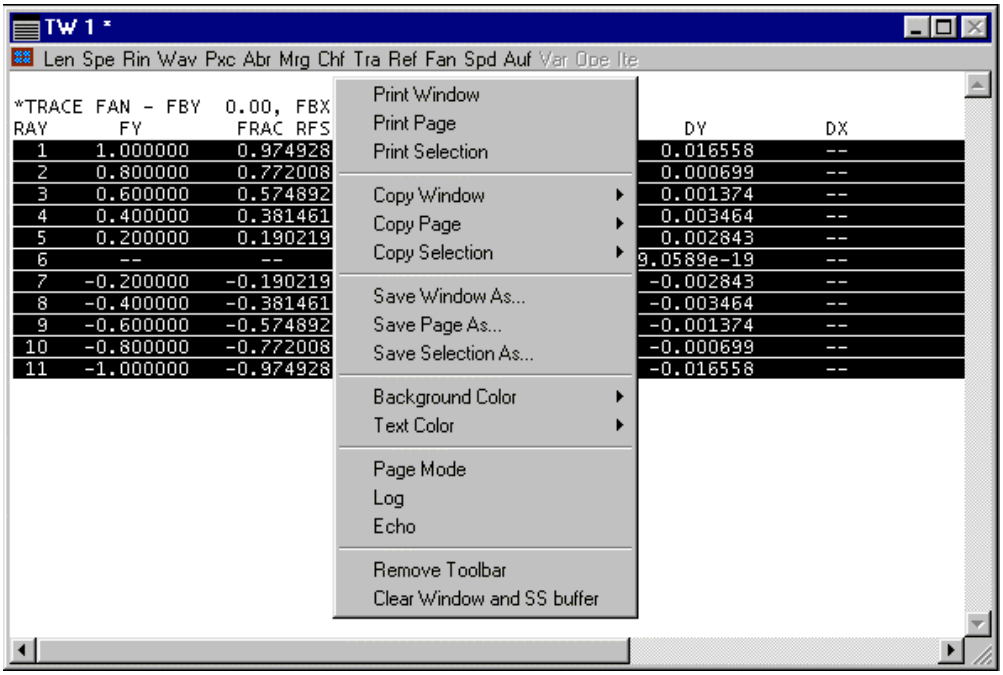


One text window is opened when the program starts. It serves as a serial record of a design session, allowing you to scroll backwards through up to 1999 lines of text output (this is also the size of the spreadsheet buffer). You can open a second text window using the Switch text window item from the Setup Window menu, or the Window >> Text >> Open menu item. Text output from the program will be directed to the window that has an asterisk on the title bar. Each text window has its own spreadsheet buffer.

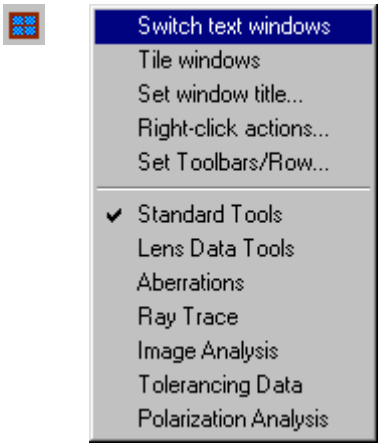
Two preferences have an important effect on the behavior of the text window: *page mode* and *output text*. If *page mode* is turned on, the first line of output from each command will appear at the top of the text window. If it is off, the last line of output from each command will appear at the bottom of the window.

If *output text* is on, output appears normally in the text window, but if it is off, output only goes to the spreadsheet buffer, and does not appear in the window. This feature is used frequently in OSLO to pass arrays to CCL routines. The settings of both *page mode* and *output text* appear in the OSLO status bar. If text windows do not operate as expected, it is a good idea to check the settings of these preferences.

You cannot type into an OSLO text window, but you can select a range of lines by holding down the left button and dragging the mouse. You can also select using the keyboard by holding down the SHIFT key and pressing the DOWN ARROW or UP ARROW. When you select from the keyboard, the first selected line will be the first visible line in the window; it may be necessary to scroll the window up or down to get the proper starting line. After you have selected a range of lines, several options for copying, cutting, or printing the selected text can be obtained by right-clicking in the window, as shown below. A special feature of OSLO text windows is that you can copy lines with either displayed or full precision (from the Spreadsheet buffer).



Text Window tool bar



The Text Window toolbar begins with the Window Setup button. This button is found in all OSLO toolbars, and displays a context-sensitive menu of standard tools that affect spreadsheet, text, and graphics windows, as well as subsidiary tools that can be concatenated to (or replace) the standard tools. The Tile windows item, common to all Window setup buttons, is a user-modifiable command that attempts to lay out the various spreadsheet, graphics, and text windows in a convenient pattern. The layout of tools in the main toolbar is controlled by the Set Toolbars/Row command. The SS right-click actions item pops up a menu showing editing commands that pertain to the selected rows in the current spreadsheet (if any). The SS right-click actions menu also appears when you right-click in a spreadsheet.

Len	Standard surface data
Spe	Special surface data
Rin	Refractive indices
Ape	Aperture data
Wav	Wavelengths
Pxc	Paraxial constants (efl, fnb, ne, etc.)
Abr	Aberration sums (through 5th order)
Mrg	Marginal (axial) ray trace
Chf	Chief (principal) ray trace
Tra	User-specified single ray trace
Ref	Current reference ray data
Fan	Trace a ray fan
Spd	Trace a spot diagram
Auf	Auto-focus for minimum spot size
Var	Show current variables
Ope	Show operands and error function
Ite	Carry out a design iteration

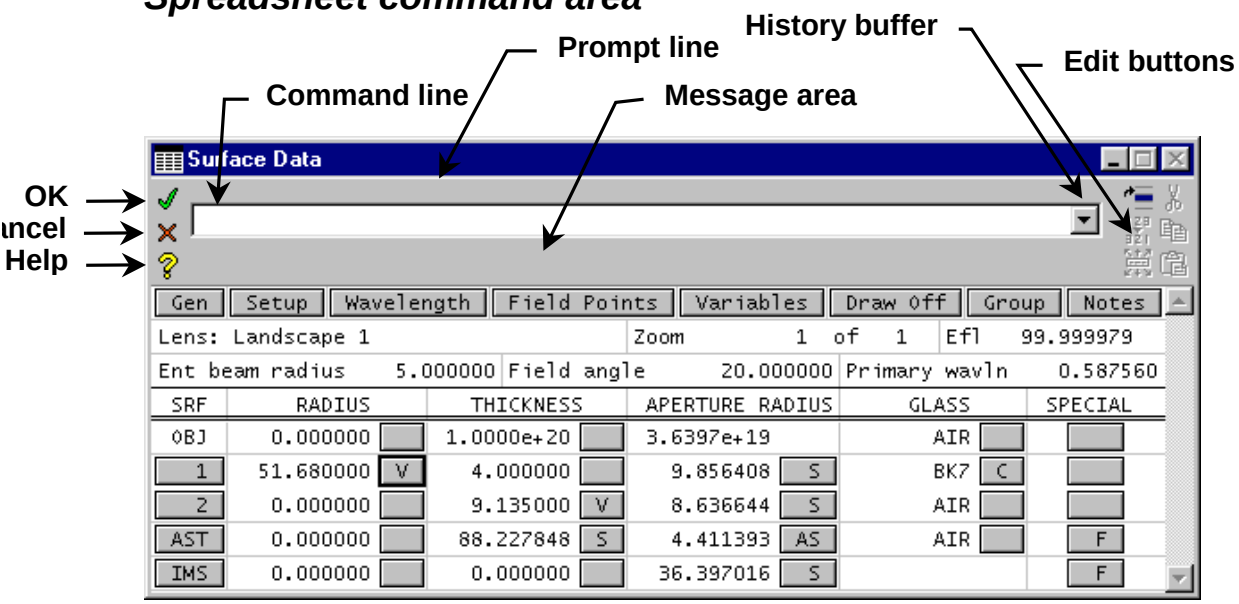
Spreadsheet windows

OSLO uses spreadsheets for data entry. Like other spreadsheets, they contain rows and columns of cells. You can move between cells with the mouse or arrow keys, and you enter data into the cells. There are several dozen spreadsheets, but they all work similarly, so only the principal ones are described in detail here.

Spreadsheets in OSLO are organized as a stack. There can only be one spreadsheet open at a time. If you open another spreadsheet while one is still open, the first one will be pushed down in the stack and will re-appear when the second is closed. The spreadsheet window title bar shows the names of all the spreadsheets currently in the stack.

The OSLO spreadsheet window consists of two parts. The top part is called the command area and consists of a prompt line, command line, and message area, plus buttons at both ends. The bottom area is the actual spreadsheet, which itself may be divided by a double line into a fixed area at the top, and a scrollable area with a variable number of rows at the bottom. Most spreadsheets consist of a number of buttons that open other spreadsheets or pop-up menus.

Spreadsheet command area



Green check button — causes OSLO to accept and terminate the current data entry. It is equivalent to an OK button. If there is no current data entry, the green check button causes OSLO to accept changes and close the spreadsheet. From the keyboard, ENTER accepts the current data entry, but you must press SHIFT+ENTER to close a spreadsheet.

Red X button — equivalent to a Cancel button. The red X is equivalent to pressing ESCAPE, except that when a spreadsheet is open, you must press SHIFT+ESCAPE to close it. The red X button causes OSLO to reject and terminate the current data entry. OSLO has a *Revert_enable* preference that allows you to cancel changes made to a lens in a spreadsheet. If this preference is set, whenever you exit a spreadsheet by clicking the red X button, a confirmation box will appear asking you to confirm that you want to revert to the data that existed when you entered the spreadsheet. Note that the normal response to this query is *yes*, otherwise you would have clicked the green check button.

Help button — opens the OSLO Help system. It opens different pages, depending on the state of the command line and spreadsheet.

- If the command line is empty and no spreadsheet is open, the Help button goes to the main page of the help system.
- If the command line contains the name of a command, the help button opens the command definition.
- If a spreadsheet is open, the help button navigates to the page of the help system that provides primary help for that spreadsheet.

History button — As mentioned before, it is possible to run OSLO using either menus or commands. The names and definitions of OSLO commands are given in the on-line help system. A convenient way to learn commands is to use the **History buffer** in conjunction with the menu system. You can execute a desired command using the menu system, then recall it from the History buffer to see the actual command, which can be entered from the command line. Many commands have two forms that are equivalent. The short form is easier to type; the long form is easier to comprehend. If the *Command_history_aliases* preference is on, the short form of commands is shown; otherwise the long form is shown.

The command area itself consists of three sub-areas for text:

- **Prompt line** — The prompt line contains a prompt that either states the current status of the program or requests specific keyboard input. OSLO contains a special feature that generates prompts for all required input data.
- **Command line** — This is where keyboard input to OSLO occurs. When a spreadsheet is active, keystrokes entered into the program are echoed in both the spreadsheet and on the command line. When the command line expects an argument that is an item from a list of allowed values, a pop-up options list appears containing all the allowed values.
- **Message area** — The message area contains informative messages of 1 or 2 lines from the program. Error messages appear in a separate alert box. The message area is also used for program output, such as calculator results or file actions.

Commands in OSLO are self-prompting. To make it easy to use commands, OSLO uses default values for command arguments (also called parameters) where possible. Any remaining arguments are prompted for. In addition, OSLO has a special question-mark parameter. If you want to be prompted for all arguments, enter a command followed by a space and question mark.

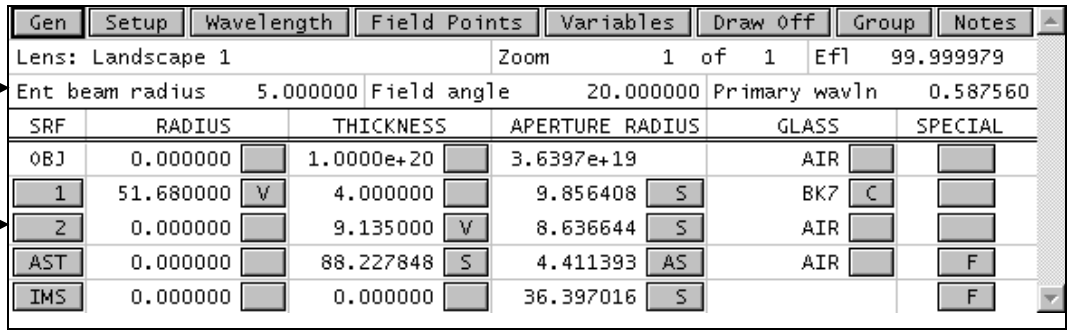
Commands can include symbolic input. All lens data can be referenced by name (e.g. CV[2], TH[i], RN[1][1], IMS, AST etc.). In addition OSLO has named storage registers (A-Z, the Spreadsheet buffer, several pre-defined arrays, standard math functions, and various other named global variables). When you enter a numeric expression in the command line, OSLO substitutes the value of the symbols, evaluates the expression, and enters the result. Several commands can be given on a single line, separated by semicolons. Each line can be up to 512 characters, and loops and other control structures allowed in C can be used.

Commands can also be entered while a spreadsheet cell is highlighted. In OSLO, each cell for numeric data is called a SmartCell™. SmartCells automatically detect and execute commands entered in spreadsheet cells. In certain cases, commands may not be legal entries to cells (e.g. you can't enter the command **cv 2 .001** in the cell for row 3) and are disabled. In addition, where a cell expects a string entry, it may be necessary to enclose it in quotes to prevent it from being interpreted as a command. For example, if you want to use the string “Paraxial_constants” as a lens id, you must enter it in quotes to prevent Paraxial_constants from being interpreted as a command.

Surface data

The Surface Data Spreadsheet (shown at the beginning of this section) is where data that specifies lens surfaces is entered and updated. The spreadsheet is opened from the Update menu, by clicking on the lens tool on the main tool bar, or by entering the command **lse**.

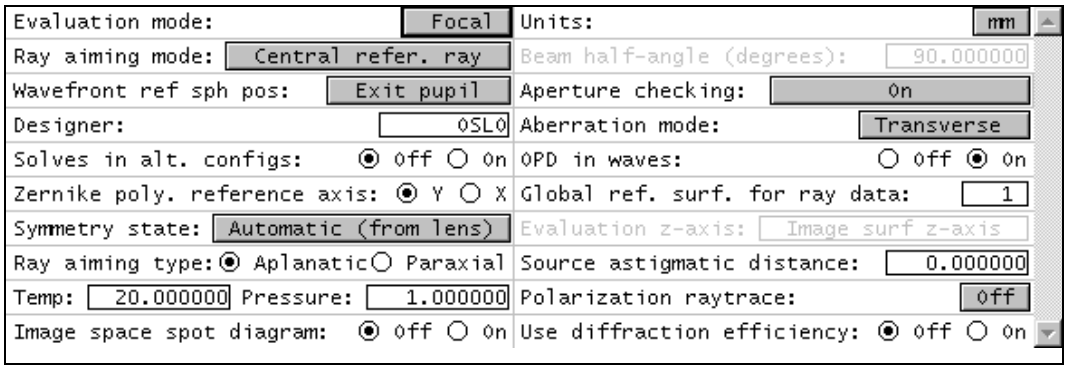
To follow the discussion here, you may wish to run OSLO and open the surface data spreadsheet with the same lens used for the example. To do this, run OSLO, click on File >> Open, and select "Landscape 1.len"from your private/len directory (either by double-clicking on it or by selecting it and clicking Ok), OSLO will load the demonstration lens file used here.



Fixed area

As indicated on the figure, the spreadsheet consists of two areas, a fixed area containing a row of buttons and some data entry fields, and a scrolled area (below the double line) that contains several rows to enter data describing the surfaces of the optical system. The buttons in the fixed area open additional spreadsheets for data input, or provide control options, as follows:

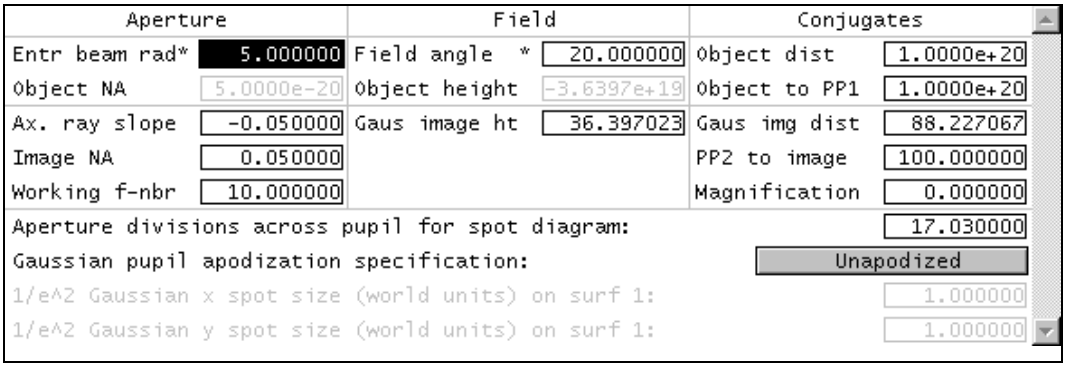
Gen — Opens the general operating conditions spreadsheet. General operating conditions include data that specifies either lens properties or the environment in which the lens is evaluated. Most of these conditions can be used without change for ordinary design work.



Setup — Opens the paraxial setup spreadsheet. Each column of this spreadsheet specifies one paraxial property for the lens. These properties are Aperture, Field, and Conjugates. Each of these properties can be specified in more than one way. You must enter one value in each column (or accept the default) for a complete set of properties. The controlling specification of the aperture or field is shown by an asterisk. Note that the first two specifications (above the line) are object space specifications, while those below the line are image space specifications. In OSLO, if a paraxial property is specified in object space, then image space data will be varied while the object specification is held constant. If the paraxial property is specified in image space, then the object space data will be varied in order to hold the image-space specification. For example, if the entrance beam radius is specified, the f-number of the lens will vary according to the focal length. On the other hand, if the f-number is specified, the entrance beam radius will vary according to the focal length.

The Paraxial Properties Spreadsheet functions like other spreadsheets in that an entry in one cell can affect values in many other cells. For example, if you change the image distance, then the object distance, magnification and working f-number all change accordingly. This means you can enter these properties using the parameters that are most natural for your problem and OSLO will calculate the other ones. Parameters not allowed in the current context are dimmed. Before clicking OK, be sure all values are as you entered them.

It is important to understand that this spreadsheet only sets the initial values of the parameters. It does not place any constraints on the lens to maintain these values. For example, if the focal length of the lens changes during a design, this may change the conjugates and aperture parameters. Note that the basic paraxial operating conditions that specify aperture and field are also shown on the second row of the surface data spreadsheet.



Aperture		Field		Conjugates	
Entr beam rad*	5.000000	Field angle *	20.000000	Object dist	1.0000e+20
Object NA	5.0000e-20	Object height	-3.6397e+19	Object to PP1	1.0000e+20
Ax. ray slope	-0.050000	Gaus image ht	36.397023	Gaus img dist	88.227067
Image NA	0.050000			PP2 to image	100.000000
Working f-nbr	10.000000			Magnification	0.000000
Aperture divisions across pupil for spot diagram:					17.030000
Gaussian pupil apodization specification:					Unapodized
1/e^2 Gaussian x spot size (world units) on surf 1:					1.000000
1/e^2 Gaussian y spot size (world units) on surf 1:					1.000000

The center portion of the paraxial setup spreadsheet contains data entry fields for specifying the sampling size and apodization of ray grids used to make spot diagrams (which are in turn the basis for several other image evaluation calculations). The "Aperture divisions across pupil" item sets the grid size, normalized to the entrance beam radius. In order to have proper sampling, the entrance beam must fill the aperture of the system, and the grid size must be fine enough (i.e. the aperture divisions large enough) to adequately characterize the wavefront transmitted through the system. The worse the performance of the system, the finer the grid size that must be used to evaluate it. Generally speaking, the sampling should be fine enough that there are several grid points per wavelength of optical path difference in the pupil. A rule of thumb for setting the aperture division is to evaluate the system using what you believe to be a proper sampling, then repeat the analysis with half the grid size and ensure that the result does not change.

The other data entry field in the setup spreadsheet allows you specify a Gaussian apodization for systems used to transmit laser beams. This applies a Gaussian weighting to rays that are traced to compute the geometrical wavefront used to calculate far-field diffraction patterns. The spot sizes can be specified independently in the x and y directions, and are expressed either in world units on surface 1, or as object-space numerical apertures for finite conjugate systems. For example, if the entrance beam radius is 2mm, then a spot size of 2 describes a Gaussian beam whose intensity falls to $1/e^2$ of its axial value at a distance 2mm from the axis on surface 1.

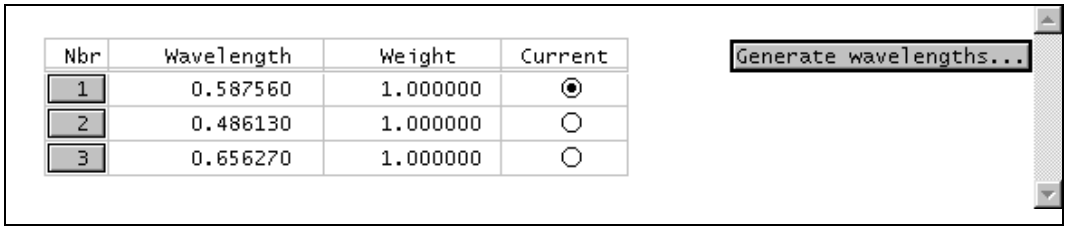
Wavelength — Opens the wavelengths spreadsheet, which allows you to specify which wavelengths OSLO will use for analysis. For a monochromatic application, e.g. a laser, a single wavelength is appropriate. For a polychromatic application, OSLO by default defines three wavelengths for its analysis. You can add more, up to a limit specified by the preference *max_wavelengths* (default = 25). The primary wavelength used for analyzing the current lens is shown at the end of the second row of the surface data spreadsheet.

Wavelengths should be spaced across the wavelength range of interest. They should be entered such that the central wavelength is #1. The remaining wavelengths are given in short-long pairs. For example, in the visible spectrum, the order should be green-blue-red. This will make the chromatic aberrations meaningful. Wavelengths in OSLO are *always* given in micrometers.

The evaluation routines in OSLO are based on the *current* wavelength, normally set to #1. If the current wavelength is set to some other value, aberrations and (single wavelength) ray-trace analyses will be carried out in that wavelength. If the wavelength used for such an analysis is not wavelength 1, the text output indicates which wavelength was used.

Each wavelength can be assigned a weight, which is used in optimization and in spot diagram analysis routines (rms spot size, MTF, PSF, etc.)

Editing the Wavelengths Spreadsheet is like editing the Surface Data Spreadsheet. Rows or groups of rows are selected, cut, copied, pasted, and inserted in the same way. For example, to reduce the number of wavelengths from 3 to 1, click on row button 2, then shift-drag to row button 3. Rows 2 and 3 are now selected. Click the scissors tool on the tool bar or press Delete. Only row 1 remains. To insert additional wavelengths, select a row button, then click on the Insert After toolbar icon, or press SHIFT+SPACE to create an additional line, then fill in the required data.



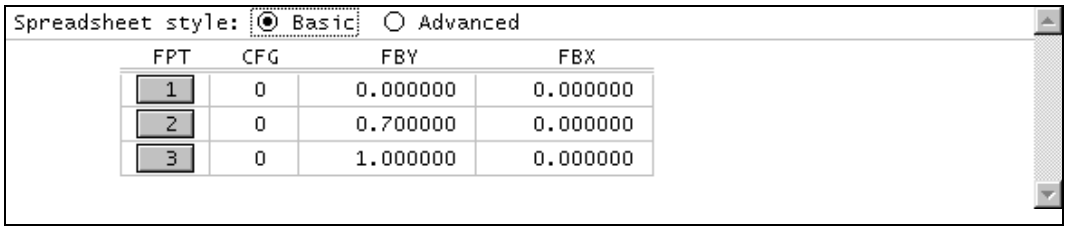
Nbr	Wavelength	Weight	Current
1	0.587560	1.000000	◉
2	0.486130	1.000000	○
3	0.656270	1.000000	○

Generate wavelengths...

To change wavelengths, click in the wavelength SmartCell. When this cell is selected, you may enter the new wavelength manually, or click on the cell again to display the wavelength menu. This menu includes standard wavelengths throughout the range most often used by OSLO.

The Wavelengths spreadsheet in OSLO Premium and Standard contains a button that can be used to generate wavelengths and wavelength weights automatically. The data are generated according to a Gaussian quadrature scheme that provides uniform spectral weight across a user-specified spectral band. OSLO SIX also has an optimization operating condition that allows user-defined weighting.

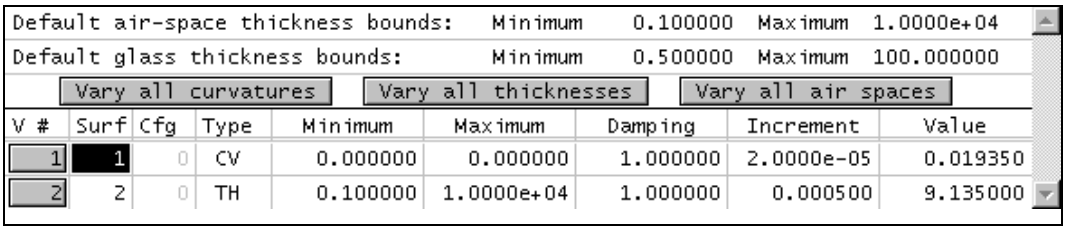
Field Points — Opens the field points spreadsheet (not available in OSLO Light), which defines points on the object surface to be used for evaluation. Field points are an essential part of OSLO, so in OSLO Light they are predefined to be at fractional coordinates (relative to the object height) of 0.0, 0.7, and 1.0. In OSLO Standard and OSLO Premium you can edit the field point data or add additional field points using the field points spreadsheet. In addition to the basic field point data that specifies the location of the field points, you can specify advanced data describing the vignetting of the system allowed during optimization, and other field-dependent properties.



Spreadsheet style: ◉ Basic ○ Advanced

FPT	CFG	FBY	FBX
1	0	0.000000	0.000000
2	0	0.700000	0.000000
3	0	1.000000	0.000000

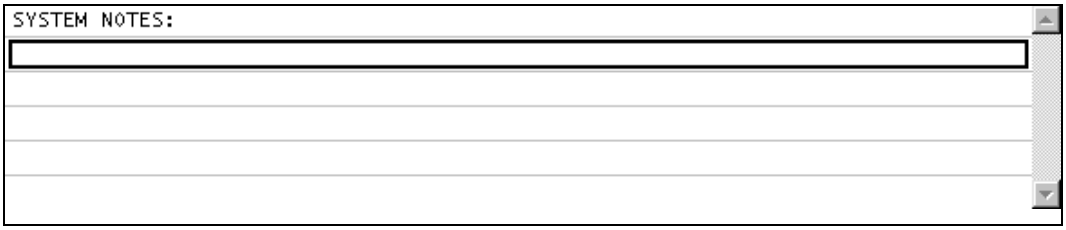
Variables — Opens the variables spreadsheet, which is used to provide a complete specification of optimization variables, including boundary conditions, derivative increments, etc. Basic surface data (radii, thicknesses) can be specified to be variable in the normal surface data spreadsheet, but special data (tilts, decenters, aspheric coefficients, etc.) must be specified using the variables spreadsheet. In addition to providing detailed specification of variables, the spreadsheet contains buttons for convenient input of multiple variables (all curvatures, etc.)



Draw On/Off — Turns the Autodraw window on/off, as described above. When Draw is turned *on*, OSLO opens a window with a plan view of the lens. This view includes the rays selected by the current operating conditions. The surface or surfaces currently selected on the Surface Data Spreadsheet are shown in a different color. This view is updated whenever the lens data is changed or whenever the highlight is moved to a new surface in the spreadsheet. If Draw is turned *off*, the Autodraw window is automatically closed.

Group/Surfs — Switches between surface and group views of the data. In *Surfs* mode, each surface is listed individually with values for all parameters, including the radius of curvature. In *Group* mode, grouped surfaces are shown as a single entity that takes two lines in the spreadsheet. For example, with a catalog lens, the radius column of the spreadsheet displays the part number of the element. The element can then be treated as a whole, so all its surfaces can be edited (cut, copy, paste, etc.) at the same time. If there are no groups in a system, this setting has no effect. Non-sequential groups display only the entrance and exit port surfaces in group mode. User-defined groups similarly show the entrance and exit surfaces.

Notes — Opens the system notes spreadsheet. This spreadsheet allows you to attach up to 5 80-character notes to a lens. In addition to descriptions, notes are sometimes used to hold special data, or even short scp programs that are specific to a particular lens.



The line below the buttons in the fixed area portion of the lens spreadsheet contains three fields. The first is a 32-character field that holds the title (also called the Lens ID) of the lens. This text is used in several graphic plots. The second field is the zoom field, described below. The rightmost field is not a data entry field, but is a display field that shows the current focal length of the lens. In addition to providing information about the lens properties, this field is often used as a check field to ensure correct data entry.

Zoom — The zoom field changes to a button when there is zoom data in the system (OSLO Standard and OSLO Premium only). To activate the zoom button, enter a number greater than 1 in one of the numeric fields:

Once the button appears, you can use it to open the zoom data spreadsheet. In addition to thicknesses, most other system and surface data can be zoomed. When non-thickness data is zoomed, the system is often called a multiconfiguration system. Zoomed systems are ones that are optimized simultaneously in different configurations, so that the performance is not optimum in any particular configuration, but is rather optimized for the ensemble of all configurations. It is

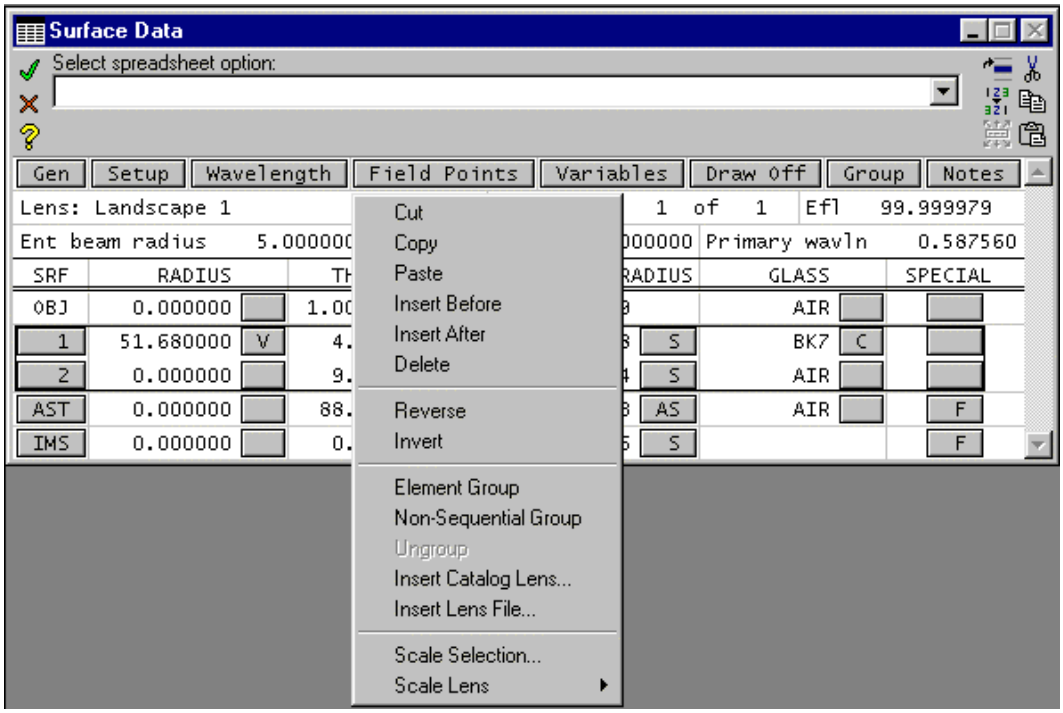
commonly thought that it is necessary to have zoom optimization to design a zoom lens, but this is not generally true. In any case, once zoom data are entered for a system, the arrow buttons next to the zoom button permit you to quickly change from one configuration to another.

NBR	ITEM	SURFACE	ZOOM	QUALIFIER	VALUE
1	TH	5	2	0	48.678000
2	TH	10	2	0	1.522600
3	TH	13	2	0	5.133000
4	ANG	0	2	0	6.160000
5	EBR	0	2	0	25.528571

Scrolled area

Below the fixed area of several spreadsheets is a scrolled area that contains a variable number of rows, each having a button at the left-hand end (whether or not the area needs to be scrolled depends on how many rows are defined, relative to the vertical extent of the spreadsheet window). Although the surface data spreadsheet is the most conspicuous example of a spreadsheet with a scrolled area, there are many others (wavelengths, field points, variables, etc.) and they all work similarly. Each is a form of object editor, where the objects are the entities described by the data in the rows. For example, in the surface data spreadsheet, the objects are lens surfaces. The spreadsheet allows you to manipulate the objects by selecting the rows of interest and then performing the desired action.

Row buttons — At the left end of a row is a row button. To select a row, click on the row button. If there is no row button, you can't select the row; this is the case when a row is part of a range, or when row selection is disallowed for some reason (e.g. the system is in an alternate zoom configuration). To select a range of rows, first click on the row button at either the beginning or ending of the desired range and then drag the mouse down or up. Alternately, you can select using the keyboard by holding down the shift key while using the arrow keys.

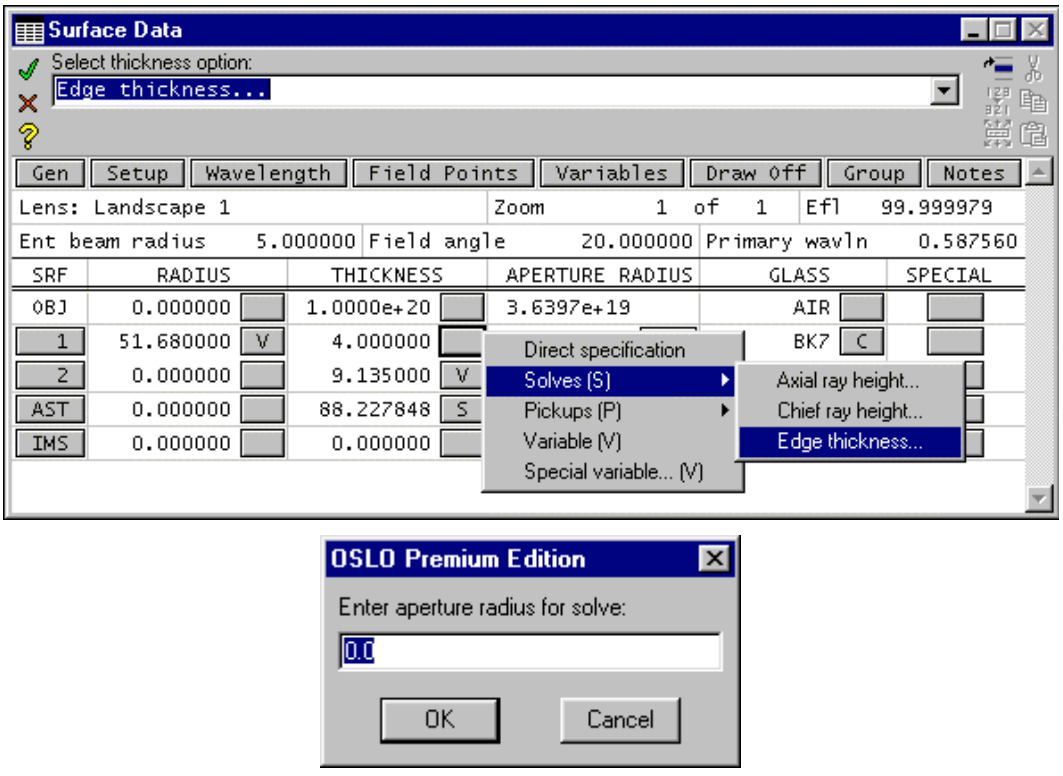


Once a row or range is selected, you can use the edit menu by right-clicking in the spreadsheet. Alternately you can use the tools at the right-hand end of the command line, which will become active when a row is selected in the spreadsheet. The edit menu contains the usual cut, copy, and paste tools, but it is important to realize that these tools work on the objects represented by rows, not text. For example, to duplicate a lens element in an optical system, you select the range of surfaces it encompasses, copy the range to the clipboard, and then paste it in wherever you want. The surface numbering of other surfaces is automatically adjusted.

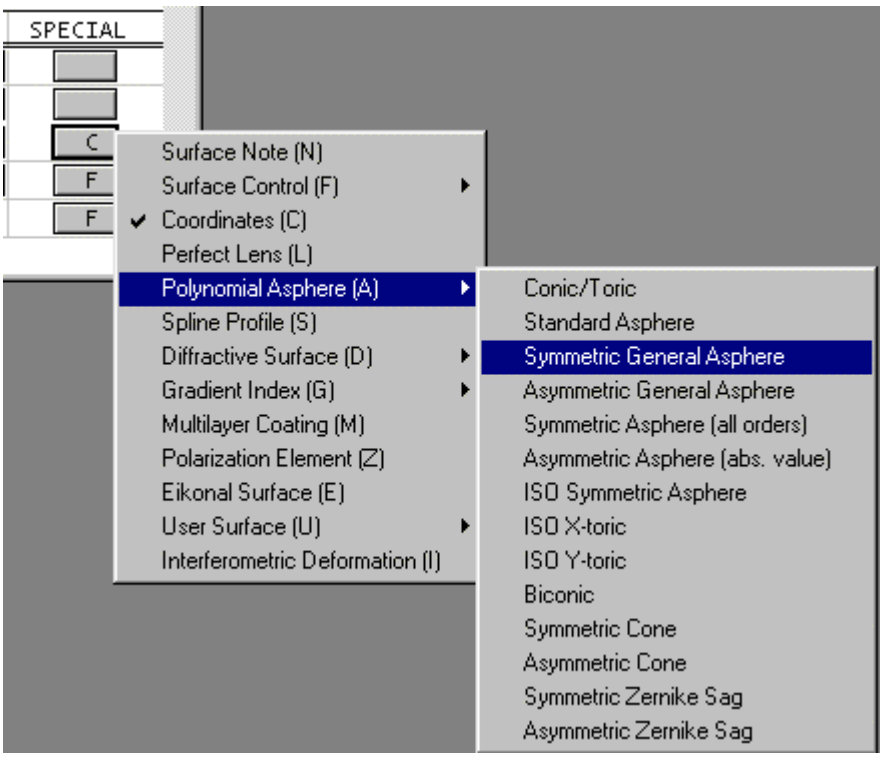
In addition to the standard clipboard management tools, the OSLO edit menu contains items for inserting and deleting rows, as well as a special tool used for reversing ranges of rows, which is particularly useful for manipulating lens elements. Note that although spreadsheet row editing is illustrated here with reference to lens surfaces, the method is applicable to coating layers, wavelengths, variables, operands or other data that are organized by rows.

In addition to the *reverse* tool mentioned above, there is an *invert* tool. The invert tool is used to change the signs of thicknesses of elements that follow a mirror, a feature that is useful when working with retro-reflecting (double-pass) systems. Other items in the edit menu allow you to group and ungroup ranges of surfaces as elements, or change a range of surfaces to or from a non-sequential group. Finally, you can use the selection mechanism to indicate a point in a lens to insert data from a lens file or the catalog lens database.

Options buttons — Row buttons enable selection to manipulate entire rows. Spreadsheets also contain options buttons, which permit alternate choices for specifying data. To the right of each spreadsheet cell is a button used to select various options to specify data in the adjoining cell, for example, applying a solve or pickup constraint. Click on the button with the mouse, or select the button with the arrow keys and press the space bar. A pop-up menu is displayed to show the possible choices. Selecting an item will often pop up a dialog box to enter data, as shown below.



The options buttons in the Special column are more complex than the others, because they provide access to special data that is not always present in a lens. For example, the aspheric surface options are shown below. In order to accommodate a wide variety of options while maintaining efficient utilization of computer resources, OSLO only allocates memory for most types of special data when it is actually used. In the present case, you can specify the order of the polynomial in the spreadsheet that pops up when the Symmetric General Asphere surface type is activated, and the required data entry cells are created. If you later change the order, cells are created or destroyed (while maintaining previous data entries) according to your request. If you want to remove the Symmetric General Asphere property from the surface altogether, you click the button at the bottom of the spreadsheet as shown.



Surface 2

ASP ASR 18

AS0: r^0	AS1: r^2	AS2: r^4	AS3: r^6	AS4: r^8
0.000000	0.000000	0.000000	0.000000	0.000000

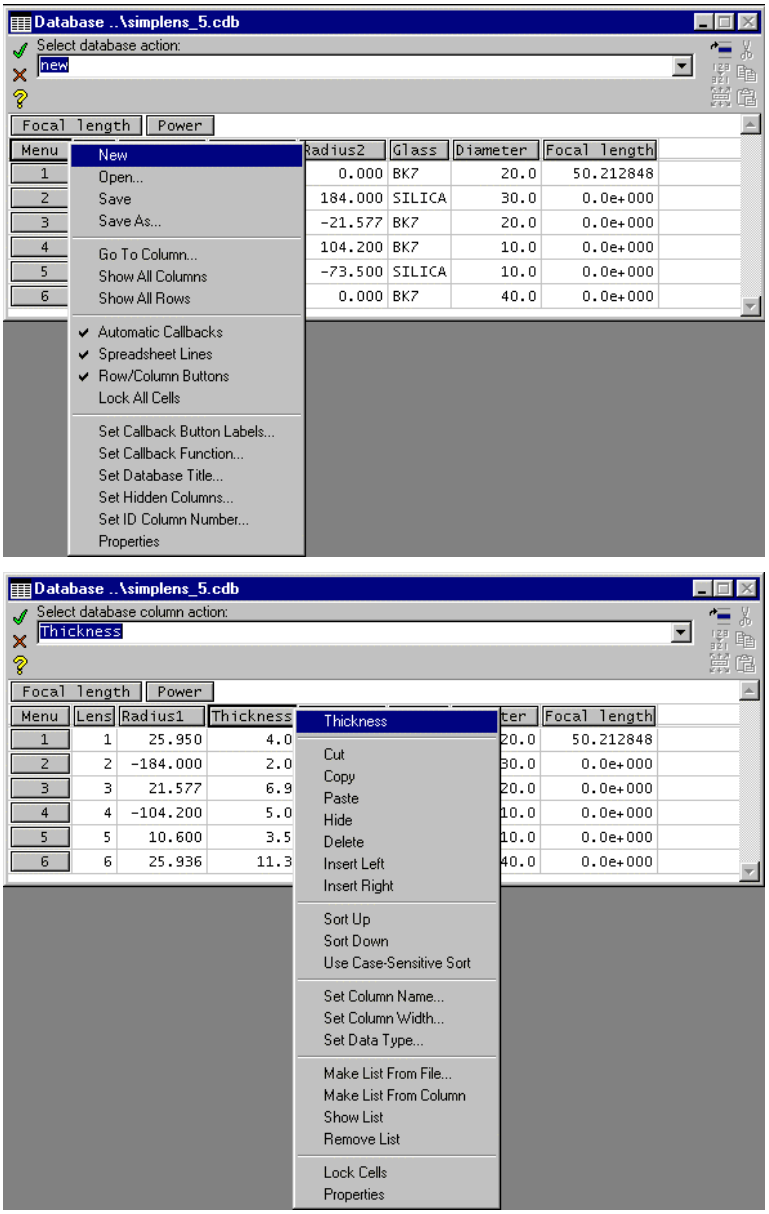
AS5: r^10	AS6: r^12	AS7: r^14	AS8: r^16	AS9: r^18
0.000000	0.000000	0.000000	0.000000	0.000000

Delete General Asphere Data

There are many additional spreadsheets that are used to enter data into OSLO. All of them work according to the principles described for the surface data, paraxial properties, and wavelengths spreadsheets described above. Most are provided for entering operating conditions, special surface data, or optimization data. A few, such as the Gaussian beam spreadsheet and the slider-wheel spreadsheet, are used to set up special analyses and are described in various examples. One spreadsheet that has a broader applicability is the database spreadsheet described below.

CCL database spreadsheet

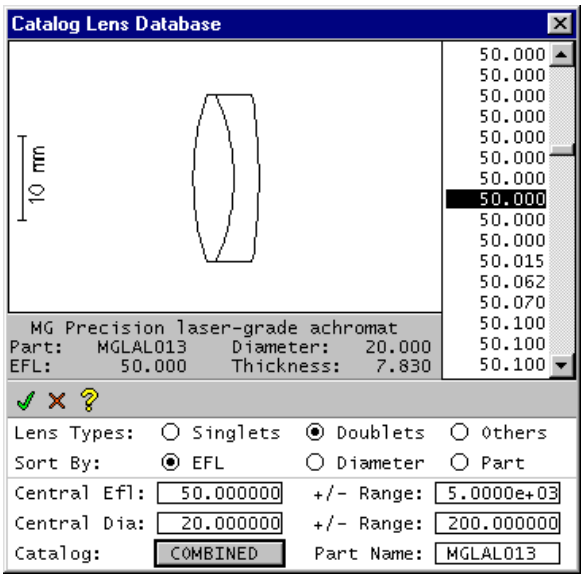
The CCL database spreadsheet provides the capability for you to add your own spreadsheets to OSLO. The spreadsheet works similarly to the built-in spreadsheets (row selection, cut-copy-paste, etc) and works with files written in *.cdb format, which is similar to standard delimited-data formats (comma or tab-separated) supported by general-use spreadsheet software. The CCL database spreadsheet provides two menus that allow convenient editing of *.cdb files. The first is an overall control menu, which provides overall control of the spreadsheet and links it to CCL callback routines, which provide the processing functions that handle cdb data. The second menu pops up when you click on a column button, and allows you to insert, move and manipulate data on a column basis.



The CCL database spreadsheet works in conjunction with the CCL database library, which provides functions for putting and getting data from the database, as well as sorting and other editing functions. The CCL database supports the same data types as CCL (real, integer, character, and string).

Catalog lens database

OSLO includes a pictorial database facility that allows you to search stock lens databases efficiently. You can sort by focal length, diameter, and part number. You can limit the search range by focal length and lens diameter. Once you select an element, OSLO enters it into Surface Data Spreadsheet ready for use.



The Catalog Lens Database facility can be accessed using the File >> New command and selecting the Catalog Lens option, or from the Edit menu using the Insert Catalog Lens option. In the first case, OSLO opens a new lens file for the element selected from the database. In the second case, OSLO merges a catalog element into the current Surface Data Spreadsheet, in front of the selected row.

Sort controls — These allow you to sort the database in several different ways. The three *Lens Types* option buttons determine what type of lens to include in the list. The three *Sort By* option buttons determine which lens parameter to use as an index. The elements are sorted and displayed in a list according to this parameter.

Range controls — These allow you to restrict the range of the elements that appear on the list. For example, if you set the central effective focal length (*Efl*) parameter to 50 mm, and the *+/- Range* parameter to 10 mm, then only lenses with *Efl*s between 40 mm and 60 mm appear on the list. The central diameter (*Central Dia.*) can further restrict the list.

List — Once you have established a restricted list, you can scroll through it using the scroll bar or arrow keys. Each element includes a drawing, so you can see its form. The message area provides additional information about the element.

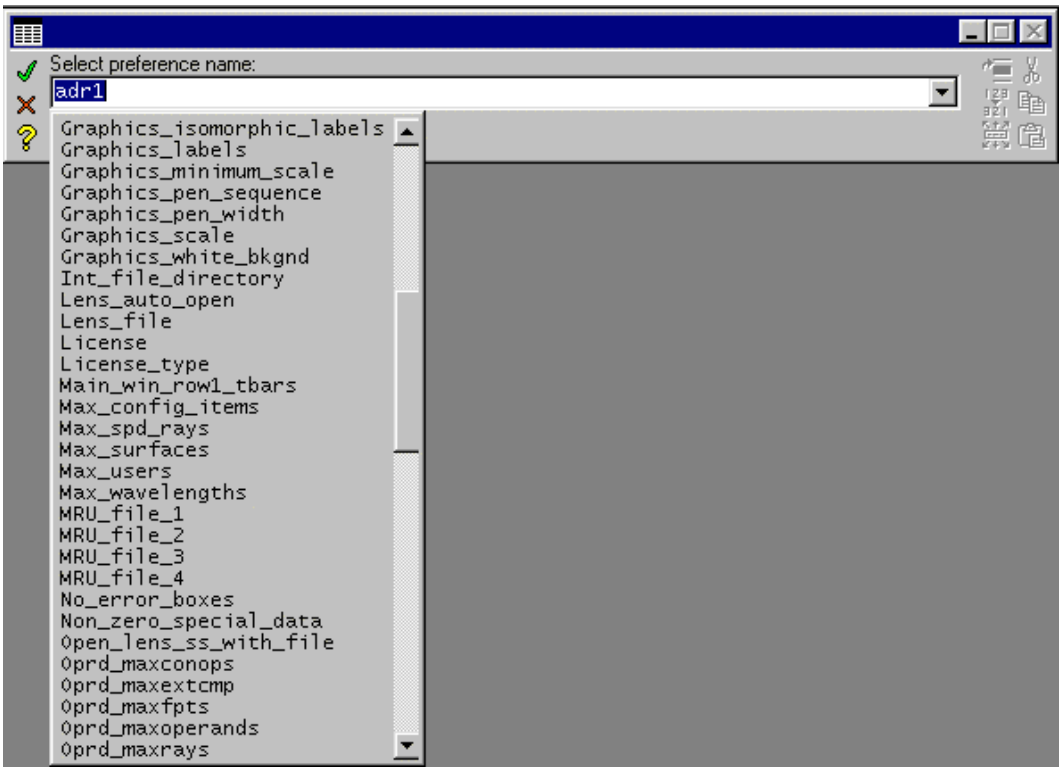
Selecting a part — To choose an element, double-click on its parameter in the list, or select it and click OK. If you know the part number of the element you need, you can also enter it manually in the *Part Name* cell. The element is then called to the list, regardless of the sort and range criteria.

When an element from the database is entered on the Surface data spreadsheet, its surfaces are set up as a type called “group.” All its radii, thicknesses, and glasses are marked “fixed” so they cannot be inadvertently changed. If you want to remove this constraint, you can “ungroup” the element using the radius button next to the part number in the surface data spreadsheet.

Changing catalog databases — OSLO contains several separate lens databases covering a wide range of components from various suppliers. The name of the current database is displayed in the Catalog Lens Database window title bar. Select the *Catalog* cell to call up an options list of the available catalog databases.

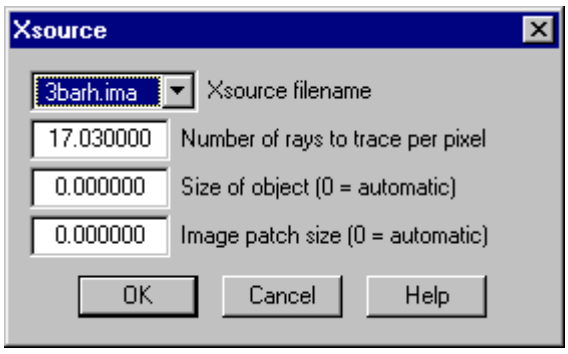
Pop-up windows

Options lists — Many OSLO commands and buttons require a choice from a list of several arguments. OSLO provides this list of arguments either in a pop-up menu or options list. Pop-up menus work similarly to other menus. You can then select an element from an options list by clicking on it or by selecting it with the arrow keys and pressing ENTER. You can also enter the desired value on the command line. If you decide not to perform the command, click the red X (or press ESCAPE). You must complete the command or cancel it before OSLO will perform other operations.



Dialog boxes

OSLO contains many dialog boxes, used to enter arguments for commands. Although a few of the dialog boxes are similar to standard Windows controls (e.g. the File dialog box), most OSLO dialog boxes are built by OSLO. There are two types. Dialog boxes that support internal OSLO commands are pre-compiled and have a fixed format. CCL dialog boxes are built as needed when the program is run, and have a variable format that depends on the display used to run the program. Internal dialog boxes have SmartCells and accept symbolic input and expressions. CCL dialog boxes, however, are limited to typed input of literal values. The following is a typical CCL dialog box.


Xsource
3barh.ima
Xsource filename
17.030000
Number of rays to trace per pixel
0.000000
Size of object (0 = automatic)
0.000000
Image patch size (0 = automatic)
OK
Cancel
Help

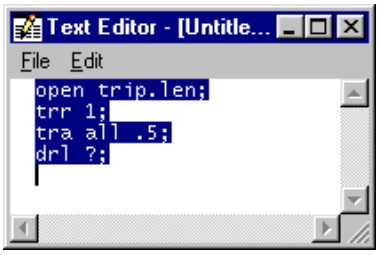
Text editor

OSLO can use either an external editor or its own internal editor. The OSLO text editor is a small memory-based editor suitable for a variety of small jobs, such as writing short SCP or CCL programs, or directly modifying lens files. For more substantial editing tasks, such as CCL development involving multiple files, an external editor is recommended. If an external editor has been installed, it is still possible to invoke the OSLO editor using the Window >> Editor >> Open command.

The OSLO text editor is linked to the CCL compiler in OSLO, so that when you save a CCL file in the private CCL directory, it is automatically compiled. If you use an external editor, you must re-compile your private CCL after saving or modifying a file in the CCL directory. A button is provided on the main toolbar for this purpose.

Note that if a lens file is opened using the File menu in the editor, it will be brought into the editor as a text file, while if it is opened from the File menu in the main window, it will be opened as binary data in memory to be used by the program.

A unique feature of the OSLO text editor is its ability to execute selected text as a series of commands. In the above window, four commands have been entered and selected in the edit window. By pressing CTRL+E, you can execute these commands. Note that the last command has been entered with a question mark, which means that you will be prompted for its arguments.



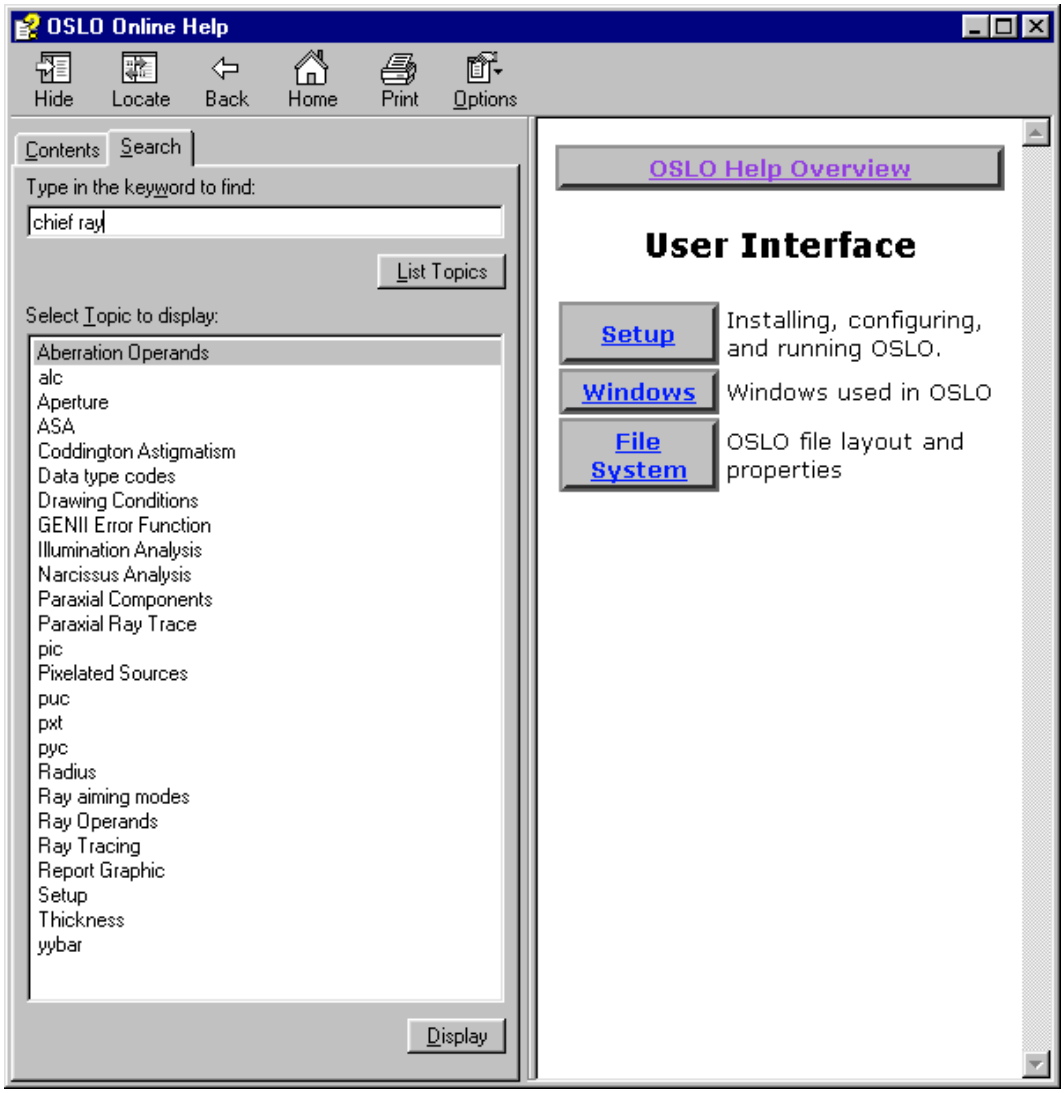
Help window

OSLO has an extensive help system based on HTML help. Most information on how to use the program, including command descriptions and definitions, how to accomplish various tasks, and the meaning of various data items, are contained in the help system. The overall OSLO help system contains more than 2000 topic files and 8000 links, and is the primary source of information on using the program. OSLO help is supplied in the form of a *.chm file (compiled html), a Microsoft proprietary format. To access the file, the essential components of Microsoft Internet Explorer must be installed on your computer (you do not need to use Internet Explorer as your web browser). During OSLO installation, the setup program checks for these components and offers to update your system if they are not present.

OSLO help is divided into two general parts. The first part consists of a few hundred topic files written manually, which describe how to use the program. The second part consists of a file prepared automatically during compilation that lists the definition of each command contained in OSLO. Because these files are prepared directly from the program source code, they are always up to date and automatically give a correct description of how the commands work.

You can access the help system directly, using the Help button on the main menu. In addition, you can obtain context-sensitive help through buttons in spreadsheets and dialog boxes. Finally, there is a help button directly to the left of the command line. If you click this button when there is a spreadsheet open, it will take you to the part of the help system that describes that spreadsheet. In addition, if you enter the name of a command in the command line and click the help button, it will take you to the command definition. Finally, if you click the button when the command line is empty and there is no spreadsheet open, it will take you to the top level of the help system, and

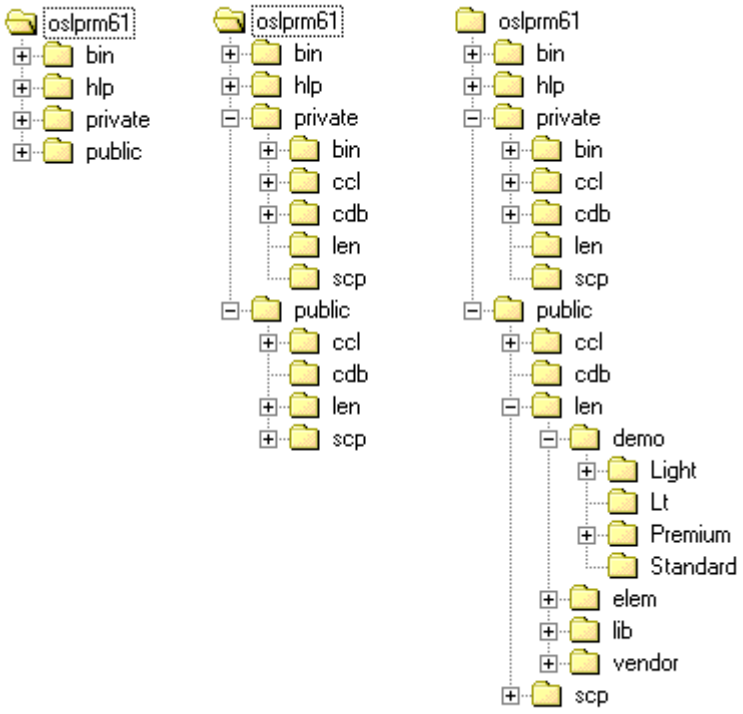
you can navigate to the information you seek, using hypertext links. The search button in the help system provides a useful tool for finding general information.



File system

OSLO deals with several different types of files. The types of files used by the program include lens files, program source files (SCP and CCL), glass catalogs, database files, help files, help graphics files, test glass lists, CCL executable files, and configuration files. OSLO files are organized as shown in the diagram below, which shows the installation directory with three degrees of expansion.

At the top level, the installation directory is divided into four subdirectories: bin, help, public, and private. The bin directory contains OSLO byte-code executable code, plus glass catalog data, dll's, bitmaps, and other support files. The help directory contains the main OSLO help file _oslohelp.chm, plus other help-related files. The two main user directories are called public and private. The public directory contains data supplied with OSLO, while the private directory is intended to hold user-generated files. The second and third columns in the diagram show expansions of these directories, which may be helpful in locating files. The lens files used in this document are located in the appropriate subdirectory of the public/len/demo directory (mostly in the Lt directory).



The commands File >> Open and File >> Save As bring up a Windows dialog box. The dialog box contains, in addition to the standard controls, two buttons that switch to the OSLO private directory or the public directory. Files shipped with the program are contained in the public directory, while files that you develop should generally be placed in the private directory.

Lens files are normally opened using the File menu in the main window, while CCL source files are normally opened using the File menu in the text editor window (or an external editor). Note that OSLO lens files are ordinary text files that contain sequences of commands needed to enter data for a lens.

Data concepts

To use OSLO effectively, you should understand the basic concepts of how it handles data. There are three general groupings of data in OSLO: surface properties, operating conditions, and preferences. You will find the program easy to use if you note this, and try to think in terms of the type of data you are manipulating.

Surface properties — The primary objects that OSLO deals with are surfaces. This places it in contrast with programs (such as TracePro) that deal with solids. Surfaces are described by some sort of equation $F(x,y,z) = 0$, and in optical design we are concerned with what happens to rays, wavefronts, or beams when they pass through surfaces. In addition to a defining equation, surfaces have a region of interest, which is called the surface aperture. In general, solids are accommodated by defining the surfaces that surround them, together with an aperture definition that limits the region to the portion of the surface that physically exists. A few surfaces, such as quadratic solids (spheres, ellipsoids) can be defined as either open or closed. More complicated geometries require the specification of several surfaces per solid object.

The advantage of OSLO's surface-based data structure is that surfaces that are not optically significant need not be defined. For example, in routine optical design, the edge of a lens is not significant. By leaving it out of the data definition, we can simplify the optical specification of the system. Moreover, in most instances, we can state the order in which surfaces are crossed by rays. This leads to what are called sequential systems, which are most efficiently traced. More generally, we have non-sequential systems, in which the ray trace algorithm must not only find the intersection of a ray with a surface, but also find which of the possible surfaces is the one actually struck. Non-sequential ray tracing is only possible in OSLO Premium.

Operating conditions — In addition to the data needed to specify the shape and aperture of surfaces that make up an optical system, there are several data items that need to be specified, but which don't relate to any particular surface. These are called operating conditions, or simply conditions. Some relate to the conditions of use of the optical system, such as the size of the object or image, the aperture of the illuminating beam, or the temperature. Others relate to the conditions of evaluation, such as the sampling grid size for spot diagrams or the ray trajectories to show on lens drawings. Other data of this general class are the sets of data that define the optimization error function: the field point set, the ray set, and the operands. Finally are the variables and the configuration (zoom) data, which are maintained separately from the surface properties. All of the operating conditions are *scoped* to the lens (i.e. they are generally different for different lenses) and are saved in lens files.

Preferences — Preferences are data items that relate to the overall OSLO program, not to a particular lens. Simple examples of preferences are the *output_text* and *page_mode* items discussed previously, but there are dozens of preferences of different types in OSLO that can be checked and in most cases reset using the File >> Preferences command. Preferences are stored in the *oslo.ini* file in the private/bin directory.

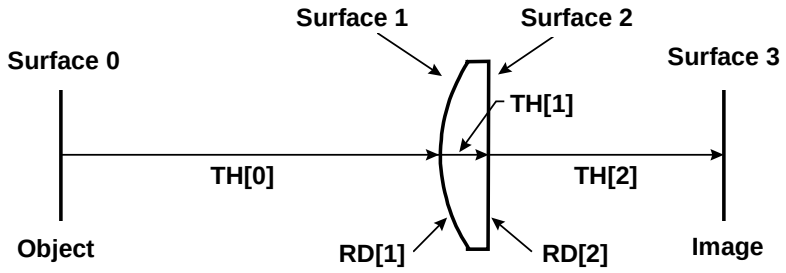
Surface data conventions

Surface numbering

A sequential optical system is specified in OSLO as a series of surfaces that a ray intersects consecutively in passing through the system. Light enters the system traveling from left to right. The object surface is given the number 0. In a sequential system, surfaces are numbered in the order in which a ray or its extension (in the case of a *virtual* surface) would intercept them. The highest numbered surface is called the image surface, whether or not there is an image formed on it. The correct ordering of the surfaces is essential for sequential ray tracing in OSLO.

Quantities that pertain to a surface, such as curvatures, aspheric coefficients, etc., carry the number of the surface. Quantities such as refractive indices, thicknesses, etc., that pertain to the space between two surfaces, are assigned to the lower-numbered surface.

Nominally refracting surfaces having no refractive index change have no effect on ray trajectories, and are called dummy surfaces. A dummy surface is often used to keep track of ray data, or to establish a base coordinate system for other surfaces.

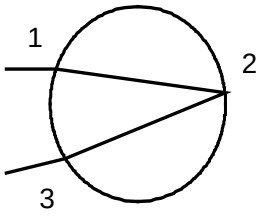


Sign conventions

The proper signs for curvatures, thicknesses and indices are determined easily for systems that do not contain tilted surfaces. Then the following rules apply:

Sign conventions for centered systems

RADIUS OF CURVATURE	The radius of curvature, or curvature of a surface is positive if the center of curvature lies to the right of the surface.
THICKNESS	The thickness separating two surfaces is positive if the next surface lies to the right of the current surface; otherwise it is negative.
REFRACTIVE INDEX	OSLO expects all refractive indices to be provided with positive signs. Reflecting surfaces are specified explicitly by the designation, **rfl**.



A simple example illustrates the sign conventions used for setting up optical systems. Consider a glass bead in which light enters from the left (1), reflects from the back edge (2), and then emerges from the same surface that it entered (3), as shown in the figure to the left. The correct surface data for this system is shown below:

```
*LENS DATA
Glass bead
 SRF  RADIUS    THICKNESS   APERTURE RADIUS    GLASS
  0    --      1.0000e+20    1.0000e+14          AIR

  1  5.000000   10.000000      4.999999 A        BK7
  2 -5.000000  -10.000000      4.999999      REFLECT
  3  5.000000      --          4.999999          AIR

  4     --         --          4.999999 S
```

Surface types

OSLO provides many ways of identifying surfaces beyond direct specification. These methods fall into three groups: pickups, solves, and variables.

Pickups pick up parameters from another surface and apply them to the current surface. This is useful, for example, in designing a system with a mirror that reflects the light back through some part of the system. The surfaces in the path back from the mirror are the same physical surfaces as in the path toward the mirror. By specifying the radii of the surfaces in the path back as pickups from those in the forward path, you guarantee they will always be identical.

Solves tell the program to calculate the value of a parameter that satisfies a given condition. For example, specifying the thickness of the next-to-last surface of a lens by an axial ray height solve with height zero tells the program to calculate the thickness of this surface so that the axial ray will pass through the vertex of the image surface. This is the same as saying the image surface remains at the paraxial focus. As other surfaces change, OSLO changes the thickness of the next-to-last surface to satisfy the condition.

The ***Variable*** specification (which is only applicable for directly specified items) tells OSLO it may vary the item during optimization. In addition to allowing the free variation of surface data during optimization, variables can be constrained to targeted values using Lagrange multipliers. This allows you to construct elaborate constraints that go beyond the solves and pickups enabled as basic data specifications.

The various ways to specify lens data rely on the fact that whenever lens data is changed, OSLO retraces the axial and chief ray through the system, resolving all pickup and solve requests. The routine that does this is called lens setup. It is an essential part of an interactive program.

Introductory Exercise - Landscape Lens

This section shows how to enter lens data in OSLO from first principles. The overall goal is not only to learn the mechanics of entering data, but also to learn how to apply some of the standard tools in OSLO to gain an understanding of the optical performance of a simple landscape lens similar to that used on low-cost cameras. You should follow the steps exactly as shown here. When you have successfully completed the formal steps, you can return to the beginning and repeat this exercise with some side trips to satisfy your curiosity.

In a cookbook example of this type, it is important that you start with the same configuration as the machine used to make the example. If you have changed many preferences before starting this exercise, you can restore your preferences to *factory-default* condition by deleting the file *oslo.ini* in your private/bin directory. Normally this will not be required.

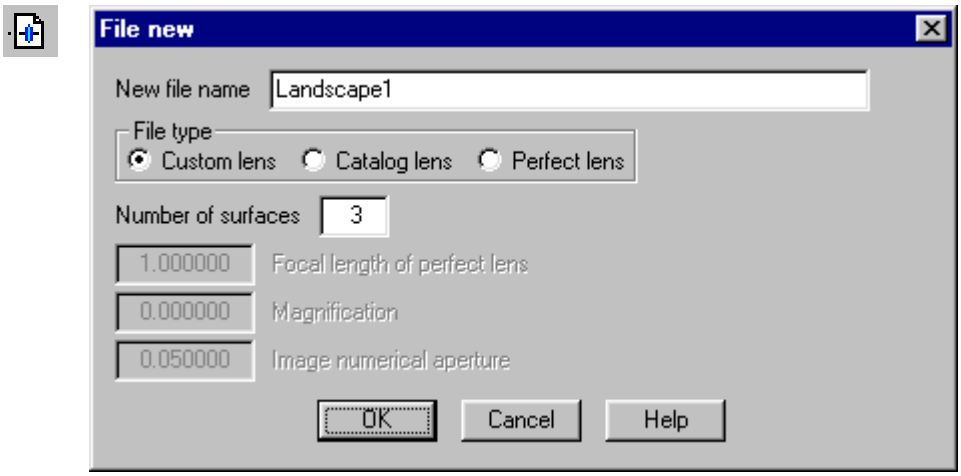
The steps in the exercise are as follows:

- **Lens entry** - Enter a convex-plano lens with a displaced aperture stop behind the lens.
- **Lens Drawing** - Set up the lens drawing conditions to show desired ray trajectories.
- **Optimization** - Optimize the lens so it has no coma, a focal length of 100, and covers a field of ±20 degrees at an aperture of f/10.
- **Slider-wheel design** - Attach sliders to parameters so you can analyze trade-offs.

You will see during the exercise that design with OSLO is quite different from most software. You don't set up the system and then press an *Auto* key that lets the computer optimize it for you. Instead, you work in a highly interactive mode, taking small steps so you can understand what is happening as you progress towards a solution.

Lens entry

Click the New Lens tool in the main toolbar to open the following dialog box.



Enter the file name "Landscape1", select Custom lens and enter 3 surfaces, then click OK. A new spreadsheet will appear. Fill out the spreadsheet with the following data.

In the fixed area of the spreadsheet, set the Lens: cell to "Landscape Lens Exercise", the entrance beam radius to 5, and the field angle to 20. Leave the other cells with their default values.

In the scrolled area of the spreadsheet, enter "BK7" for the glass on surface 1.

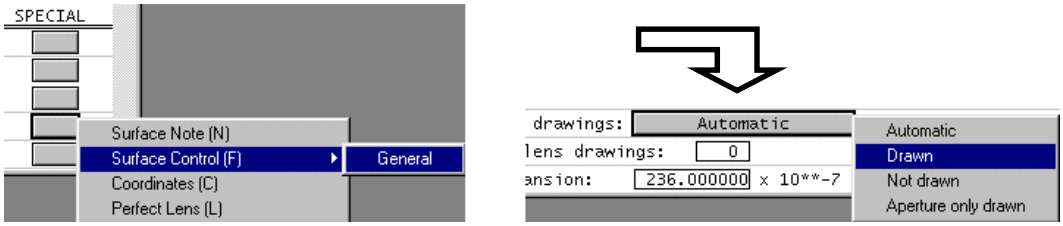
In row 3, click the button in the Aperture Radius cell, then click Aperture Stop from the list that pops-up. Note that the row button for row 3 will now say AST, and A will be added to the button in the Aperture Radius column.

Now click in the Radius cell of row 1, enter 50 for the radius of curvature, then click on the thickness cell and enter 4 for the thickness. Click on the thickness cell for row 2 and enter 10.

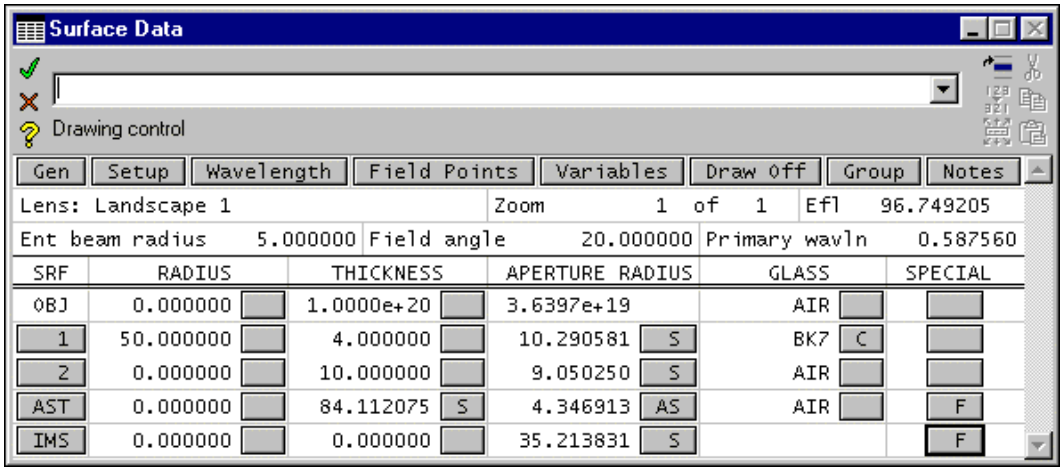
In row 3, instead of entering a thickness directly, click on the button in the thickness cell, and select Axial ray height solve from the pop-up menu. This will cause a data-entry box to pop-up prompting for the value of the solve. Accept the default value (0) by clicking OK.

Click the button in the Special cell for row 3. From the list that pops up, select Surface control, and from the flyout menu, select General. This will cause a whole new spreadsheet to cover up the current spreadsheet. Click the button that says Automatic in the 5th row, and from the list that pops up, select Drawn. Then click the Green checkmark to close the spreadsheet and return to the surface data spreadsheet. You will see an F on the special button for row 3.

Repeat the above sequence for the special button in row 4.



This completes the basic data entry for your lens. At this point, your spreadsheet should look like the following.



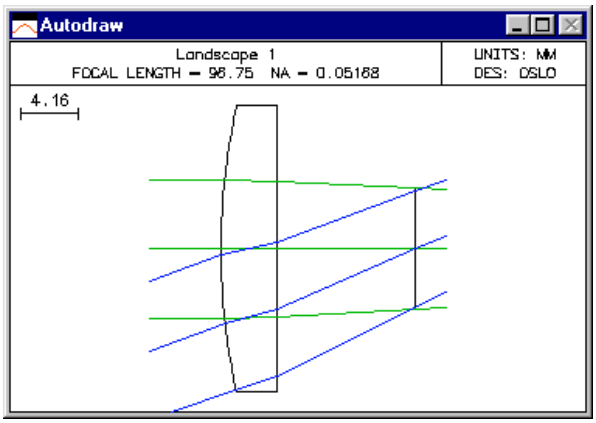
Check to make sure you have the same Efl (focal length), and that the buttons have the same labels. The S on the thickness button means that the value was determined by a solve (the axial ray height solve). The S on the aperture buttons also means that the apertures were determined by solves; this is the default action for apertures and is based on the entrance beam radius and the field angle.

Click the Green check to close the spreadsheet and save the lens by clicking the Save lens toolbar button. You have already indicated the file name on the opening dialog box. The lens will be

saved in the current lens directory, which is normally your private directory. If you want to confirm where the lens is saved, you can enter the command *shp lfil* (*show_preference lensfile*) in the command line. The file name will appear in the message area.

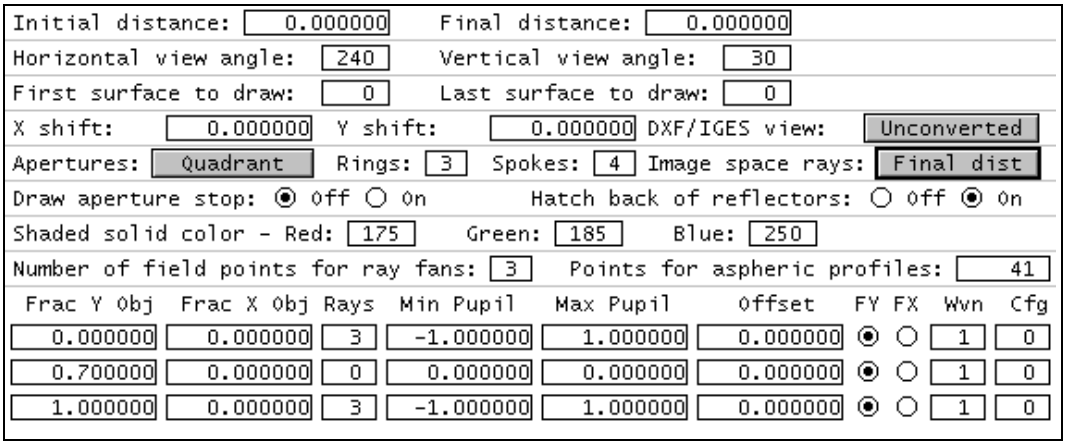
Lens Drawing

Click the DRAW OFF button in the fixed area of the lens spreadsheet. A window labeled Autodraw will appear showing the lens that you have entered. If you click on a cell in the spreadsheet, you will see that the surface corresponding to the cell becomes dotted. If you were to change any data for the lens, the picture would be updated automatically.

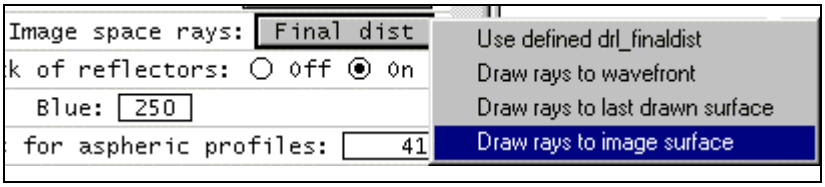


The Autodraw window is a special window that does not have the full functionality of a regular graphics window, but is updated automatically, which is not the case for ordinary graphics windows. The lens picture that is show is drawn according to the standard lens conditions used for all lens drawings. In order to illustrate this exercise better, we are going to change one of the operating conditions so that the ray trajectories are drawn all the way to the image surface.

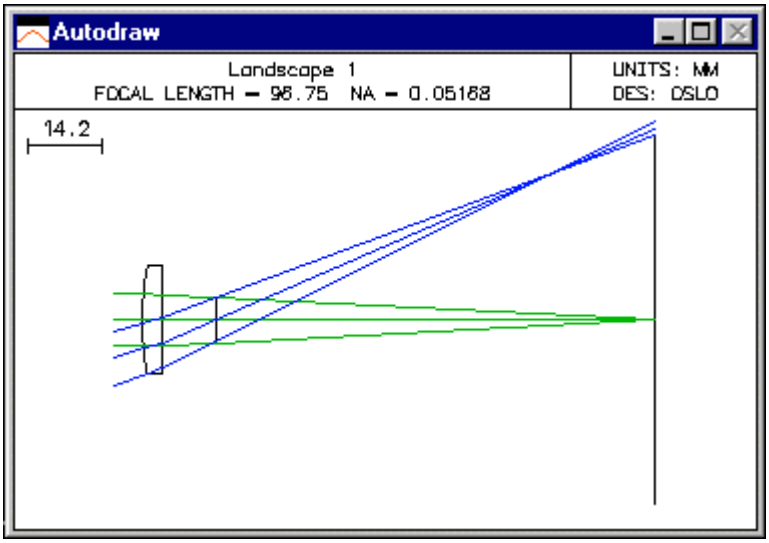
From the Lens menu on the main tool bar, select the bottom item, Lens Drawing Conditions. This will cause the following spreadsheet to cover the lens spreadsheet:



The item to change on this spreadsheet is called Image space rays: You should change it from Final dist to Draw to image surface.



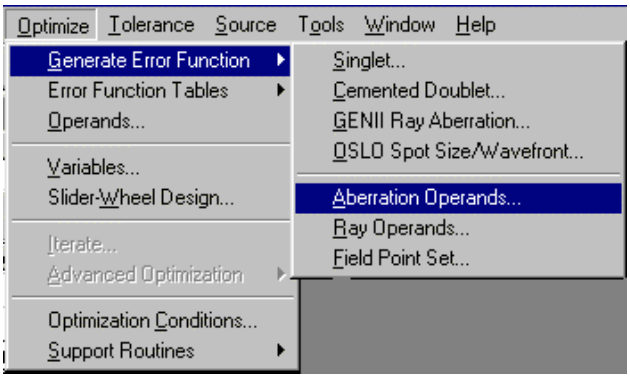
After making the modification, click the Green check to close the spreadsheet. The lens spreadsheet will reappear, and the Autodraw window will be updated as follows.



Now you can see the details of image formation by the lens. Obviously the off-axis image is not so good. The main aberration that you see is field curvature; the off-axis image is not formed on a plane surface perpendicular to the axis, but rather on a curved surface. Unfortunately, there is not too much that can be done about this.

Optimization

In order to study the imaging of this lens in more detail, we will set up a small optimization task. We will define an error function that makes the focal length exactly 100mm, and also eliminates the Seidel coma from the image. To do this, click the optimize button on the main menu, then navigate to the generate error function >> Aberration Operands item. This will pop up the spreadsheet shown on the following page.



OP	MODE	WGT	NAME	DEFINITION
1	Min	1.000000	PY	0CM1
2	Min	1.000000	PU	0CM2
3	Min	1.000000	PYC	0CM3
4	Min	1.000000	PUC	0CM4
5	Min	1.000000	PAC	0CM5
6	Min	1.000000	PLC	0CM6
7	Min	1.000000	SAC	0CM7
8	Min	1.000000	SLC	0CM8
9	Min	1.000000	SA3	0CM9
10	Min	1.000000	CMA3	0CM10
11	Min	1.000000	AST3	0CM11
12	Min	1.000000	PTZ3	0CM12
13	Min	1.000000	DIS3	0CM13
14	Min	1.000000	SA5	0CM14
15	Min	1.000000	CMA5	0CM15
16	Min	1.000000	AST5	0CM16
17	Min	1.000000	PTZ5	0CM17
18	Min	1.000000	DIS5	0CM18
19	Min	1.000000	SA7	0CM19
20	Min	1.000000	TOTAL_SPH	0CM20
21	Min	1.000000	EFL	0CM21

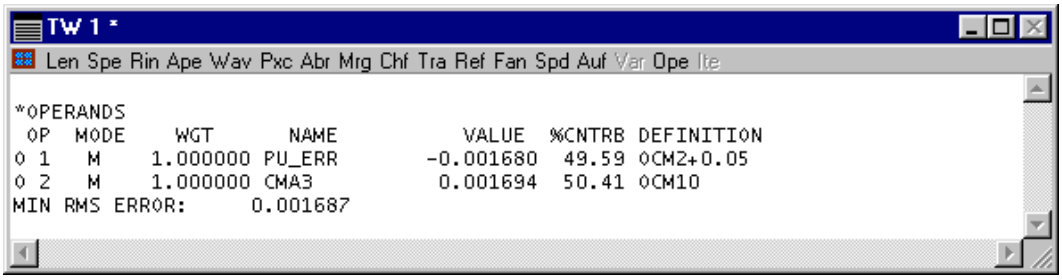
At his point, we need to decide which of the terms will be used for the error function. The initial spreadsheet lists all of the possibilities, and you must delete the ones you don't want. In this case, the operands to keep are PU and CMA3. All the other should be deleted. To do this, click and drag from row buttons 11-21, then press the delete key. Next click and drag from row buttons 3-9, then press the delete key. Finally, click row button 1 and press the delete key. The spreadsheet should now appear as follows.

OP	MODE	WGT	NAME	DEFINITION
1	Min	1.000000	PU	0CM2
2	Min	1.000000	CMA3	0CM10

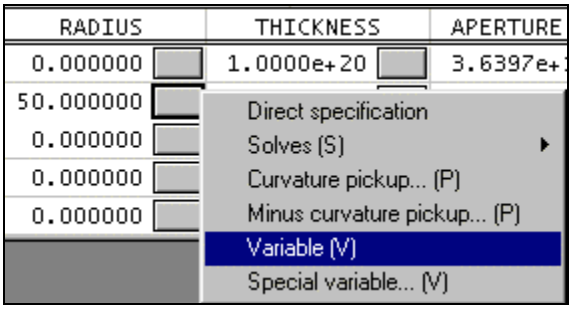
Next, we need to modify the definition of the first operand, PU. In OSLO, all operands are targeted to zero. That is, the optimization algorithm will minimize all of its individual terms. The PU operand is the axial ray slope leaving the lens. We don't want this to be zero, but rather the value that will make the lens have an f-number of 10, which was prescribed in the initial data. Since the f-number is -1/2*PU (when the object is at infinity) the desired value for PU is -.05. We accommodate this by modifying the definition of the first operand to be OCM2+0.05, and we will also change the name to PU_ERR. Click on the NAME and DEFINITION cells and enter the new data.

OP	MODE	WGT	NAME	DEFINITION
1	Min	1.000000	PU_ERR	0CM2+0.05
2	Min	1.000000	CMA3	0CM10

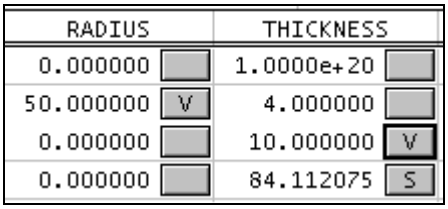
Now click the Green check to close the spreadsheet, and click the Ope button in the text window. The current operand values, together with the current value of the error function, will be shown.



Now the operands and the error functions are defined. If the error function is minimized to zero, we will have a f/10 lens with a focal length of 100, since the focal length is -PU*EBR (EBR is the entrace beam radius), and the coma will be zero. We need to specify the variables that are to be used to achieve this. The ones that we will use are the curvature of the first surface (CV 1), and the distance between the lens and the aperture stop (TH 2). In the lens spreadsheet, click on the button in the Radius of Curvature cell for row 1, and select Variable from the options list.

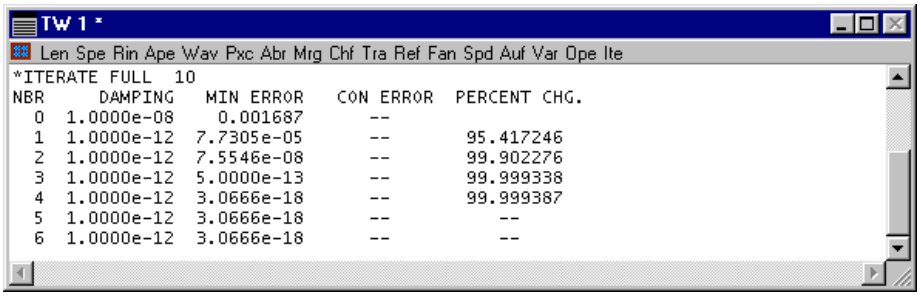


Repeat the same procedure for using the button in the Thickness cell for row 2. The buttons should now have V (variable) on them.



Assuming that they do, you should now be ready to optimize the lens. Before doing this, close the lens spreadsheet (Green check) and immediately re-open it. The reason for doing this is that OSLO gives you the capability, by canceling (red X) from a spreadsheet, to revert to the system that existed when the spreadsheet was opened. By closing and opening the spreadsheet in this way, you establish a base system to revert to.

To optimize, click the Ite button in the text window (you may have noticed that this button was disabled until you entered both operands and variables). The text window should show the following output. You see that the error function has been reduced to essentially zero, indicating that the variables were appropriate for solving the problem. The lens spreadsheet shows their current values.



Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: Landscape 1 | Zoom 1 of 1 | Efl 100.000000

Ent beam radius 5.000000 | Field angle 20.000000 | Primary wavln 0.587560

SRF	RADIUS		THICKNESS		APERTURE RADIUS		GLASS		SPECIAL
OBJ	0.000000		1.0000e+20		3.6397e+19		AIR		
1	51.680011	V	4.000000		9.856783	S	BK7	C	
2	0.000000		9.135802	V	8.637010	S	AIR		
AST	0.000000		88.227067	S	4.411353	AS	AIR		F
IMS	0.000000		0.000000		36.397023	S			F

You can see in the lens spreadsheet that the Efl is exactly 100, as it must be in order to correct the operands. Click the Abr button in the text window, and you see all the aberration data for the lens, showing that PU = -0.05, and that the Seidel coma CMA3 is zero, as requested.

TW 1 *

Len Spe Rin Ape Wav Pxc Abr Mrg Chf Tra Ref Fan Spd Auf Var Ope Ite

```
*PARAXIAL TRACE
 SRF       PY           PU          PI          PYC         PUC         PIC
  4        --       -0.050000   -0.050000   36.397023    0.412538    0.412538

*CHROMATIC ABERRATIONS
 SRF       PAC          SAC         PLC         SLC
 SUM    -0.077222    -0.053466    0.069914    0.048406

*SEIDEL ABERRATIONS
 SRF       SA3          CMA3        AST3        PTZ3        DIS3
 SUM    -0.013612  4.3368e-18    -0.313405   -0.218345    1.101908

*FIFTH-ORDER ABERRATIONS
 SRF       SA5          CMA5        AST5        PTZ5        DIS5        SA7
 SUM -9.6466e-05 -5.9253e-05    -0.020638    0.017274    0.070480 -7.7081e-07
```

At this point, the initial task has been completed. Close the spreadsheet (Green check) and save the lens. This will become the base system for the next phase of the exercise, which shows how to use OSLO's slider window to study the effects of changing other lens parameters. You might want to save another copy of this system under a different file name (e.g. landscape_bkp.len) so you can always return to a defined starting point, even if the current lens data is somehow corrupted.

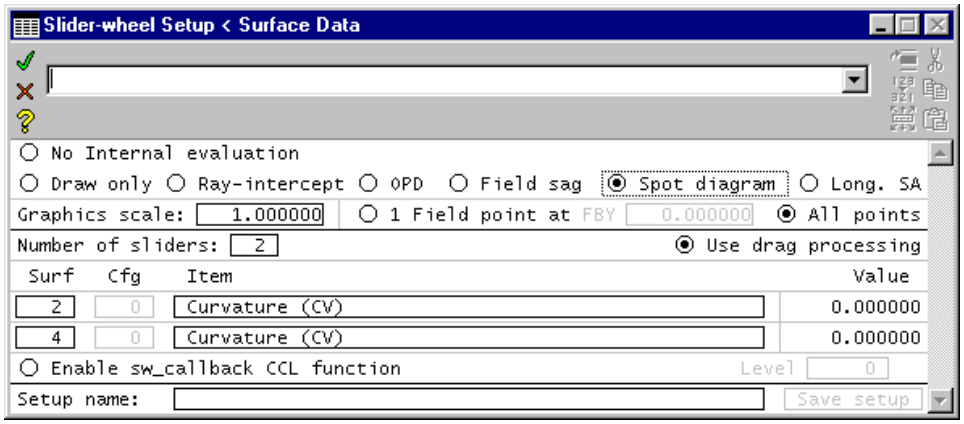
Slider-wheel design

A singlet such as the landscape lens doesn't have very many degrees of freedom, but in this exercise, we have constrained the back surface of the lens to be plane. Actually, this should not be called a landscape lens; so far all we have done is to find the position of the aperture stop that removes the coma for a convex-plano lens. A landscape lens is normally a meniscus form. Next we show how to use OSLO's slider-wheel window to find the optimum form. We also show the effects of using a curved image surface.

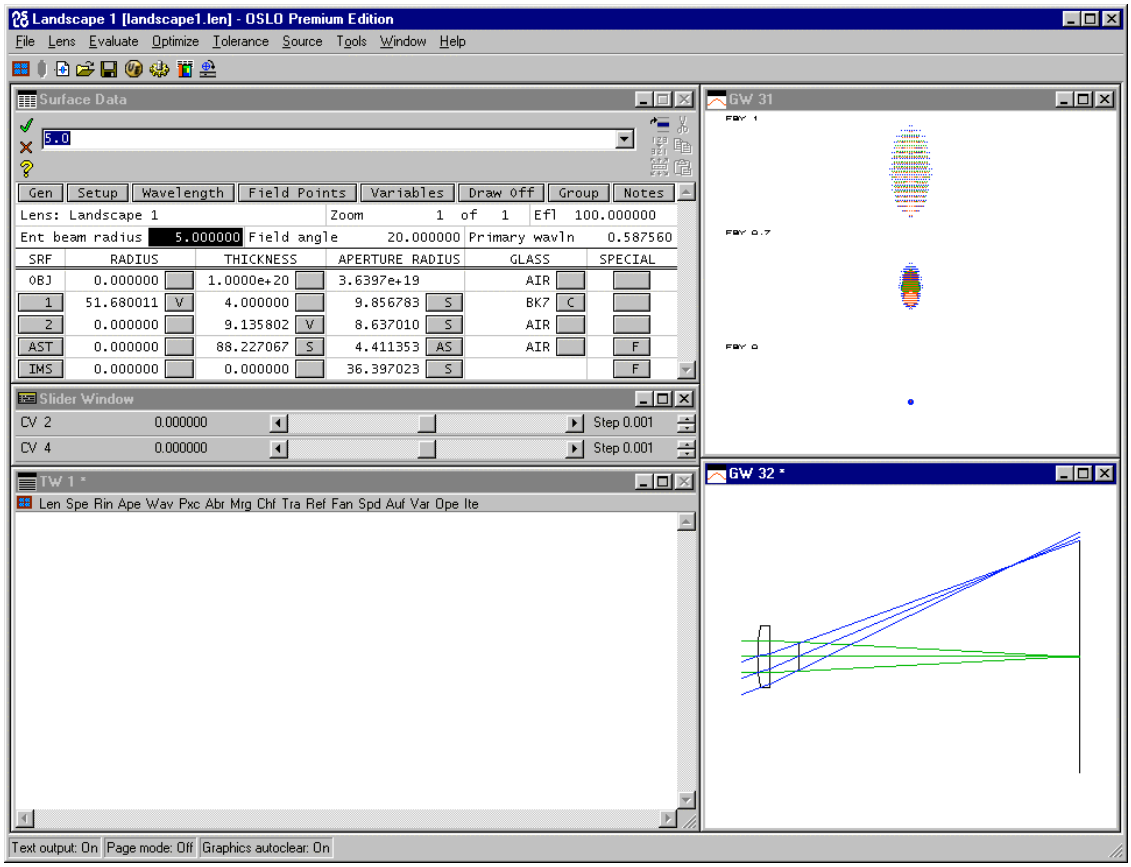
First, we set up a basic slider-wheel window, attaching the curvatures of surfaces 2 and 4 to sliders. Later, we show how to use the full power of OSLO's optimization callback to continuously optimize the system while dragging a slider or rotating the mouse wheel.

With the landscape lens as set up in the preceding portion of this exercise, click the slider-wheel tool in the main tool bar to open the slider-wheel spreadsheet. This spreadsheet is used to set up slider-wheel windows.

Initially, set the options shown in the figure below. That is, click Spot diagram, set the Graphics scale to 1.0, and select All points. Leave the number of sliders at the default (2), enter 2 and 4 in the Surfs column, and set the parameters to CV for both sliders.

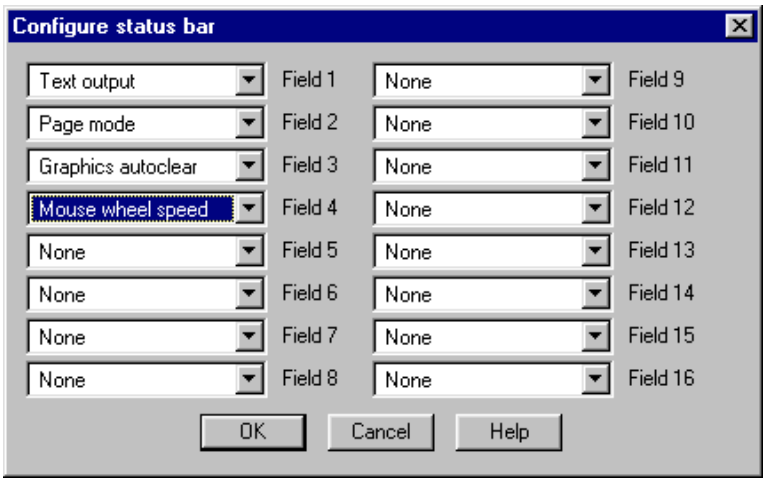


After you have entered the data in the slider-wheel spreadsheet, close the spreadsheet. You should see a slider-wheel window and two additional graphics windows (31 & 32) on your screen. If these windows have toolbars, remove them by left-clicking to select a window, then right-clicking in the window and choosing the Remove Toolbar item. Since the slider-wheel maintains its own graphics windows, you should minimize the normal graphics output window (GW1). Next, use the Tile windows command on the main window toolbar to tile the windows. Your overall screen should look similar to the one below.



The slider-wheel window appears below the spreadsheet, with the two sliders that we set up. You can manipulate sliders by dragging or clicking as you would with a normal scroll bar (the sliders are in fact scroll bars), or by using the mouse wheel. The mouse wheel is usually preferred (if it is installed and properly supported on your computer) but you may need to change its settings.To use the mouse wheel with a slider, place the pointer *anywhere* in a slider and rotate the wheel.

To change the settings of your mouse wheel, you need to first find out what they are at present. To do this, double click on the status bar in the main OSLO window. This will pop up a dialog box as shown below. Navigate to the first empty field, click on the combo box, and select Mouse wheel speed, as shown.



Click OK to dismiss the box. You should see an additional field in the status bar, as shown. OSLO supports two mouse-wheel speeds: fast and slow. Slow means one event per wheel notch; fast means several events per wheel notch (however many you set up in the mouse dialog in your computer's control panel). You change between fast and slow by clicking the mouse button itself. Generally in OSLO, you will prefer Slow.

If you experiment, you will see that the slider-wheel window can be dragged anywhere on the display, even outside the OSLO window. Also, you will see that you can drag the window width down to the point where the slider itself completely disappears. If you use the mouse wheel, this setting may be preferable because it minimizes the use of screen space while still providing full functionality. You will see that the step size in the parameter taken per click can be adjusted using the Step control at the right-hand end of the window.

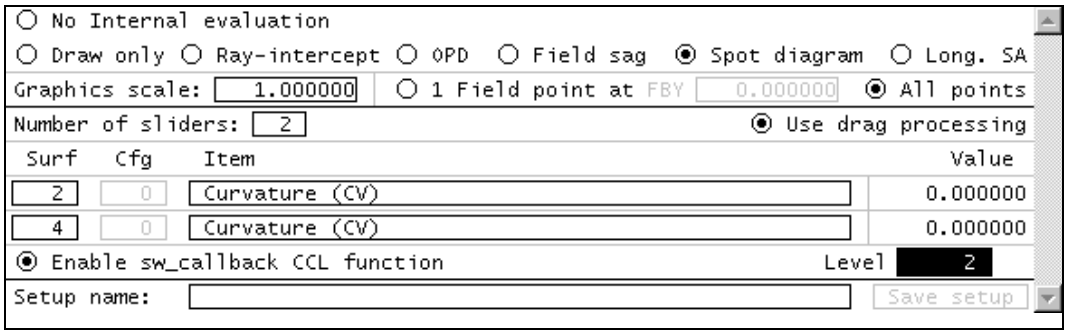
For now, it is best to leave the slider window where the Tile windows command puts it, with its default settings. To reset these, enter the original values (0) of CV[2] and CV[4] in the lens spreadsheet, then click the slider-window tool in the main toolbar and close the slider-wheel setup spreadsheet immediately to update the slider-wheel window (note that the settings for the analysis options are not changed).

Returning to optics, you see that the spot diagrams show what the image looks like at the center, zone, and edge of the field. The elliptical shape is characteristic of a system with astigmatism, but no coma. Now you can drag the sliders or rotate your mouse wheel to see what happens. Actually, if you do not use the mouse wheel, you would be better advised to click on a button at the end of a slider rather than drag, so you can move in small enough steps.

What you see is pretty much what you would expect. If you make CV[2] negative, the lens becomes positive and the focal length gets smaller. The image quality changes, since the system is no longer free of coma. If you change the curvature too much (< -0.033) the ray trace will fail. If this happens, click OK to dismiss the error box, then move the slider back towards the center. If you make CV[2] positive, the lens becomes first a meniscus with a long focal length, but eventually the lens becomes negative; the beam diverges and the display blows up. After you have experimented with CV[2], set it back to zero.

When you change CV[4] (the curvature of the image surface) you see that by making it negative, you can improve the size of the image off-axis, and in fact you can find a position where there is a line focus in the horizontal direction (tangential focus), or the vertical direction (sagittal focus). This is indicative of a system with no coma. If you set CV[2] to a value where the system has substantial coma (try -.020) you will not be able to achieve this. This shows what designers mean when they say that you can't focus out coma.

The slider-wheel analysis you have observed so far is perhaps interesting, but the real power of sliders in OSLO comes when you allow the program to re-optimize the system as you drag a slider. To do this, reset the curvatures to zero, then re-open the slider spreadsheet, set *Enable sw_callback CCL function* to On, and set the level to 2.

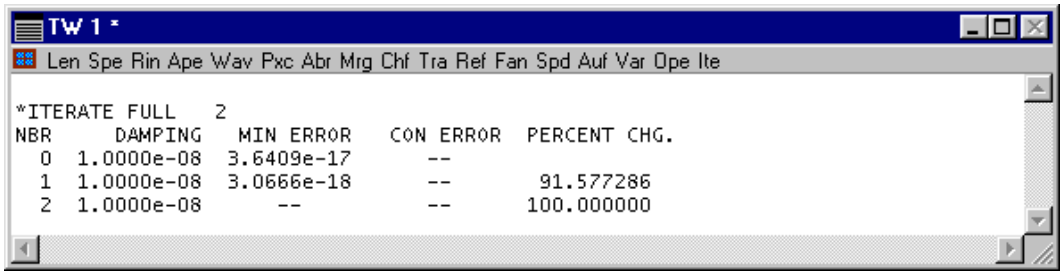


A callback function is a command that gets executed whenever some event happens in the program. Callback functions are the essence of a windows program, and OSLO has several of them, many written in CCL (such as sw_callback). Generally, these are not complicated routines written by computer programmers, but simple actions set up by users. The default sw_callback routine is as follows.

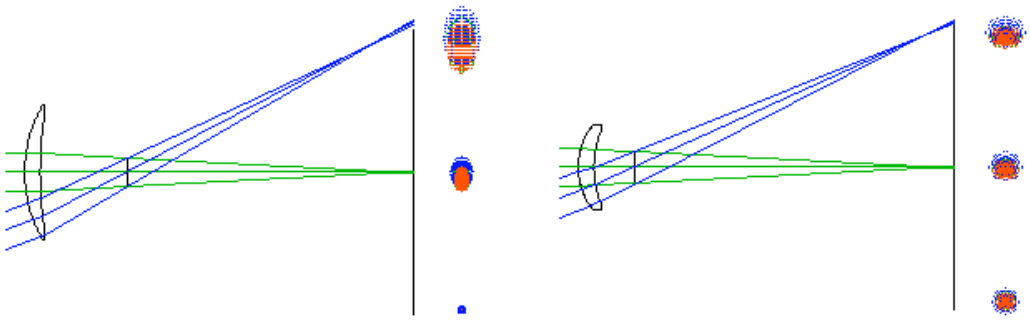
```
Cmd Sw_callback(int cblevel, int item, int srf)
{
        if (cblevel)
                ite cblevel;
}
```

When *Enable sw_callback CCL function* is turned on, this routine is executed every time a graphic slider is moved, or the mouse-wheel is rotated when the pointer is in the slider-wheel window. The arguments of this function are *cblevel*, the value entered in the Level field in the slider-wheel spreadsheet; *item*, the type of variable (CV = 1, TH = 3, etc.) that was changed, and *srf*, the number of the surface associated with that variable. These arguments can be used to produce whatever action you want. In the present case, the function only looks to see if *cblevel* is non-zero, and if it is, the function carries out that many design iterations.

To see what this does, close the slider window. You should not see any change in the system, because it was already optimized, but you will see text out similar to the following. As you move a slider, you will see the numbers flash in the text output window as the system is re-optimized.



In the graphics windows, you will see a quite different scenario from before. To begin, drag the CV[2] slider to make it more positive. This makes the lens into a meniscus, but now the focal length is held constant by virtue of the first curvature's being continually re-optimized. Moreover, the aperture stop surface moves as the coma is continually re-optimized to zero. The image surface is held in the paraxial image plane by the height solve on surface 3, and the diameter of the lens is adjusted by the aperture solves so that the beam from the edge of the field of view can pass through the system. The effect of all of this is that the stop initially moves away from the lens as it becomes a meniscus, but as the bending becomes larger, the stop shift reverses and the aperture stop moves back towards the lens. The following shows what you should observe when the surface 2 curvature has values 0.01, and 0.04 (n.b. the images pertain to center-to-edge, not to bottom-to-top; the bottom spot is the on-axis image).



Of course, the spherical aberration of the system is not constant, and the improvement in image quality at the edge of the field comes at the expense of a poorer on-axis image. This leads to a consideration of what might be accomplished by using an aspheric lens, but that is beyond the scope of this introductory exercise, whose purpose is to introduce OSLO data entry and the general steps for using the slider-wheel window.

Chapter 2 Fundamentals

In this chapter we consider some fundamental properties of light and optical materials. The reason for doing so is not to teach the subject, but to review some general physics that is relevant to optical design. Optical design has been taught in many ways, ranging from simple numerical computation to complex geometrical analysis. The approach used here is based on classical electromagnetic theory, and emphasizes understanding of both the physics and the methods used for solving practical problems.

Waves, Rays, and Beams

Usually, the analysis of optical systems assumes that light propagation can be described using wavefronts, or rays, which are defined to be vectors perpendicular to wavefronts. Except in focal regions and near aperture boundaries, most optical imaging problems can be solved using a simple geometrical wavefront or ray model. To investigate the details of an image, Fourier analysis of wavefronts is used to account for diffraction effects.

Rays and wavefronts have shortcomings, however. A ray is a line, and a wavefront is a surface. Neither has a finite width or thickness, and the intuitive notion that rays or waves carry energy is problematical. For a normal (incoherent) source, it is necessary to use a stochastic approach to describe energy propagation, summing the contributions of several rays or wavefronts. On the other hand, laser (coherent) light generally appears as a beam. A single-mode laser beam can propagate energy, and the diameter and divergence of the beam can be computed using diffraction theory. Beams provide a smooth transition between geometrical and physical optics, and we consider them as fundamental entities like rays and wavefronts.

Maxwell's equations

We begin with Maxwell's equations. Maxwell's equations for electromagnetic fields describe the relationships between the basic fields (the electric field **E** and the magnetic induction **B**) and the derived fields (the electric displacement **D** and the magnetic field **H**), taking into account the charge distribution ρ and electric current density **J**. The derived fields are a result of the interaction of the basic field with matter. Using the symbol *t* for time, Maxwell's equations in mks units are

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \qquad (2.1)$$

$$\nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{J} \qquad (2.2)$$

$$\nabla \bullet \mathbf{D} = \rho \qquad (2.3)$$

$$\nabla \bullet \mathbf{B} = 0 \qquad (2.4)$$

The interactions of the field with matter are described by *material equations*. Fortunately, for most situations of interest in optical design, the material equations are linear (i.e., the conditions are static and isotropic). In this case, the material equations are just constant relations, involving the dielectric constant (permittivity) ε, the permeability μ, and the conductivity σ. These three quantities are constant at a given point in space, but are functions of frequency (wavelength). For isotropic materials, the permittivity is a scalar quantity (although a function of wavelength). By contrast, the permittivity of an anisotropic medium such as a crystal must be described by a tensor. In other words, $\boldsymbol{\varepsilon}$ is a 3 x 3 matrix that relates the components of **E** to the components of **D**. In the present discussion, we will consider only the case where the permittivity is a scalar quantity.

$$\mathbf{D} = \varepsilon\mathbf{E} \tag{2.5}$$

$$\mathbf{B} = \mu\mathbf{H} \tag{2.6}$$

$$\mathbf{J} = \sigma\mathbf{E} \tag{2.7}$$

These material equations can be combined with Maxwell's equations to derive a *wave equation* for the electric field:.

$$\nabla^2\mathbf{E} = \mu\varepsilon\frac{\partial^2\mathbf{E}}{\partial t^2} + \mu\sigma\frac{\partial\mathbf{E}}{\partial t} \tag{2.8}$$

An analogous equation can be derived for the magnetic field. For the dielectric materials used in optical systems, the conductivity σ is zero, so Eq. (2.8) simplifies to

$$\nabla^2\mathbf{E} = \mu\varepsilon\frac{\partial^2\mathbf{E}}{\partial t^2} \tag{2.9}$$

Equation (2.9) is in the standard form for a wave propagating with a velocity v, where

$$v = \frac{1}{\sqrt{\mu\varepsilon}} \tag{2.10}$$

For a monochromatic wave of frequency ν (and angular frequency $\omega = 2\pi\nu$), the electric field has the form

$$\mathbf{E}(\mathbf{r},t) = \mathbf{E}_r(\mathbf{r})\exp(\pm i\omega t) \tag{2.11}$$

where **r** is the vector distance from some origin to the observation point. A monochromatic wave has the shortcoming that it carries no power, so for thermal sources one must consider a finite frequency band. Again, the laser differs from thermal sources because stimulated emission adds power to the mode that stimulated it, causing the beam to have temporal coherence. Of course, a strictly monochromatic beam would need to have an infinite duration, so even laser beams have some finite bandwidth. We usually consider the temporal properties of light to be described by the product of two terms: a rapidly varying term like the exponential in Eq. (2.11) , and a slowly varying term incorporated in $\mathbf{E}_r$. In practice, for imaging systems, the time-varying terms are neglected, and the wave equation takes the form of the Helmholtz equation

$$\nabla^2\mathbf{E} + k^2\mathbf{E} = 0 \tag{2.12}$$

where $k = \sqrt{\mu\varepsilon}\omega = 2\pi/\lambda$. For many optical systems, the propagation space is isotropic, so it suffices to consider only one (scalar) component of the electric field, often called the optical disturbance, written as $u(x, y, z)$. All scalar wave phenomena require finding solutions $u(x, y, z)$ to the scalar wave equation:

$$\nabla^2 u(x,y,z) + k^2 u(x,y,z) = 0 \tag{2.13}$$

Well-known solutions to Eq. (2.13) include the spherical wave

$$u_{spherical}(x,y,z) = \frac{A_s \exp\left(\pm ik\sqrt{x^2+y^2+z^2}\right)}{\sqrt{x^2+y^2+z^2}} \tag{2.14}$$

and the plane wave (taken here to be traveling along the z-axis)

$$u_{plane}(x,y,z) = A_p \exp(\pm ikz) \tag{2.15}$$

(In Eqs. (2.14) and (2.15), A_s and A_p are constants.)

Laser beams have a well-defined propagation direction. Thus, assuming that the beam is traveling in the z direction and motivated by the plane wave solution to the wave equation, we look for a solution of the form

$$u(x,y,z) = \psi(x,y,z)\exp(-ikz) \tag{2.16}$$

where $\psi(x, y, z)$ is a function that describes the amplitude and phase differences between the beam solution and a plane wave. That is, we assume that the basic oscillatory behavior of the beam is given by the exponential term, with $\psi(x, y, z)$ being a relatively slowly varying amplitude. If we substitute Eq. (2.16) into Eq. (2.13), we find that $\psi(x, y, z)$ must satisfy

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}+\frac{\partial^2\psi}{\partial z^2}-2ik\frac{\partial\psi}{\partial z}=0 \tag{2.17}$$

We now assume that the beam is well confined to a region near the z-axis. Then, since $\psi(x, y, z)$ varies slowly with respect to z, we can ignore the second derivative of ψ with respect to z in Eq. (2.17). So, we find that $\psi(x, y, z)$ must satisfy the so-called paraxial (or parabolic) wave equation:

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}-2ik\frac{\partial\psi}{\partial z}=0 \tag{2.18}$$

There are many solutions to Eq. (2.18). Depending on the geometry considered, one can find solutions (called *modes*) in terms of products of orthogonal polynomials and Gaussian functions.

$$u_{beam}(x,y,z)=\frac{w_0}{w(z)}H_m\left(\frac{\sqrt{2}}{w}x\right)H_n\left(\frac{\sqrt{2}}{w}y\right)\exp\left[-i\left(kz-\Phi(z)\right)-r^2\left(\frac{1}{w^2(z)}+\frac{ik}{2R(z)}\right)\right] \tag{2.19}$$

where

$$r^2 = x^2 + y^2 \tag{2.20}$$

$$\Phi(z)=\tan^{-1}\left(\frac{\lambda z}{\pi w_0^2}\right)$$

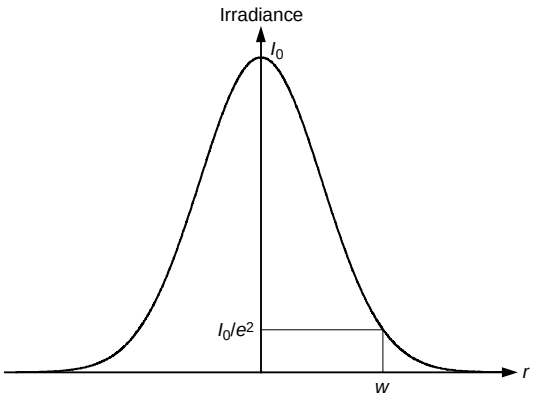
The functions H_m and H_n are Hermite polynomials, given by

$$H_0(\xi)=1 \tag{2.21}$$

$$H_1(\xi)=2\xi$$

$$H_n(\xi)=(-1)^n e^{\xi^2}\frac{d^n}{d\xi^n}e^{-\xi^2}$$

The lowest-order (TEM_{00}) mode is a simple Gaussian function. In Eq. (2.19), w and R are functions of z and w_0 is a constant. We can see from Eq. (2.19) that w describes the radial amplitude distribution in the beam and R is the radius of curvature of the wavefront. At a distance from the axis of $r = w$, the amplitude of the beam is equal to $1/e$ of its on-axis value. Since the irradiance is equal to the squared modulus of the field, the irradiance at $r = w$ is equal to $1/e^2$ of its axial value, as illustrated in the figure below. It is customary to call w the spot size of the Gaussian beam. Note that the spot size is a measure of the radius of the beam, not its diameter.



Gaussian beam propagation

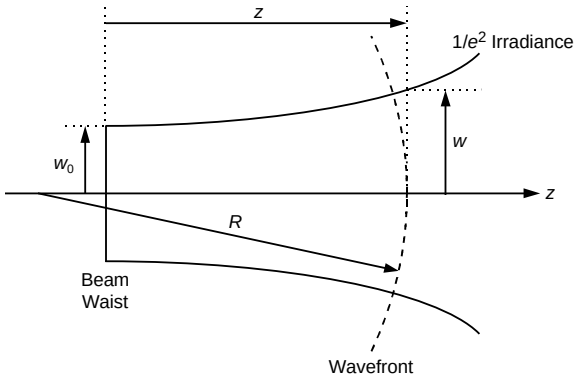
The Gaussian beam described by Eq. (2.19) has a minimum spot size at the same axial location at which the wavefront is planar. The formulae for $w(z)$ and $R(z)$ take on simple forms if we let $z = 0$ correspond to this point of minimum spot size. Then, the explicit forms of the variation of the spot size w and wavefront radius of curvature R with z are

$$w^2(z) = w_0^2\left[1+\left(\frac{\lambda z}{\pi w_0^2}\right)^2\right] \tag{2.22}$$

and

$$R(z) = z\left[1+\left(\frac{\pi w_0^2}{\lambda z}\right)^2\right] \tag{2.23}$$

Examination of Eq. (2.22) confirms that the spot size takes on its minimum value at $z = 0$ and this minimum value is equal to the constant w_0, which is called the beam waist. Also, we find from Eq. (2.23) that the wavefront radius of curvature becomes infinite at the beam waist, i.e., the wavefront is planar. Note that the convention of measuring z from the waist to the observation point implies that z and $R(z)$ have the same sign, as can be seen from Eq. (2.23). In the figure below, both z and R are positive as shown. This sign convention for radius of curvature is fairly standard in the Gaussian beam and laser literature, but is opposite the convention used by OSLO for the sign of radius of curvature. OSLO uses the above equations when computing the propagation of a Gaussian beam, but, for consistency, reports the wavefront radius of curvature using the OSLO sign convention. Thus, you should interpret waist distances (z) reported by OSLO as being measured *from* the observation point *to* the waist. For example, in reporting the beam parameters for the beam in the figure, OSLO would display both z and R as negative.



A Gaussian beam is completely specified by any two of the four parameters w, w_0, z, and R (in addition to the wavelength λ). Given any two of the parameters, the other two can be computed

using Eqs. (2.22) and (2.23), or obvious rearrangements of these equations. There are several other parameters of the beam that are often useful in analysis. The $1/e^2$ irradiance far-field divergence angle θ, measured from the z-axis, is given by

$$\theta = \tan^{-1}\left(\frac{\lambda}{\pi w_0}\right) \cong \frac{\lambda}{\pi w_0} \tag{2.24}$$

The distance from the waist to the axial point of minimum wavefront radius of curvature is called the Rayleigh range z_R

$$z_R = \frac{\pi w_0^2}{\lambda} \tag{2.25}$$

The spot size at the Rayleigh range point is equal to $\sqrt{2}w_0$. The Rayleigh range can be used to define the length of the focal region, or collimation length, of a Gaussian beam. The value of the minimum wavefront radius of curvature is the confocal radius b_0:

$$b_0 = \frac{2\pi w_0^2}{\lambda} = 2z_R \tag{2.26}$$

The solutions of Eqs. (2.22) and (2.23) provide a great deal of insight into the process of focusing a light beam. Imagine a convergent beam propagating from left to right. Well before it reaches the focus point, the radius of curvature of the wavefront is centered on a well-defined point at the expected focus ($z = 0$). As the wavefront approaches the focal region, the focus point appears to recede. When the wavefront is one Rayleigh range in front of the original focus, the apparent focus is one Rayleigh range behind it. Inside this point, the apparent focus recedes rapidly, reaching infinity at $z = 0$. The wavefront then becomes divergent with a radius of curvature symmetrical in magnitude to a corresponding point on the other side of the original focus point.

Regardless of the position along the z-axis, the beam always has a Gaussian intensity profile. The profile is not at all like the beam from a uniform “top-hat” beam, whose intensity profile changes as it passes through focus. Nevertheless, the salient aspects of Gaussian beam focusing are not different from any other beam, including a shift of focus towards the lens. Prior to the invention of the laser, these effects were not commonly understood, because they are only easy to see in weakly focused beams.

Propagation Circles

Using the Rayleigh range as the parameter characterizing a Gaussian beam, it is possible to determine the spot size and radius of curvature at any point using a geometrical construction called the propagation-circle method. The quantities determined in the propagation-circle method are the radius of curvature of the wavefront and a beam parameter b, related to the spot size by

$$b = \frac{\pi w^2}{\lambda} \tag{2.27}$$

Using this in Eq. (2.22) and Eq. (2.25), we find an equation for the beam parameter:

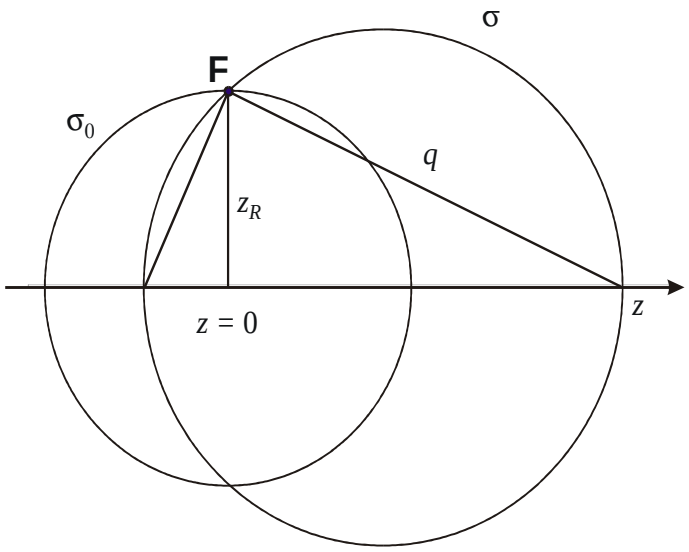
$$b = \frac{z_R{}^2 + z^2}{z_R} \tag{2.28}$$

We may rewrite Eq. (2.23) as

$$R = \frac{z_R{}^2 + z^2}{z} \tag{2.29}$$

These last two equations form the basis for the propagation-circle method. Consider a Gaussian beam with a Rayleigh range z_R, as shown above. First construct a circle σ_0 centered on the point $z = 0$, with a radius equal to the Rayleigh range.

Construct a vertical line through the point $z = 0$, and denote the intersection of the line with the circle σ_0 by **F**. This point is called the complex focal point of the beam. By construction, the distance from the beam axis to **F** is equal to the Rayleigh range.



Now construct a circle σ, centered on the axis and passing through the complex focal point **F**. The radius of curvature of a Gaussian beam wavefront at both points where the circle σ intersects the axis is equal to the *diameter* of the circle σ. To see this, note that

$$q^2 = z_R{}^2 + z^2 \tag{2.30}$$

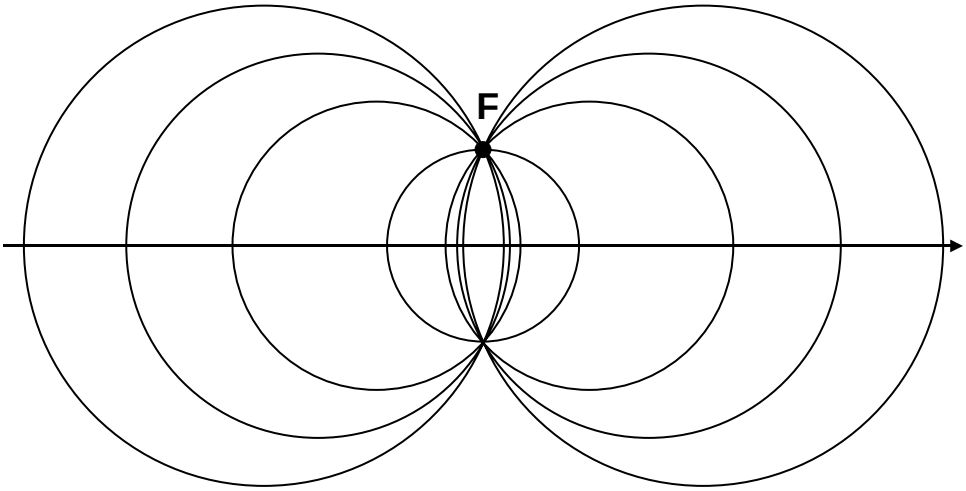
and

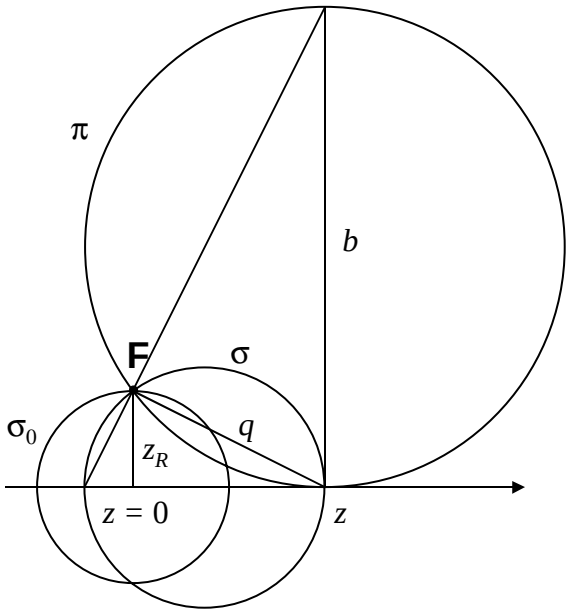
$$\frac{R}{q} = \frac{q}{z} \tag{2.31}$$

Therefore,

$$R = \frac{z_R{}^2 + z^2}{z} \tag{2.32}$$

which is simply Eq. (2.29) again. It thus follows that at any point z on the axis of a Gaussian beam one can construct a circle, which we shall call a σ circle, whose diameter is equal to the radius of curvature of the wavefront of the TEM_{00} mode at the point z. The σ circle is defined by the requirement that it pass through the point z and the complex focal point F. The figure below shows a graphical determination of the wavefronts of a beam passing through a focal region using σ circles. Note that the wavefront radius is equal to the *diameter* of the σ circles.





A different geometrical construction is used to find the beam parameter as a function of distance along the axis from the beam waist. Construct two circles σ_0 and σ as before, and determine the complex focal point, as shown above. Now construct a circle π that passes through the complex focal point **F** and is tangent to the optical axis at the point z. The diameter of this circle is equal to the beam parameter b. To see this, note that

$$q^2 = z_R{}^2 + z^2 \tag{2.33}$$

and

$$\frac{Z_R}{q} = \frac{q}{b} \tag{2.34}$$

Thus

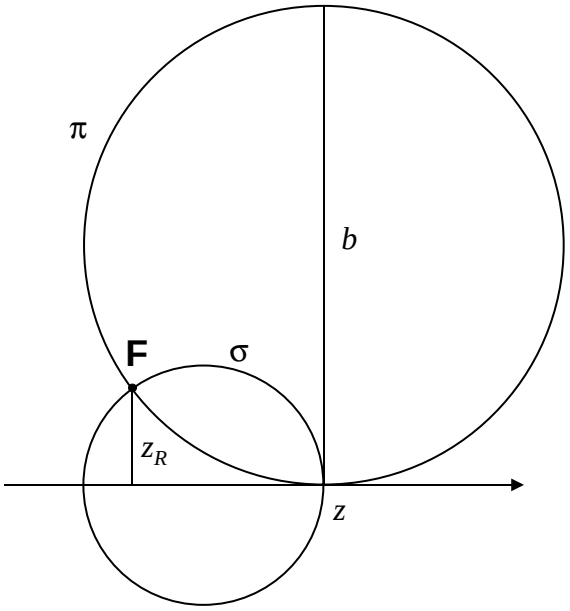
$$b = \frac{z_R^{\,2} + z^2}{z_R} \tag{2.35}$$

which is simply a restatement of Eq. (2.28). It follows that at any point z on the axis of a Gaussian beam one can construct a circle called a π circle, whose diameter is equal to the beam parameter b at the point z. The π circle is defined by the requirements that it pass through the complex focal point **F** and that it be tangent to the optical axis at the point z. The spot size w of the transverse mode at the distance z is determined from the beam parameter b by the relation

$$w = \sqrt{\frac{b\lambda}{\pi}} \tag{2.36}$$

The above constructions form the basis for the propagation-circle method. If we know the beam parameter and radius of curvature at one point, we can find the beam parameter and radius of curvature at any other point by constructing suitable σ and π circles as follows.

Suppose the beam parameter and radius of curvature have values b and R, at some point z. We associate with the beam parameter a circle π, tangent to the beam axis, whose diameter is equal to b, and we associate with the radius of curvature a circle σ whose diameter is equal to R, as shown below. According to the propagation-circle method, the circle σ and the circle π define the complex focal point F. Since the propagation of a Gaussian beam is completely determined by the location of its complex focal point, we can find the beam parameter and radius of curvature at any other point by constructing new σ and π circles that pass through F and the new point. That's all there is to it! Of course, underlying the propagation-circle method is the assumption that the beam is not truncated by an aperture, and that it propagates according to the paraxial wave equation.

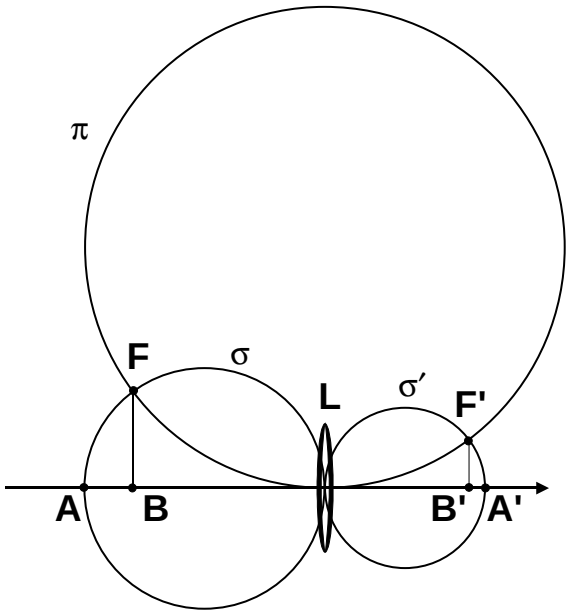


Assuming that we can neglect aberrations, it is easy to extend the concept of propagation circles to describe the propagation of a Gaussian beam through an optical system. We may begin by considering propagation through a thin lens. A beam of radius of curvature R_1 entering a thin lens of focal length f is transformed into a beam of radius of curvature R_2 leaving the thin lens, where R_1 and R_2 are related by the thin-lens equation,

$$\frac{1}{R_2} - \frac{1}{R_1} = \frac{1}{f} \tag{2.37}$$

If the lens is thin, the diameter of the beam is not changed in passing through it. Consider a Gaussian beam incident on a thin lens L. Let the incident beam be characterized by the complex focal point F as shown. The beam parameter and radius of curvature of the beam incident on the lens can then be characterized by the propagation circles σ and π as shown in the figure on the next page.

Since the lens will alter the radius of curvature of the beam according to Eq. (2.37), the beam emerging from the lens will be characterized by a new a circle, σ'. The circle σ' is defined by the requirement that it intersect the axis at the vertex of the lens and at the point A' shown below. The point A' is the geometrical image of the point A. Since a thin lens does not alter the beam parameter, the emergent beam will be characterized by the same π circle as the incident beam. The beam emerging from the thin lens L is thus characterized by the propagation circles σ' and π. The point where these circles intersect determines the complex focal point F' of the emergent beam. Once the new complex focal point is found, the propagation of the beam after it passes through the lens is completely determined.



We see from the figure that the minimum spot size of the emergent beam, or the "beam waist," is always closer to the lens than the geometrical focus. That is, the minimum spot size occurs at the point B', where the perpendicular from F_1 intersects the optical axis; whereas the geometrical focus is at the point A'. The minimum spot size of the emergent beam is

$$w_0 = \sqrt{\frac{z_R \lambda}{\pi}} \qquad (2.38)$$

where z_R is the length of the line B'F' (that is, z_R is the new Rayleigh range of the beam).

The propagation circle method shows when the geometrical-optics approximation provides a valid picture of image formation: the geometrical-optics approximation is valid when the complex focal point F' coincides with the geometrical focus at point A'. This will occur when the propagation circle π is large: that is, when the spot size of the beam on the lens is large, or when the wavelength is small compared to the beam diameter.

The propagation-circle method described above can also be used to determine the propagation of Gaussian beams through an interface between two materials having different refractive indices. In order to determine the propagation circle π_2, we must generalize the discussion given for the case of a thin lens. In particular, we have in general that the *optical length* of the beam parameter remains constant as the beam traverses the interface between two refracting media. It follows that

the beam parameter b_2 at the interface is related to the beam parameter b_1 at the interface as follows:

$$n_2 b_2 = n_1 b_1 \tag{2.39}$$

The propagation circle π_2 thus has a diameter that is n_1/n_2 times the diameter of π_1. The complex focal point F_2 is determined by the intersection of the propagation circles σ_2 and π_2; once F_2 is found, the propagation of the beam in medium 2 is completely determined.

Algebraic Methods

Although the propagation-circle method described in the preceding section is useful for visualizing the propagation of Gaussian beams through optical systems, for calculations it is desirable to work directly with the algebraic expressions for the beam properties. In particular, it is convenient to redefine the quantity q introduced before as a complex beam parameter:

$$\frac{1}{q} = \frac{1}{R} - i\frac{\lambda}{\pi w^2} \tag{2.40}$$

Defining the q parameter in this way has several important uses. First, the radial portion of the Gaussian beam can be written in the simple form of $\exp(-ikr^2/2q)$, which is the form of a spherical wave in the paraxial approximation. The difference here is that now q, which corresponds to the radius of curvature of the spherical wave, is complex. Second, the propagation laws for the Gaussian beam parameters [Eqs. (2.22) and (2.23)] can be combined into the single equation

$$q(z) = i\frac{\pi w_0^2}{\lambda} + z \tag{2.41}$$

Third, if we assume that the beam is smaller than any apertures in the optical system, we can neglect the effects of diffraction by finite apertures as the beam propagates along the axis of an *orthogonal* optical system. (Orthogonal means the propagation can be considered independently for the x and y coordinates.) A Gaussian beam remains Gaussian as it propagates through such a system and the output q parameter (q', in a medium of refractive index n') is related to the input q parameter (q, in a medium of refractive index n) by

$$\frac{q'}{n'} = \frac{A(q/n) + B}{C(q/n) + D} \tag{2.42}$$

where A, B, C, and D are the elements of the 2×2 paraxial matrices. Equation (2.42) is often referred to as the *ABCD* law. At this level of analysis, the only difference in propagating a Gaussian, rather than point source, beam through a system is the use of the complex parameter q rather than the purely real radius of curvature R. We don't need any additional information other than that which is necessary to compute the four Gaussian constants for the desired system or subsystem. For a Gaussian beam, the q parameter plays the same role in the *ABCD* law as the wavefront radius of curvature does for point sources. For this reason, q is sometimes referred to as the complex radius of curvature.

Note that the q parameter as defined in Eq. (2.40) uses the wavelength λ in the medium in which the beam is propagating. For some calculations, it is useful to introduce a reduced q parameter, $\hat{q}$, that is defined in terms of the standard wavelength $\lambda_0 = n\lambda$:

$$\frac{1}{\hat{q}} = \frac{n}{q} = \frac{n}{R} - i\frac{\lambda_0}{\pi w^2} \tag{2.43}$$

Then, the *ABCD* law takes on the somewhat simpler form

$$\hat{q}' = \frac{A\hat{q} + B}{C\hat{q} + D} \tag{2.44}$$

The Gaussian beam solution to the paraxial wave equation can be generalized to the case of an astigmatic beam with different spot sizes and wavefront radii of curvature in x and y. If we denote the complex beam parameters in the xz and yz planes by q_x and q_y, respectively, we can write a solution to Eq. (2.18) as

$$\psi(x, y, z) = (q_x q_y)^{-\frac{1}{2}} \exp\left[-i\frac{k}{2}\left(\frac{x^2}{q_x} + \frac{y^2}{q_y}\right)\right] \tag{2.45}$$

A beam of this form can be propagated through an orthogonal system by computing the Gaussian constants in the *xz* and *yz* planes and applying the *ABCD* law separately in each azimuth. It can further be shown that we can rotate the astigmatic beam of Eq. (2.45) by an angle ϕ around the *z* axis and the resulting expression is still a solution of Eq. (2.18):

$$\psi(x,y,z) = (q_x q_y)^{-\frac{1}{2}} \exp\left\{-i\frac{k}{2}\left[\left(\frac{\cos^2\phi}{q_x}+\frac{\sin^2\phi}{q_y}\right)x^2 + \left(\frac{\sin^2\phi}{q_x}+\frac{\cos^2\phi}{q_y}\right)y^2 + \sin(2\phi)\left(\frac{1}{q_y}-\frac{1}{q_x}\right)xy\right]\right\} \tag{2.46}$$

The beam described by Eq. (2.46) has what is termed *simple astigmatism*. The loci of constant intensity and constant phase have a fixed orientation (at angle ϕ) as the beam propagates. It can be demonstrated, however, that Eq. (2.46) remains a solution of the paraxial wave equation even if ϕ is complex. In this case, the beam has *general astigmatism*. In general, for a beam with general astigmatism, the loci of constant intensity and constant phase are never aligned and change their orientation as the beam propagates. General astigmatism results when a beam is passed through a nonorthogonal optical system, e.g., cylindrical lenses oriented at an angle other than 90 degrees. For a complete discussion of generally astigmatic beams, see Arnaud and Kogelnik(1).

If we assume that diffraction from apertures can be ignored, the most general Gaussian beam [Eq. (2.46)] also propagates as a Gaussian; the optical system transforms q_x, q_y, and ϕ (all are complex) in a manner analogous to the *ABCD* law [Eq. (2.42)]. The transformation relations are, of course, much more complex since one cannot make the assumptions of rotational symmetry or orthogonality for the optical system.

OSLO contains two different methods of Gaussian beam propagation analysis. For the common case of a beam propagating along the axis of a centered system, there is an interactive analysis spreadsheet, based on the *ABCD* law. This analysis is valid for centered, orthogonal systems consisting of refracting or reflecting surfaces. For a more general system, the astigmatic beam trace can be used to propagate a Gaussian beam along the current reference ray. This analysis should be used if the optical system contains any special data, is nonorthogonal, or the beam does not propagate along the optical axis of an orthogonal system.

The Gaussian beams discussed so far can be considered as the output of ideal laser resonators. In reality, of course, real resonators have hard-edged apertures, misaligned mirrors, etc., so the output beam will not, in general, be the ideal Gaussian described by Eq. (2.19). One method of quantifying how much a real beam departs from a Gaussian is the so-called M^2 factor, introduced by Siegman(2). M^2 is related to the second moments of the irradiance distribution of the beam in the near and far fields. For an ideal Gaussian beam, the product of the beam waist and far-field divergence angle [Eq. (2.24)] is given by

$$w_0\theta = \frac{\lambda}{\pi} \tag{2.47}$$

which is only a function of the wavelength λ. Denoting the beam waist radius for a real beam by W_0, and the far-field divergence angle of the beam by Θ, the corresponding product for the real beam can be written as

$$W_0\Theta = M^2 w_0\theta = M^2\frac{\lambda}{\pi} \tag{2.48}$$

Thus, M^2 is the amount by which the beam waist-far-field product exceeds the diffraction limit of an ideal Gaussian beam of the same wavelength. It can be shown that the propagation of the spot size of real beams described by an M^2 factor is described by the same equation as for an ideal Gaussian [Eq. (2.22)] but with λ replaced by $M^2\lambda$.

Polarization analysis

For most of the problems the optical designer is called upon to solve, it is unnecessary to take into account that the optical system is actually transforming the properties of an electromagnetic field. Usually, the description of light as a ray phenomenon (i.e., geometrical optics) or as a scalar wave phenomenon provides sufficient accuracy to predict the performance of a lens. There are conditions, however, under which it is necessary to treat the optical field using its true vector nature; consider, for example, the calculation of the amount of light reflected from an air-glass interface. In this section, we will review some basic electromagnetism and see how to couple this information with ray tracing to provide a way to analyze the polarization-dependent properties of optical systems.

For the discussion of polarization effects, we will use *plane wave* solutions to Eq. (2.12):

$$u_{plane}(x, y, z) = A_p \exp(\pm ikz) \tag{2.49}$$

where, if **s** is a unit vector in the direction of propagation of the plane wave, **k** = *k***s** and δ is a constant phase offset. (The actual electric field, a real, physical quantity, is given by the real part of Eq. (2.49).) The solutions given by Eq. (2.49) are called plane waves since, at a fixed time *t*, the electric field is constant over the planes defined by **s** • **r** = constant. Using Eq. (2.49) in Maxwell's equations, it can be shown that

$$\mathbf{E} = -\sqrt{\frac{\mu}{\varepsilon}}\mathbf{s}\times\mathbf{H} \tag{2.50}$$

and

$$\mathbf{H} = \sqrt{\frac{\varepsilon}{\mu}}\mathbf{s}\times\mathbf{E} \tag{2.51}$$

Taking the dot product of Eqs. (2.50) and (2.51) with **s** gives

$$\mathbf{E}\bullet\mathbf{s} = \mathbf{H}\bullet\mathbf{s} = 0 \tag{2.52}$$

Equation (2.52) reveals that the field is *transverse*, i.e., the electric and magnetic vectors lie in a plane perpendicular to the direction of propagation. Also, we see that **E**, **H**, and **s** form a right-handed orthogonal set of vectors.

The instantaneous Poynting vector **S**, defined by **S** = **E** × **H**, represents the amount and direction of energy flow, and is seen to be in the direction of **s**, i.e., the direction of propagation of the plane wave. In geometrical optics, rays represent the flow of energy. These facts provide a mechanism for us to combine the results of ray tracing with electromagnetic analysis. For each ray that is traced, there is an associated electromagnetic field that is locally planar in a small region around the ray and has its electric and magnetic vectors in a plane that is orthogonal to the ray. The process of polarization ray tracing involves finding the transformation of these locally planar electromagnetic fields, in addition to the normal refraction/reflection of the geometric ray.

Obviously, if we know any two of the three vectors of interest (**s**, **E**, and **H**), we can find the third by the use of the above equations. We will want to use **s**, since it corresponds to the geometrical ray. It is logical to use the electric vector **E** as the light vector. Wiener's experiment on standing light waves (1890) demonstrated that the photochemical process (i.e., the exposure of a photographic emulsion) is directly related to the electric vector and not to the magnetic vector. Also, the Lorentz equation for the force exerted on a charged particle shows that the electric field acts upon the particle even if the particle is at rest. The magnetic field, on the other hand, results in a force upon the particle that is proportional to the ratio of the particle's velocity to the speed of light. Since this ratio is usually very small, the effect of the magnetic field can usually be ignored. Thus, we will concern ourselves only with the electric field when considering the interaction of the vector nature of the light field with optical systems.

Polarization ellipse

We have seen that it is necessary to characterize the electric field. The description of the electric field vector defines its *polarization* properties. We have already noted one fundamental property, namely that the field is transverse to the direction of propagation of the wave. To describe this transverse field, we will choose a coordinate system that has its *z*-axis along the direction of propagation. Because the field is transverse, the electric field vector then lies in the *xy* plane. We want to study the orientation of this vector in the *xy* plane as a function of time.

From the general form of the plane wave solution [Eq. (2.49)], we see that each Cartesian component of the electric field is of the form

$$a\cos\left(\omega t - \mathbf{k}\bullet\mathbf{r} + \delta\right) = \Re\left\{ae^{i(\omega t - \mathbf{k}\bullet\mathbf{r} + \delta)}\right\} \tag{2.53}$$

Thus, the *x*, *y*, and *z* components of the electric field are given by

$$E_x = a_x\cos\left(\omega t - \mathbf{k}\bullet\mathbf{r} + \delta_x\right) \tag{2.54}$$

$$E_y = a_y\cos\left(\omega t - \mathbf{k}\bullet\mathbf{r} + \delta_y\right) \tag{2.55}$$

$$E_z = 0 \tag{2.56}$$

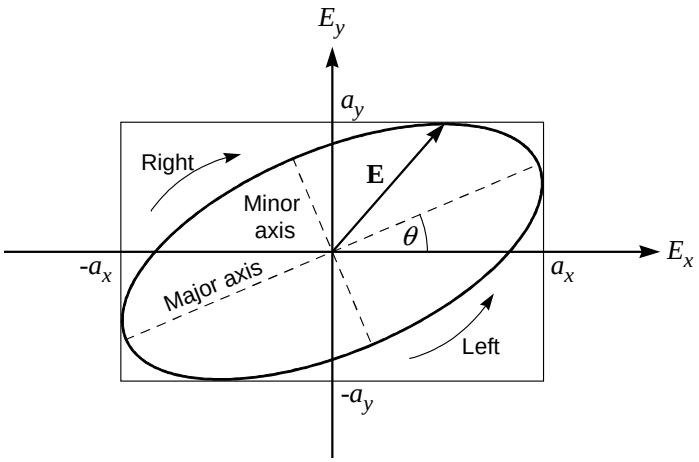
We are interested in the curve that the end point of **E** sweeps out in the *xy* plane as a function of time at a fixed point in space; this is the locus of points (E_x, E_y), where E_x and E_y are given by Eqs. (2.54) and (2.55). Since this curve is not a function of position or time, we can eliminate $\omega t - \mathbf{k}\bullet\mathbf{r}$ from Eqs. (2.54) and (2.55) to yield

$$\left(\frac{E_x}{a_x}\right)^2 + \left(\frac{E_y}{a_y}\right)^2 - 2\frac{E_x}{a_x}\frac{E_y}{a_y}\cos\delta = \sin^2\delta \tag{2.57}$$

where $\delta = \delta_y - \delta_x$. This is the equation of an ellipse making an angle θ with the (E_x, E_y) coordinate system such that

$$\tan 2\theta = \frac{2a_x a_y}{a_x^2 - a_y^2}\cos\delta \tag{2.58}$$

This ellipse is called the *polarization ellipse* and is illustrated in the figure below; it is the curve swept out by the tip of the electric vector in the plane orthogonal to the direction of propagation of the plane wave.



The principal axes of the ellipse are aligned with the (E_x, E_y) coordinate axes only if $\theta = 0$, i.e., if δ is an odd multiple of $\pi/2$. Note that, in time, the ellipse may be traced in either a clockwise or counter-clockwise sense. Conventionally, the rotation direction is based on the behavior of the electric vector when viewed *from* the direction of propagation, i.e., the wave is traveling toward the observer. If the **E** vector is rotating clockwise ($\sin\delta > 0$), the polarization is said to be *right-handed*. Conversely, if the rotation is counter-clockwise ($\sin\delta < 0$), the polarization is *left-handed*. In summary, the polarization state is determined by three quantities: *i*) the ratio of the minor axis of the polarization ellipse to the major axis of the polarization ellipse, *ii*) the orientation angle of the polarization ellipse major axis to one of the coordinate system axes (This is the *y*-axis in OSLO, so the orientation angle is $90° - \theta$.), and *iii*) the handedness of the polarization. In OSLO, these quantities for the incident electric field are specified by the *polarization operating conditions*.

There are two important special cases of the general case of elliptical polarization. If δ is equal to an integer multiple of π, then Eq. (2.57) reduces to

$$E_y = \pm\frac{a_y}{a_x}E_x \tag{2.59}$$

This is the equation of a straight line and so we say that the light is *linearly polarized*. For linear polarization, the ratio of the axes of polarization ellipse is zero and the orientation angle of the ellipse is the angle between the *y*-axis and the plane of vibration of the electric vector. Obviously, handedness is not really a meaningful quantity for linearly polarized light.

The other special case occurs when $a_x = a_y = a$, and δ is equal to an odd multiple of $\pi/2$. In this case, Eq. (2.57) reduces to

$$E_x^2 + E_y^2 = a^2 \tag{2.60}$$

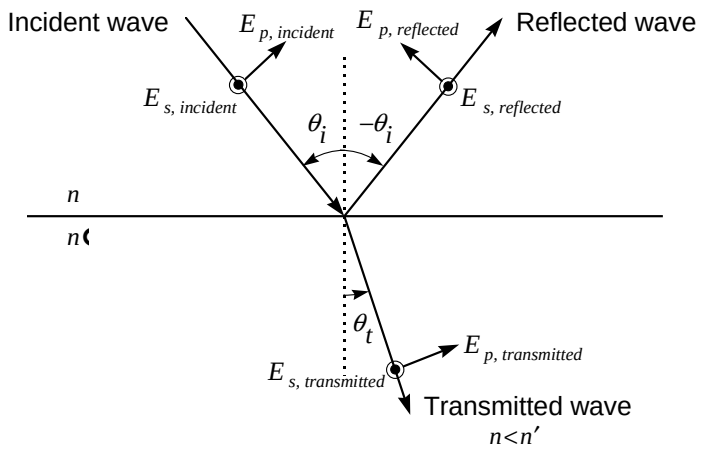
This is the equation of a circle and so we say that the light is *circularly polarized*. For circular polarization, the ratio of the axes of the polarization ellipse is unity and the orientation angle of the ellipse is arbitrary.

For this analysis, we have assumed that the electric field vector varies in a regular way. For thermal light sources, which are based on spontaneous emission from excited atoms or molecules, the changes in the polarization state may occur too rapidly and in too unpredictable a fashion for a defined polarization state to be determined. In this case the light is called *unpolarized* (or *natural*) light. In general, light is neither completely polarized nor completely unpolarized. In this case, we say that the light is *partially polarized*. The *degree of polarization* is the ratio of the intensity of the polarized portion of the light to the total intensity in the beam. If the degree of polarization is zero, the light is completely unpolarized. Conversely, a degree of polarization of one means the light is completely polarized.

Fresnel equations

In order to calculate the polarization state along the rays being traced through an optical system, we must calculate the effect on the electric field upon passing through an interface between media of different refractive indices. In general, when a plane wave is incident upon the interface part of the wave is refracted and part is reflected. (We assume the media are non-absorbing, for simplicity.) The relations describing the ratios of the transmitted and reflected fields to the incident field are called the *Fresnel equations*.

The geometrical optics laws of refraction and reflection state the incident ray, the refracted or reflected ray, and the normal to a surface lie in a plane. Solving Maxwell's equations and applying the necessary boundary conditions yields results entirely consistent with geometrical optics: a beam is reflected at an angle that is equal in magnitude but opposite in sign from the incident beam and a beam is refracted into the second medium at an angle given by Snell's law. The plane defined by the propagation vectors **s** of the incident, reflected and refracted beams (i.e., the rays) and the normal vector to the surface is called the *plane of incidence*, which is the plane illustrated in the figure below.



The electric field can always be decomposed into a component that is parallel to the plane of incidence and a component that is perpendicular to the plane of incidence. The parallel component is known as *p*, π, or TM (transverse magnetic) polarization, while the perpendicular component is known as *s*, σ, or TE (transverse electric) polarization. [*s* stands for *senkrecht*, the German word for orthogonal.] If we denote the angle of incidence by θ_i, the angle of refraction by θ_t, the ratio of the amplitude of the reflected beam to the amplitude of the incident beam by *r*, and the ratio of the amplitude of the transmitted (refracted) beam to the amplitude of the incident beam by *t*, the Fresnel equations have the following form.

$$r_s = \frac{n\cos\theta_i - n'\cos\theta_t}{n\cos\theta_i + n'\cos\theta_t} \tag{2.61}$$

$$r_p = \frac{n'\cos\theta_i - n\cos\theta_t}{n'\cos\theta_i + n\cos\theta_t} \tag{2.62}$$

$$t_s = \frac{2n\cos\theta_i}{n\cos\theta_i + n'\cos\theta_t} \tag{2.63}$$

$$t_p = \frac{2n\cos\theta_i}{n'\cos\theta_i + n\cos\theta_t} \tag{2.64}$$

In Eqs. (2.61) - (2.64), *n* is the refractive index of the medium in which the incident wave travels, and n' is the refractive index of the medium in which the transmitted (refracted) wave travels. Also note that we have made the assumption that we are dealing with dielectrics with permeabilities equal to that of free space, i.e., $\mu = \mu' = \mu_0$. The reflectance *R* (the fraction of the incident power or energy contained in the reflected beam is given by the squared modulus of the amplitude reflection coefficient

$$R_s = r_s r_s^* \qquad R_p = r_p r_p^* \tag{2.65}$$

It can be shown that

$$|r_s|^2 + |t_s|^2 = 1 \qquad |r_p|^2 + |t_p|^2 = 1 \tag{2.66}$$

Equation (2.66) can be interpreted as a statement that energy is conserved for a light wave incident upon a boundary between two dielectric media. In general, the reflectance is different for *s* and *p* polarization. Only for the case of normal incidence (where *s* and *p* are indistinguishable) are the reflectances the same value, namely

$$R_{s,normal} = R_{p,normal} = \left(\frac{n-n'}{n+n'}\right)^2 \tag{2.67}$$

Using values for a typical air-glass interface (n = 1.0, n' = 1.5) in Eq. (2.67) yields the familiar result of 4% reflection loss for an uncoated refractive optical surface.

As part of the polarization ray trace in OSLO, the incident electric field associated with each ray at a surface is decomposed into its *s* and *p* components, and the Fresnel equations are used to compute the amplitude of the transmitted electric field. The *s* and *p* directions for each ray are determined by the ray direction and the surface normal vector; they are, in general, different for each ray incident upon the surface. Thus one can not usually define overall *s* and *p* directions for non-planar waves incident upon non-planar surfaces.

Jones calculus

A convenient method, based on the use of linear algebra, for analyzing the propagation of polarized light is the *Jones calculus*, named for its inventor, R. Clark Jones. If we consider, as usual, a polarized wave traveling in the *z* direction, then the electric field has components in the *x* and *y* directions only. We write the instantaneous *x* and *y* scalar components of **E** as the column vector (*Jones vector*)

$$\mathbf{E} = \begin{bmatrix} E_x(t) \\ E_y(t) \end{bmatrix} \tag{2.68}$$

Using the complex representations of Eqs. (2.54) and (2.55), we can rewrite Eq. (2.68) as

$$\mathbf{E} = \begin{bmatrix} a_x e^{i\delta_x} \\ a_y e^{i\delta_y} \end{bmatrix} \tag{2.69}$$

It is only the phase difference $\delta = \delta_y - \delta_x$ that affects the state of polarization. Thus, it is common to just use this relative phase difference in writing Eq. (2.69) as

$$\mathbf{E} = \begin{bmatrix} a_x \\ a_y e^{i\delta} \end{bmatrix} \tag{2.70}$$

For example, light that is linearly polarized in the *y* direction has the Jones vector

$$\mathbf{E}_{linear,y} = \begin{bmatrix} 0 \\ a_y e^{i\delta_y} \end{bmatrix} \tag{2.71}$$

Right-handed, circularly polarized light has the Jones vector

$$\mathbf{E}_{circular,right} = a\begin{bmatrix} 1 \\ i \end{bmatrix} \tag{2.72}$$

while left-handed, circularly polarized light is represented by the Jones vector

$$\mathbf{E}_{circular,left} = a\begin{bmatrix} 1 \\ -i \end{bmatrix} \tag{2.73}$$

In terms of describing the state of polarization, an optical element or system can be considered by transforming an incident Jones vector $\mathbf{E}_i$ into a transmitted Jones vector $\mathbf{E}_t$. Mathematically, this transformation is represented by the 2 x 2 *Jones matrix* **J**, so that

$$\mathbf{E}_t = \mathbf{J}\mathbf{E}_i \tag{2.74}$$

$$\begin{bmatrix} E_{tx} \\ E_{ty} \end{bmatrix} = \begin{bmatrix} J_A & J_B \\ J_C & J_D \end{bmatrix} \begin{bmatrix} E_{ix} \\ E_{iy} \end{bmatrix} \tag{2.75}$$

The elements of the Jones matrix, J_A, J_B, J_C, and J_D, are, in general, complex quantities. The utility of the Jones calculus is that the Jones matrix **J** can represent any linear optical element. In OSLO, you can enter a polarization element by defining a Jones matrix for the surface. By default, an entered polarization element does not change the state of polarization, i.e., **J** is the identity matrix ($J_A = J_D = 1$; $J_B = J_C = 0$). Several example Jones matrices for common polarization elements are given below.

- Ideal linear polarizer with pass-plane oriented along the *x*-axis

$$\mathbf{J}_{x-pzr} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.76}$$

- Ideal linear polarizer with pass-plane oriented along the *y*-axis

$$\mathbf{J}_{y-pzr} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.77}$$

- Ideal linear polarizer with pass-plane oriented at an angle ϕ with respect to the *y*-axis (ϕ is a positive angle when measured from the positive *y*-axis toward the positive *x*-axis.)

$$\mathbf{J}_{\phi-pzr} = \begin{bmatrix} \sin^2\phi & \cos\phi\sin\phi \\ \cos\phi\sin\phi & \cos^2\phi \end{bmatrix} \tag{2.78}$$

- Quarter-wave plate with fast axis along the *x*-axis

$$\mathbf{J}_{x,\frac{\lambda}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} \tag{2.79}$$

- Quarter-wave plate with fast axis along the *y*-axis

$$\mathbf{J}_{y,\frac{\lambda}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{2.80}$$

- Half-wave plate with fast axis oriented at an angle ϕ with respect to the *y*-axis

$$\mathbf{J}_{\phi,\frac{\lambda}{2}} = \begin{bmatrix} -\cos 2\phi & \sin 2\phi \\ \sin 2\phi & \cos 2\phi \end{bmatrix} \tag{2.81}$$

- Linear retarder with retardation δ and with fast axis oriented at an angle ϕ with respect to the *y*-axis

$$\mathbf{J}_{\phi,\delta} = \begin{bmatrix} \sin^2\phi + \cos^2\phi\exp(-i\delta) & \sin\phi\cos\phi\left[1-\exp(-i\delta)\right] \\ \sin\phi\cos\phi\left[1-\exp(-i\delta)\right] & \sin^2\phi\exp(-i\delta) + \cos^2\phi \end{bmatrix} \tag{2.82}$$

- Homogeneous right circular polarizer

$$\mathbf{J}_{right\ circular\ pzr} = \frac{1}{2}\begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix} \tag{2.83}$$

- Homogeneous left circular polarizer

$$\mathbf{J}_{left\ circular\ pzr} = \frac{1}{2}\begin{bmatrix} 1 & i \\ -i & 1 \end{bmatrix} \tag{2.84}$$

Optical materials

The design of any optical system involves the selection of the proper materials, which depend on the spectral region, the environment, and the application. Of course the predominant optical

material is glass, at least for refracting systems, so this term is used generically, in much the same way that the term *lens* is used to denote an optical imaging system. In practice, the term *glass* may refer to a wide variety of materials, ranging from polycrystalline infrared materials to reflectors.

The performance of any optical system depends on the wavelength at which it is used, by virtue of diffraction if nothing else. In the case of refracting systems, the optical properties of materials change with wavelength. The variation of refractive index of optical materials with wavelength is called *dispersion*, and the defects in optical imagery caused by dispersion are called *chromatic aberrations*. Chromatic aberrations are considered separately from monochromatic aberrations because they can exist by themselves, in addition to causing chromatic variations of monochromatic aberrations.

There are several factors that are important in selecting a material for an optical design. As noted above, the most important is often the dispersion, but many other attributes must also be considered, such as thermal characteristics, weight, mechanical and chemical properties, availability, and cost. Many of these attributes do not directly affect the computer optimization process, but are made available for the designer when data are available.

Dispersion

The refractive index of many optical materials can be described by the Sellmeier formula

$$n^2(\lambda) = 1.0 + \frac{b_1\lambda^2}{\lambda^2 - c_1} + \frac{b_2\lambda^2}{\lambda^2 - c_2} + \frac{b_3\lambda^2}{\lambda^2 - c_3} \tag{2.85}$$

where λ is the wavelength in μm. This formula has recently been adopted by Schott and other glass manufacturers for describing the refractive index of optical glasses in the visible portion of the spectrum. Formerly, optical glasses were usually described by a Laurent series, sometimes called the Schott formula

$$n^2(\lambda) = A_0 + A_1\lambda^2 + \frac{A_2}{\lambda^2} + \frac{A_3}{\lambda^4} + \frac{A_4}{\lambda^6} + \frac{A_5}{\lambda^8} \tag{2.86}$$

Various other formulas are used for special purposes. Conrady found that in the visible portion of the spectrum, using data for only three refractive index-wavelength pairs, a good fit could be obtained using the formula

$$n(\lambda) = n_0 + \frac{A}{\lambda} + \frac{B}{\lambda^{3.5}} \tag{2.87}$$

More recently, Buchdahl introduced a *chromatic coordinate* for accurately characterizing the refractive index. The motivation for the chromatic coordinate was that the usual dispersion models (e.g., the equations above) do not have the form of a Taylor series, the function expansion form used in aberration theory. Starting from the Hartmann formula, Buchdahl proposed the use of the chromatic coordinate $\omega(\lambda)$

$$\omega(\lambda) = \frac{\lambda - \lambda_0}{1 + 2.5(\lambda - \lambda_0)} \tag{2.88}$$

where the wavelength λ is expressed in μm and λ_0 is a reference wavelength, typically the d line (0.5876 μm) for visible light. The refractive index n is then given by a power series in ω

$$n(\omega) = n_0 + \nu_1\omega + \nu_2\omega^2 + \nu_3\omega^3 + \cdots \tag{2.89}$$

where n_0 is the index of refraction at the reference wavelength λ_0. The ν_i coefficients are specific to each glass. The advantage of the chromatic coordinate is indicated by the rapid convergence of the above expansion. The paper by Robb and Mercado(3) states that a quadratic model (n_0, ν_1, ν_2) yields a maximum error in n of 0.0001 over the visible spectrum for a sample of 813 glasses from five manufacturers. More details on the use of the chromatic coordinate may be found in Forbes.(4)

A number of other parameters are used to describe the dispersion of optical glass. The one most used in the visible portion of the spectrum is the V (or ν) number, or Abbe number, defined by

$$V \equiv \frac{n_d - 1}{n_F - n_C} \tag{2.90}$$

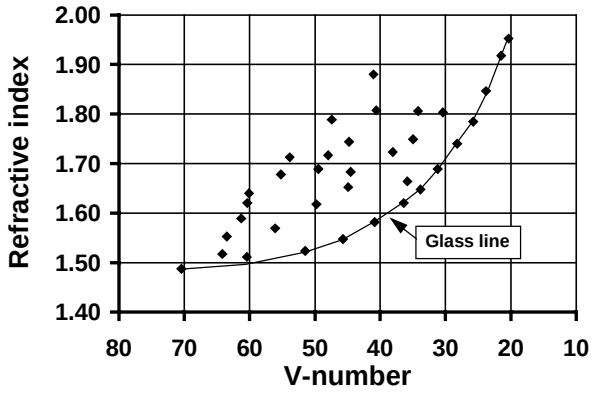
where n_d is the refractive index of the glass at the helium d (0.5876 μm) line, n_F is the refractive index at the hydrogen F (0.4861 μm) line, and n_C is the refractive index at the hydrogen C (0.6563 μm) line. Depending on the particular spectral region under study, wavelengths other than the F, d, and C lines may be used for the V number and partial dispersion.

In OSLO, wavelength 1 is taken to be the primary wavelength, wavelength 2 to be the short wavelength, and wavelength 3 to be the long wavelength. Three wavelengths must be specified to enable the computation of chromatic aberrations. The default values of these wavelengths are d, F and C. If other wavelengths are used, the program computes the V-number using the assigned wavelengths.

Typically, in the visible spectrum, the refractive index of optical glasses is a few percent higher in the blue portion of the spectrum than in the red. The difference in the refractive indices of a glass at two different wavelengths is known as the *dispersion* for the two lines in question. In the case where the wavelengths are the F and C lines, the dispersion is called the *principal* dispersion, and the dispersion for other lines is known as the *partial dispersion*. The partial dispersion is often expressed as a ratio. For example, the relative partial dispersion for the F and d lines is given as

$$P_{F,d} = \frac{n_F - n_d}{n_F - n_C} \tag{2.91}$$

The characteristics of optical glasses are often displayed on a two-dimensional graph, called a *glass map*. A glass map is a graph of refractive index plotted as a function of V-number, and any particular glass corresponds to a point on the glass map. By convention, the V-number is plotted along the *x*-axis, with decreasing values of V (i.e., increasing dispersive power) toward the right. A typical glass map is shown below.



Most glasses lie along or near a line forming the lower right boundary of the region occupied by optical glasses. This line is called the *glass line*. The availability of optical glasses that are located a considerable distance above the glass line gives the optical designer considerable flexibility in correcting the chromatic (and other) aberrations of optical systems. However, when an arbitrary choice is to be made, the glass chosen should be one on or near the glass line, since such glasses are cheaper and more readily available.

Thermal coefficients

OSLO has commands for setting the temperature of a lens and the atmospheric pressure of its air spaces. The **tem** command is used to set the temperature of the lens (in degrees Celsius), and the **pre** command sets the atmospheric pressure (in atmospheres). The default temperature is 20

degrees C and the default pressure is 1 atmosphere. Changing the temperature changes the absolute refractive indices of the glasses in the lens; since the refractive index of the air changes with temperature and pressure, the relative (to air) refractive indices of the glasses also change. OSLO uses relative refractive indices for all computation, so the index of AIR is always given as 1.0 for all temperatures, pressures, and wavelengths. To specify a vacuum as the medium between glass elements, set the pressure to zero.

Changing the temperature also causes the glasses and mounting/spacer materials in the lens to expand or contract. In OSLO, the radii of curvature, axial thickness, and aperture radii of glass elements expand according to a linear model:

$$L(T+\Delta T)=(1+\alpha\Delta T)L(T) \qquad (2.92)$$

where L is a length (e.g., thickness, radius of curvature, or aperture radius), T is the "base" temperature, ΔT is the change in temperature, and α is the thermal coefficient of expansion (TCE). Values of expansion coefficients are expressed in units of 1×10^{-7}. The default values of thermal coefficients of expansion for glasses are taken from the glass catalogs; the default value for air spaces is that of aluminum (236.0×10^{-7}). The default value can be overridden for any surface by using the **tce** command. Thermal expansion can be inhibited for any surface by setting the TCE value to zero.

Other glass data

In addition to the coefficients used to compute the refractive index, dispersion, and thermal properties, OSLO glass catalogs contain data describing several other properties of optical material (where available). Although most of the data, excepting the thermal coefficient of expansion and dn/dT, are not used explicitly by the program, they are displayed for the optical designer to see whenever a glass is entered from a glass catalog. The data are as follows.(5)

n	Refractive index (at wavelength 1)
V	Abbe number (at wavelengths 1, 2, and 3)
dens	Density (g/cm^3)
hard	Knoop hardness, HK
chem	Chemical properties (first digits of climatic resistance CR, stain resistance FR, acid resistance SR, alkali resistance AR, and phosphate resistance PR)
dndT	Derivative of refractive index with respect to temperature (1×10^{-6}/K)
TCE	Thermal coefficient of expansion (1×10^{-7}/K)
bub	Bubble group
trans	Internal transmittance (25 mm thickness, 400 nm wavelength)
cost	Cost relative to BK7 (for Schott catalog) or BSL7 (for Ohara catalog)
avail	Availability code

Model glasses

The refractive index and dispersion of optical materials are discrete quantities. That is, the optical properties are characteristic of the material and cannot be varied. In order to optimize the glass in an optical system, it is necessary to construct a model that allows the optical properties to vary continuously, at least for the damped least squares algorithm used in OSLO. Following optimization, it is necessary to pick a real glass that has properties that are suitably close to those

used in the model. OSLO contains a routine that takes a model glass and looks through the available glass catalogs to find the closest real match. It works by minimizing the RMS distance of the glass on the refractive index vs. V-number glass map.

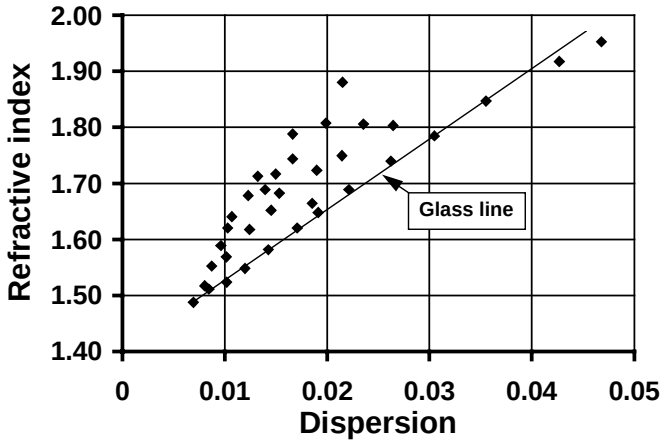
Because there are only a comparatively few glasses available, and because glass variables are inherently two-dimensional (refractive index and dispersion), optimization of glasses is more difficult than optimization of curvatures and thicknesses. To develop a practical optimization model for the optical properties of a glass, it is necessary to make a felicitous choice of variables, to reduce the dimensionality of the problem. As mentioned above, the model is two-dimensional, but actually the dimensionality is higher than two because of the nonlinearity of refractive index vs. wavelength. Using the Conrady formula mentioned above, for example, gives a three-dimensional (n_0, A, B) model, as does the quadratic chromatic coordinate model.

It is possible to reduce the problem to a two-dimensional one by assuming that only *normal* glasses are to be used. A normal glass is one for which the partial dispersion at any wavelength is proportional to the V-number. That is, for any wavelengths x and y, the partial dispersion is

$$P_{xy} \equiv \frac{n_x - n_y}{n_F - n_C} \approx a_{xy} + b_{xy}\mathrm{V} \tag{2.93}$$

The constants in this equation can be determined by taking two glasses deemed to be *normal*, and solving using the actual data. In OSLO, the two glasses used are K7 and F2, according to the recommendation in the Schott catalog. By using only normal glasses, we disallow the possibility of accounting for the nonlinearity of refractive index vs. wavelength, and hence of finding *apochromatic* combinations of glasses, i.e., pairs that correct the secondary spectrum, during optimization.

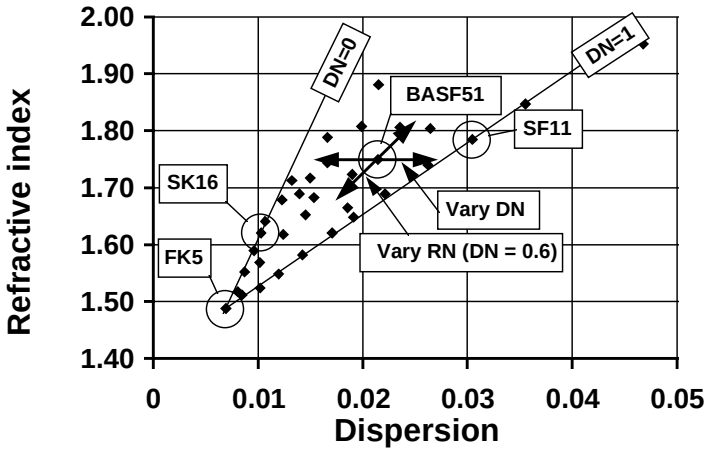
Although the model described above is 2-dimensional, it is implemented in OSLO in a way that makes it possible to optimize the glass for most systems using a single variable. The basis for the OSLO model comes from modifying the standard glass map so that the refractive index is shown as a function of dispersion ($n_F - n_C$), rather than V-number. The figure below shows data for the same glasses used before.



The so-called *glass line*, which bounds the lower right edge of the chart and contains most of the real glasses, has been converted from a curve to a nearly straight line. This motivates the transformation of the material parameters n (refractive index) and Δn (dispersion) into two derived variables RN and DN. The derived variables RN and DN are defined in OSLO according to the following conditions.

The variable RN changes both the refractive index and dispersion of a model glass along a straight line plotted on the new glass map. The line along which the index and dispersion is varied is the one that connects the point characterizing the initial glass, to the point that characterizes FK5 glass (at the extreme lower left). The figure below shows the case where the initial glass is BASF51.

The variable DN varies the dispersion without changing the refractive index. It is normalized in such a way that a value of 0.0 corresponds to a point along a line connecting the glass SK16 to the glass FK5, and the value 1.0 corresponds to a point along a line connecting the glass SF11 to the glass FK5. A DN of 1.0 thus denotes a glass along the glass line, and a value of 0.0 denotes a glass of approximately the lowest dispersion that can be obtained for the given refractive index.



The above rules make it feasible to use only the RN variable for many problems. If an initial glass is selected along the glass line (DN = 1.0), OSLO will hold the glass along that line, increasing the likelihood that an actual glass can be found that matches the model glass found during optimization.

---

1. J. A. Arnaud and H. Kogelnik, “Gaussian light beams with general astigmatism,” Appl. Opt. **8**, 1687-1693 (1969).
2. A. E. Siegman, Proc. SPIE **1224**, 2, 1990.
3. P. N. Robb and R. I. Mercado, “Calculation of refractive indices using Buchdahl’s chromatic coordinate,” Appl. Opt. **22**, 1198-1215 (1983).
4. G. W. Forbes, “Chromatic coordinates in aberration theory,” J. Opt. Soc. Am. A **1**, 344-349 (1984).
5. For information on the meaning and units of the data, see, for example, the Schott Optical Glass catalog (Schott Glass Technologies, Inc. 400 York Avenue, Duryea, PA 18642, tel 717-457-7485, fax 717-457-6960).

Chapter 3 Paraxial Optics

Paraxial optics deals with the propagation of light through a centered optical system. Such a system consists of rotationally symmetric refracting or reflecting surfaces having a common axis, called the *optical axis*. A simple lens is an example of such a centered system; its axis is the line passing through the centers of curvature of the two surfaces. A succession of simple lenses is a centered system if the lenses are aligned on a common axis. A system containing tilted plane mirrors *may* result in a centered system, if the lenses on either side of a given mirror lie on the symmetry axis, as reflected by the mirror. However, in general, systems containing tilted surfaces do not have a common axis of symmetry, and do not form a centered system.

Centered optical systems have the property that a ray passing through them sufficiently close to the optical axis always makes a small angle of incidence with the normal to any surface. Such a ray is called a paraxial ray, and the refraction of such a ray is described by a much-simplified equation, compared to the law of refraction for an arbitrary ray.

If a ray incident on a refracting surface makes an angle I with the normal to the surface, the refracted ray will make an angle I' with the normal to the surface, where I and I' are given by Snell's law,

$$n \sin I = n' \sin I' \tag{3.1}$$

and n and n' are refractive indices on either side of surface. The incident and refracted rays intersect at the surface, and define a plane, called the *plane of incidence*. According to the law of refraction, the plane of incidence contains the normal to the surface at the point of incidence. If the system is a centered system, and if the optical axis is also contained in the plane of incidence, the ray is said to be a *meridional ray*; otherwise it a *skew ray*.

The law of reflection states that the angle between the ray and the surface normal after reflection is equal in magnitude and opposite in sign to the angle of incidence. It follows from Snell's law that reflection can be considered a special case of refraction, in which the refractive index following the surface is the negative of the refractive index preceding in the surface. In practice, however, it may not be desirable to use the negative index convention. In OSLO, all refractive indices are positive, and reflection is accommodated by the way the ray trace equations are programmed.

The paraxial approximation

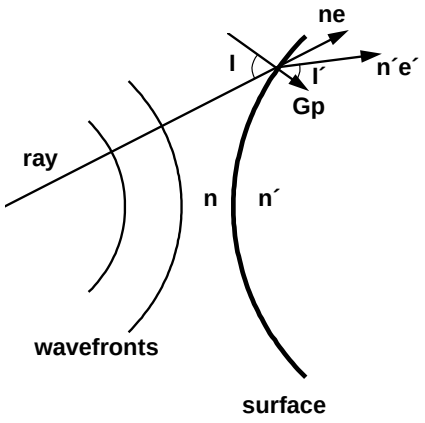
If the ray is sufficiently close to the optical axis at all points, the angle of incidence of the ray on all surfaces of the system will necessarily be small, so the sines of the angles can be satisfactorily approximated by the angles themselves, and the law of refraction becomes

$$ni = n'i' \tag{3.2}$$

where the lower case i is used to denote the paraxial value of the real angle of incidence I. The simplification of the law of refraction that is achieved by substituting the angles for their sines leads to the development of the paraxial theory of image formation. Before considering this theory in detail, we consider a simple graphical description of refraction for both real rays, which obey Eq. (3.1), and paraxial rays, which obey Eq. (3.2).

According to the law of refraction, at a boundary that separates media of different refractive indices, the incident ray, the refracted ray, and surface normal form a certain vector triangle. In particular, if the incident ray is represented by a vector $n\mathbf{e}$ of length n and the refracted ray by a vector $n'\mathbf{e}'$ of length n', the law of refraction can be expressed as

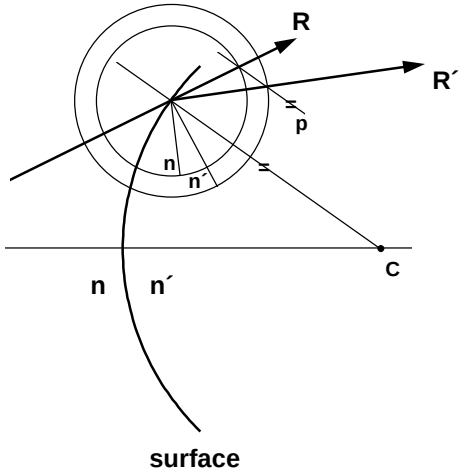
$$n'\mathbf{e}' = n\mathbf{e} + G\mathbf{p} \tag{3.3}$$



where **p** is a unit vector in the direction of the surface normal, **e** and **e′** are unit vectors in the directions of the incident and refracted rays, respectively, and *G* is a quantity that can (in principle) be obtained by taking the scalar product of **p** with the above equation, noting that $\mathbf{p}\cdot\mathbf{p} = 1$. Thus

$$G = n'(\mathbf{p}\cdot\mathbf{e}') - n(\mathbf{p}\cdot\mathbf{e}) \qquad (3.4)$$
$$= n'\cos I' - n\cos I$$

The figure below shows a graphical construction that illustrates the law of refraction. Consider a ray incident on a surface whose normal vector at the point of incidence is known. At the point where the ray intersects the surface, construct two index circles having radii that are proportional to *n* and *n′*. Extend the incident ray until it intersects the circle of radius *n*, then draw a line parallel to the surface normal from this intersection point to the point where this line intersects the index circle of radius *n′*. The line from the point of incidence to this last point represents the refracted ray.



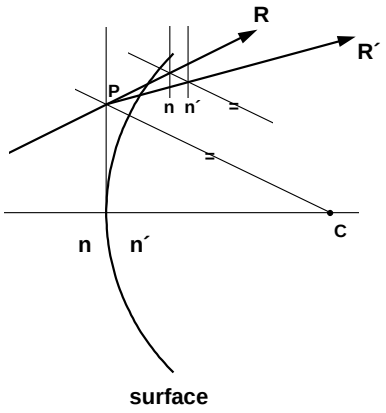
This construction is useful for graphically determining the trajectories of real rays through optical systems, and is convenient for use with "back-of-the-envelope" sketches when no computer is available. For the present, we are interested in how the construction is simplified when it is applied to paraxial rays.

Consider a real ray passing through an optical system near the optical axis. As described above, the angle of incidence of the ray on any surface must then be small. In addition, under these conditions a spherical surface will be indistinguishable from a plane surface, so that the surface and the index arcs can be approximated by straight lines.

The refraction law as shown above accurately predicts the trajectories of rays that pass through a system near the axis, but it is cumbersome to use in actual practice, because the rays are refracted by too small an amount to draw accurately. On the other hand, it turns out to be possible to use the

paraxial construction to predict the approximate trajectories of rays that pass through the system at large distances from the axis, as we shall now see.

In the figure below, we show the paraxial construction for a ray passing through a system away from the axis. The surface is replaced by its tangent plane, and the index arcs are replaced by straight lines. To construct the refracted ray, first extend the incident ray until it crosses the first index line, then draw a line parallel to the line connecting the point of incidence of the ray on the tangent plane and the center of curvature. The line connecting the point of incidence with the intersection of the construction line and the second index line gives the refracted ray, in a manner similar to that found for real rays.

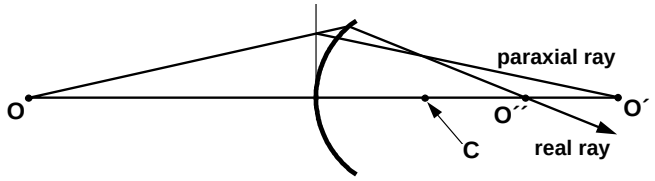


Because paraxial rays determine the locations of images in ideal systems, it is sometimes thought that a paraxial lens is an ideal lens. This is not true, because light does not travel along the trajectories of formal paraxial rays, even in an ideal system.

The difference between the two constructions is that the second one is an approximation, and hence does not give the exact trajectory for the ray. A ray constructed according to the second scheme should be called a formal paraxial ray, to distinguish it from a true paraxial ray, which passes through the system close to the axis. In practice, the designation “formal” is usually omitted, and such rays are just called paraxial rays.

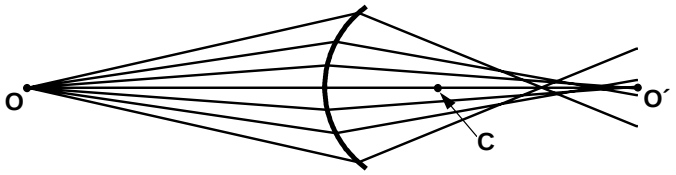
The particular construction used to describe the trajectories of formal paraxial rays is of great utility in describing the general characteristics of optical systems. The reason for this is that the construction predicts ideal image formation by paraxial rays. Although trajectories of formal paraxial rays through an optical system do not represent the actual passage of light, they do lead to a prediction of the locations of image points in an ideal system. It is for this reason that these rays are extensively used in the description of image formation.

The trajectories of a real ray and a formal paraxial ray are compared in the next figure. The two rays are refracted at different angles, of course, since they are constructed by different procedures. In addition, the real ray is refracted at the actual surface, while the paraxial ray is refracted at the tangent plane. The axial intercept of the paraxial refracted ray defines the location O' of the paraxial image of the object point O; this is the point where true paraxial rays would form an image. The real ray crosses the axis at some other point O''; the fact that this is displaced from O' indicates that the surface introduces aberration.



The aberration introduced by refraction of real rays at a spherical surface is shown more clearly in the figure that follows, where we show an accurate drawing of the trajectories of several rays

leaving a single object point and refracting into a material having a refractive index of approximately 1.9. Rays that strike the refracting surface at large angles of incidence depart strongly from the conditions needed for good image formation. In this particular case, the aberration introduced by the surface is called spherical aberration, and is characteristic of the refraction of a spherical wave by a spherical surface.

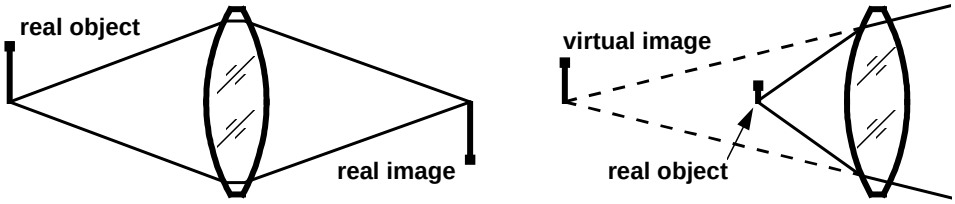


Cardinal points

A great many approaches have been used to predict the image-forming properties of optical systems. Although real optical systems form images whose quality depends on the system and the conditions under which it is used, theories of image formation are usually developed within the framework of paraxial optics, for two reasons. First, paraxial rays propagate according to relatively simple laws that lead to straightforward techniques for describing image formation. Second, the ultimate formation of an image is determined by balancing or elimination of the aberrations so that the real rays converge upon the same image point as the paraxial rays. In this section, we consider the theory of paraxial image formation developed by Gauss, which characterizes optical systems by a number of special points, known as *cardinal points*.

It is important to understand the definitions of *object space* and *image space*. Object space refers to the space in which a ray, *or its extension*, propagates before traveling through an optical system. Image space refers to the space in which a ray, *or its extension*, propagates after passing through the system.

By convention, light initially travels from left to right. If light leaves an object, passes through a lens, then converges to the image, the image is *real*. If light diverges after passing through the lens, the image is *virtual*. Since both the object and image can be either real or virtual, object space and image space extend to both sides of the lens.

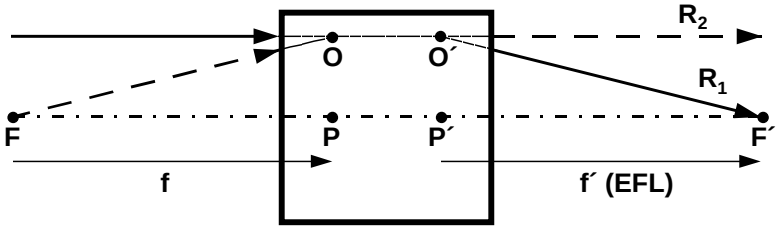


It is possible to find the properties of an image by direct paraxial ray tracing. For example, if one had a lens that formed an image of an extended object, one could select several typical points on the object and trace several rays from each point through the lens, using the graphical technique described in the previous section, to see where they converge after emerging from the system. Such a brute force approach to image formation is overly cumbersome. Since all paraxial rays leaving the same object point converge to the same image point, it suffices to trace only two rays from each object point to determine the location of the image point where all rays from that object point converge.

Consider a centered lens in some sort of mount that does not allow us to observe the actual elements that are contained in the lens, as shown below. A ray R_1 that enters the lens parallel to the optical axis from the left will emerge and intersect the optical axis at some point F'. This point is called the *focal point* (specifically, the second focal point) of the lens.

Consider a ray R_2 from a point on the left that passes through the lens to emerge parallel to the axis at the same height that R_1 entered the system. This defines the first focal point F. Now we have

two rays that intersect at some point O in object space and point O' in image space. This implies that O' is the image of O.



Paraxial optical system

Because the height of the rays entering the lens is arbitrary, it follows that any plane normal to the axis that contains O is imaged into the plane normal to the axis that contains O'. The intersections of these planes with the optical axis are called the *principal points* P and P'. The first principal point P is the one in object space, and the second principal point P' is the one in image space; P' is the image of P. Moreover, since O and O' are equidistant from the axis, the lateral magnification of the image is unity. The planes OP and $O'P'$ are called the *principal planes* of an optical system.

The distance from the first focal point to the first principal point is called the first focal length, and the distance from the second principal point to the second focal point is called the second focal length, or *effective focal length* (EFL), or simply the *focal length* f'. If the refractive index is different in object and image space, the first and second focal lengths are related by

$$\frac{n}{f} = \frac{n'}{f'} \equiv \phi \tag{3.5}$$

where ϕ is the power of the lens.

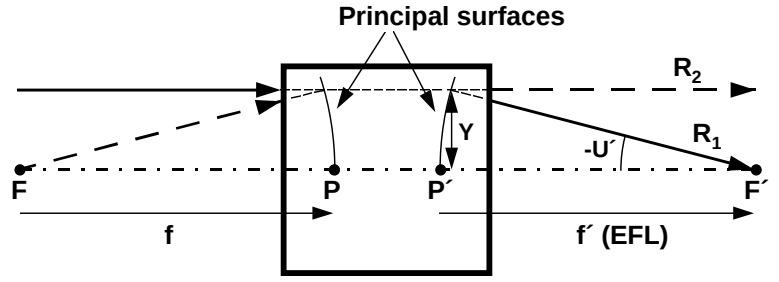
The concept of principal planes is very useful for tracing paraxial rays through an optical system, but for real systems, the surfaces of apparent refraction are not planes. A necessary condition for a perfect lens is that all rays from an object point infinitesimally displaced laterally from the axis pass through a single point in the image plane, independent of the height of the rays in the aperture of the lens. Such a lens is free from coma, and is called an *aplanat*. An aplanatic lens obeys the *Abbe sine condition*

$$\frac{u}{u'} = \frac{\sin U}{\sin U'} \tag{3.6}$$

which for rays from infinity can be written as

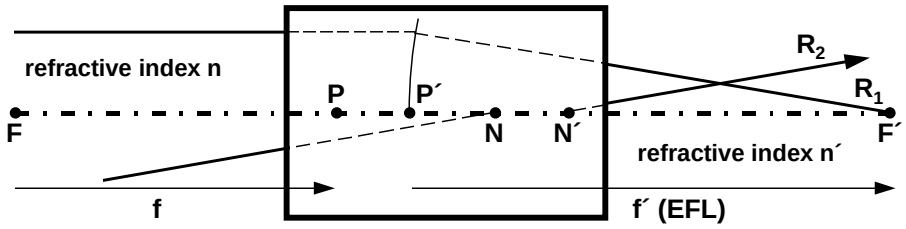
$$f' = \frac{-Y}{\sin U'} \tag{3.7}$$

For an aplanatic lens, the effective refracting surface for rays coming from infinity is a sphere centered on the second focal point, as shown below. Most real lenses more closely resemble aplanatic systems than paraxial systems.



Aplanatic optical system

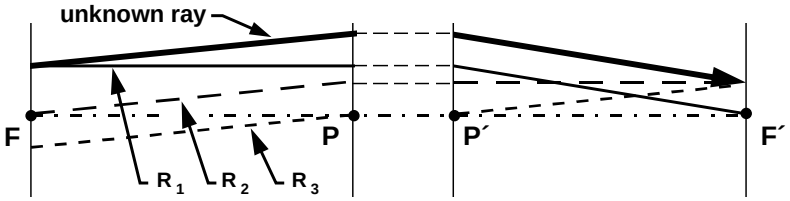
There is another pair of conjugate points that is of significant importance in determining the first-order imaging properties of an optical system. These are called the *nodal points N* and *N′*, which are defined to be the conjugate points of unit positive angular magnification. If the refractive index is the same on both sides of the optical system, the nodal points are coincident with the principal points. However, if the refractive index is different in object and image space, the nodal points are shifted from the principal points. The figure below, ray R_2 passes through nodal point *N* in object space and emerges from *N′* in images space. The distance between the nodal points is equal to the distance between the principal points, and it can be shown that $\overline{FP} = \overline{N'F'}$. Another term, the *back focus*, is the distance from the last surface to the image point. If the object is at infinity, the *back focus* it is equal to the back focal length, but for finite conjugates it is called the *working distance*.



Aplanatic system with n´ > n

The focal points, principal points, and nodal points are called the cardinal points of an optical system. A knowledge of their locations allows one to determine its image-forming properties. Note that the cardinal points are defined in terms of the action of the system on light propagating through it. There are no direct relations between the positions of the elements of the system and the locations of the cardinal points. To provide such a link, some quantity that measures the distance of a cardinal point from a lens surface is needed. The most common quantity used for such a specification is called the *back focal length* of the system, which is defined to be the distance from the last optical surface to the second focal point. It is important to note that the effective focal length of a system is quite different from the back focal length of the system; the two measure totally different quantities.

Once the locations of the cardinal points of a system are known, it is straightforward to find how the system transforms rays from object to image space. In the figure below, an unknown ray enters the first principal plane at a certain ray height and emerges from the second principal plane at the same ray height (since the principal planes are conjugate planes of unit positive lateral magnification). To determine the trajectory of the ray that emerges from the system, we can use any of three construction rays.



First, we may use for a construction ray the ray R_1 that intersects the unknown ray in the first focal plane, and enters the system parallel to the axis. This ray must pass through the second focal point after it emerges from the system, but since it intersects the unknown ray in the first focal plane, it follows that the unknown ray must emerge parallel to it.

Second, we may choose as a construction ray the ray R_2 that enters the system parallel to the unknown ray, but which passes through the first focal point. Such a ray must emerge from the system parallel to the axis. Again, since the two rays enter the system parallel to each other, they must intersect in the second focal plane, as shown.

Third, we construct a ray R_3 parallel to the unknown ray that passes through the first nodal point of the system. Such a ray must emerge undeviated from the second nodal point of the system. Since

the construction ray entered the system parallel to the unknown ray, the two rays must intersect in the second focal plane.

Paraxial ray tracing

Sign conventions

To compute the propagation of rays through optical systems mathematically, it is necessary to adopt a number of conventions regarding nomenclature and the signs of variables. We consider an optical system to be described using a series of local right-handed coordinate systems in which the *z*-direction coincides with the optical axis and points to the right, the *y*-direction points up, and *x*-direction points into the paper. Each surface of the system is described in the local coordinate system for that surface. Paraxial calculations are greatly simplified by the fact that tilted and/or decentered surfaces are not allowed. Thus the origin of each local coordinate system is on the *z*-axis of the previous local coordinate system.

When it is necessary to distinguish a quantity on both sides of surface, the quantity on the object side of the surface is written with an unprimed symbol, while the quantity on the image side of the surface is written with a primed symbol. Thus, for example, n is the refractive index on the object side of a surface, and n' is the refractive index on the image side. By convention, light enters an optical system from the left, so that quantities on the left side of a surface are unprimed, and those to the right are primed, but the more general convention is needed for reflecting systems, where light may propagate from right to left. The prime notation is also used to denote conjugate relationships. Thus, the image point corresponding to the object point O is O'.

If a ray intersects the optical axis of a system, the ray and the optical axis are coplanar, and the plane defined by them is called the *meridional plane*. By convention, the *yz* plane is chosen as the meridional plane, so for meridional rays, $x = 0$ and the ray is fully described on a given surface by its ray height, y, and its slope, u. Rays that have non-zero values of x on any surface are called *skew rays*.

OSLO handles cylindrical systems by providing a separate paraxial trace for the xz plane. However, this trace is not used in lens setup, which implies that skew paraxial constraints (e.g. solves) must be implemented as optimization operands rather than lens data.

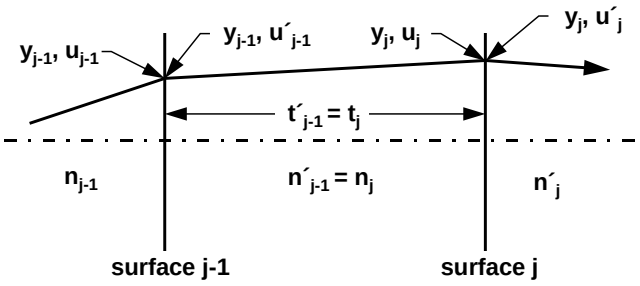
Note that the most general definition of a centered system would not necessarily imply rotational symmetry about the optical axis. Cylindrical surfaces, for example, could be permitted. However, in our discussion here, we consider only rotationally symmetric systems and assume that all object points lie on the *y*-axis. Because of the rotational symmetry, this entails no loss of generality.

A spherical surface may be specified by its radius of curvature, r, or by its curvature c, which is defined to be the reciprocal of the radius of curvature. The distance between two surfaces is called the thickness t of the space separating the two surfaces.

In a system containing several surfaces, the various parameters specifying the system are given subscripts to identify them with a particular surface. Surface parameters, such as curvatures, are subscripted with the surface number to which they pertain. Inter-surface parameters, such as thicknesses and refractive indices, are given both subscripts and primes to indicate the surface with which they are associated. This results in a tautology, in which the same physical quantity can be described by two symbols. For example, the thickness that separates the current surface from the next can be designated as either t' on the current surface or t on the next surface. In OSLO, the thickness associated with a given surface is called TH, which corresponds to t'.

The figure below illustrates the parameters used to describe an optical system for paraxial ray tracing. Note that all surfaces are replaced by their tangent planes. Although the subscripts j and $j–1$ have been attached to all parameters in the figure to clearly identify them with the proper surface, usually we will eliminate the subscript j in cases where it is clear that it should be present. For example,

$$y = y_j \qquad (3.8)$$
$$y_{-1} = y_{j-1}$$



Paraxial ray trace equations

Paraxial ray tracing is a subject that has received a great deal of study, and several efficient techniques for performing the required calculations have been developed. We consider here three methods that are in common use. These are the *ynu* method, and *yui* method, and the matrix method. The *ynu* method is the most efficient of the three, and is widely used for hand work. The *yui* method requires a few more steps but generates the paraxial angles of incidence during the trace and is probably the most common method used in computer programs.

Consider the propagation of a true paraxial ray (as opposed to a formal paraxial ray) through an optical system. For such a ray, the slope is infinitesimal, so that the angle the ray makes with the axis and the tangent of that angle (i.e., the slope) are the same. Thus we have

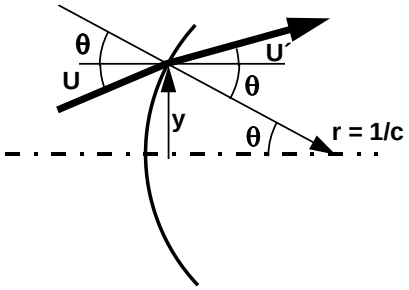
$$u = \frac{y - y_{-1}}{t} \qquad (3.9)$$

where we have omitted the subscript *j*, as described above. This equation may be transposed to give the translation equation for paraxial rays from one surface to the next,

$$y = y_{-1} + tu \qquad (3.10)$$

The refraction equation may be derived with reference to the figure below. From the figure, we see that

$$I = U + \theta \qquad (3.11)$$
$$I' = U' + \theta$$



Now in the paraxial limit, these angles can be taken equal to their sines or tangents, and hence we can write

$$i = \frac{y}{r} + u = yc + u \qquad (3.12)$$
$$i' = \frac{y}{r} + u' = yc + u'$$

Also, in this limit, the law of refraction can be written

$$ni = n'i' \tag{3.13}$$

so that we have, using the above,

$$u' = \frac{nu - y\phi}{n'} \tag{3.14}$$

where $\phi = c(n' - n)$ is called the *power* of the surface. This is the paraxial refraction equation; it gives the slope of a ray emerging from a surface in terms of the ray height and slope of the ray incident on a surface having power ϕ. The sequential application of the paraxial translation and refraction equations allows one to trace any paraxial ray through an optical system. All of the ray trace methods discussed in this section are based on these equations, or on simple variations of them.

YNU *ray tracing*

The most efficient way to trace paraxial rays by hand is the so-called *ynu* method. This method utilizes a formatted worksheet to carry out the repetitive calculations of paraxial ray heights and slopes using an optimized form of the ray trace equations.

For *ynu* ray tracing, the refraction equation is written in the form

$$n'u' = nu - y\phi \tag{3.15}$$

and the translation equation is written

$$y = y_{-1} + \frac{t}{n}(nu) \tag{3.16}$$

Since *nu* is the same as $n'u'$ on the previous surface, the quantities *y* and *nu* calculated at one stage in the trace are used as input to the next stage. For example, given *y* and *nu* on the first surface of the system, use the refraction equation to calculate $n'u'$ on this surface. This quantity is used, together with *y*, to calculate the ray height on the second surface, using the translation equation. The process then repeats, using the refraction and translation equations at the second surface.

The *ynu* ray trace can be used for a number of purposes in addition to the straightforward calculation of ray data for a given system. One of the most important of these is to synthesize a system having desired properties. In such a case, the ray data is given, instead of the lens parameters, and the lens parameters are determined from the ray data. An alternative way to specify a lens is thus to state where the paraxial rays go.

This way of specifying lens parameters is very common in computer programs for optical design. Two types of specification are commonly used. The first is called an *angle solve*, and specifies the curvature of a surface by the angle of a ray after refraction. For example, suppose we have traced a ray up to the j^{th} surface, so that we know *y* and *nu*. If we want the ray to emerge from the surface with a slope u', then according to the paraxial refraction equation, the curvature of the surface must be

$$c = \frac{n'u' - nu}{y(n - n')} \tag{3.17}$$

The second type of solve that is in common use is the *height solve*, in which the thickness on the image side of a surface is specified by giving the height of a ray on the next surface. According to the paraxial translation equation, the needed thickness is

$$t = \frac{y - y_{-1}}{u} \tag{3.18}$$

where *y* is given, and y_{-1} on the previous surface and *u* are known.

YUI *ray tracing*

An alternate method of paraxial ray tracing, which we call the *yui* method, uses the following equations, which can be readily derived from the paraxial refraction and translation equations:

$$\begin{aligned} y &= y_{-1} + tu \\ i &= u + yc \\ u' &= u + \left(\frac{n}{n'} - 1\right) i \end{aligned} \tag{3.19}$$

Although three, rather than two, equations are used to trace a ray through a surface, the quantity *i* is needed to calculate the aberration contributions of the surface, so the additional labor produces useful data. In fact, if the aberrations of the system are to be calculated, the *yui* method is actually more efficient than the *ynu* method. The *yui* method is used in OSLO for paraxial ray tracing. The listing below shows typical output for the **pxt all** command. The lens is the same one used for the *ynu* example above.

```
*PARAXIAL TRACE
 SRF       PY          PU          PI          PYC         PUC         PIC
  0         --     1.0000e-20  1.0000e-20 -1.0000e+20     1.00000     1.00000

  1      1.00000    -0.03407     0.10000       --        0.65928     1.00000
  2      0.99319    -0.02584    -0.13339     0.13186     0.61940     0.64610
  3      0.99189    -0.04177    -0.02584     0.16283     1.00132     0.61940

  4   2.8940e-05    -0.04177    -0.04177    23.94108     1.00132     1.00132
```

As shown in the next section, it is necessary to trace two paraxial rays (the axial and chief rays) to completely characterize a system. In this text, the axial ray height and slope are indicated by y_a, u_a; the chief ray height and slope are y_b, u_b. Another common scheme is to indicate the axial ray by *y*, *u* and the chief ray by $\bar{y}, \bar{u}$. The "bar" notation for the chief ray was popularized by the Military Handbook for Optical Design, and is the source of the nomenclature for the so-called *y–ybar* method of paraxial analysis, in which the chief ray height is plotted vs. the axial ray height. OSLO uses PY, PU, and PI to give data for the height, slope, and angle of incidence of the axial ray (the *a-ray*), and PYC, PUC, and PIC to give corresponding data for the chief ray (the *b-ray*).

Matrix optics

It is possible to write the paraxial ray trace equations in matrix form, so that a series of matrix multiplications are used to determine the trajectory of a paraxial ray through an optical system.

Let a ray be represented by a column vector giving the ray height and slope immediately to the image side of the j^{th} surface of an optical system:

$$R'_j = \begin{bmatrix} y_j \\ n'_j u'_j \end{bmatrix} \tag{3.20}$$

Define a translation matrix

$$\mathbf{T}'_j = \begin{bmatrix} 1 & t'_j / n'_j \\ 0 & 1 \end{bmatrix} \tag{3.21}$$

Then

$$R_{j+1} = \mathbf{T}'_j R'_j \tag{3.22}$$

or

$$\begin{bmatrix} y_j \\ n_j u_j \end{bmatrix} = \begin{bmatrix} 1 & t_j / n_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{j-1} \\ n_j u_j \end{bmatrix} \tag{3.23}$$

Writing this out in full, one obtains

$$y_j = y_{j-1} + \frac{t_j}{n_j}\left(n_j u_j\right) \tag{3.24}$$
$$n_j u_j = n_j u_j$$

The translation matrix thus takes a ray from one surface and translates it to the next surface.

We next define a refraction matrix

$$\mathbf{R}_j = \begin{bmatrix} 1 & 0 \\ -\phi_j & 1 \end{bmatrix} \tag{3.25}$$

then

$$R'_j = \mathbf{R}_j R_j \tag{3.26}$$

or

$$\begin{bmatrix} y_j \\ n'_j u'_j \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\phi & 1 \end{bmatrix} \begin{bmatrix} y_j \\ n_j u_j \end{bmatrix} \tag{3.27}$$

that is,

$$y_j = y_j \tag{3.28}$$
$$n'_j u'_j = n_{j+1} u_{j+1} = n_j u_j - y_j \phi_j$$

The refraction matrix thus duplicates the paraxial refraction equation. By successive application of the refraction and translation matrices, one can trace rays through an optical system.

Suppose we know the coordinates of a ray incident on surface 1. Then we have $R'_k = \mathbf{M}'_{k1} R_1$, or

$$\begin{bmatrix} y_k \\ n'_j u'_j \end{bmatrix} = \begin{bmatrix} A'_{k1} & B'_{k1} \\ C'_{k1} & D'_{k1} \end{bmatrix} \begin{bmatrix} y_1 \\ n_1 u_1 \end{bmatrix} \tag{3.29}$$

where the transfer matrix is given by

$$\mathbf{M}_{k1} = \begin{bmatrix} A'_{k1} & B'_{k1} \\ C'_{k1} & D'_{k1} \end{bmatrix} = \mathbf{R}_k \mathbf{T}_k \text{.....} \mathbf{T}_2 \mathbf{R}_1 \tag{3.30}$$

The quantities *A, B, C, D* are sometimes called the *Gaussian constants* of the system, and the transfer matrix is also known as the *ABCD* matrix. The main reason for using matrix techniques in paraxial optics is to provide a compact formulation for describing the general transformation properties of an optical system. When numerical values of the matrix elements are needed, they can be found from the results of *ynu* ray trace, as we shall now show.

Suppose that a ray R_a having initial coordinates (y_{a1}, $n_1 u_{a1}$) is traced through the system. Then

$$y_{ak} = A'_{k1} y_{a1} + B'_{k1} n_1 u_{a1} \tag{3.31}$$
$$n'_k u'_{ak} = C'_{k1} y_{a1} + D'_{k1} n_1 u_{a1}$$

Next we trace a different ray, R_b, whose initial coordinates are (y_{b1}, $n_1 u_{b1}$). Then

$$y_{bk} = A'_{k1} y_{b1} + B'_{k1} n_1 u_{b1} \tag{3.32}$$
$$n'_k u'_{bk} = C'_{k1} y_{b1} + D'_{k1} n_1 u_{b1}$$

The ray traces generate the data needed to solve equations for the 4 unknowns A'_{k1}, B'_{k1}, C'_{k1}, and D'_{k1}. We find that

$$
\begin{aligned}
A'_{k1} &= L_{ab}^{-1}\left[n_1\left(y_{bk}u_{a1} - y_{ak}u_{b1}\right)\right] \\
B'_{k1} &= L_{ab}^{-1}\left(y_{ak}y_{b1} - y_{a1}y_{bk}\right) \\
C'_{k1} &= L_{ab}^{-1}\left[n_1 n'_k\left(u'_{bk}u_{a1} - u'_{ak}u_{b1}\right)\right] \\
D'_{k1} &= L_{ab}^{-1}\left[n'_k\left(y_{b1}u'_{ak} - y_{a1}u'_{bk}\right)\right]
\end{aligned}
\tag{3.33}
$$

where

$$L_{ab} = n_1\left(y_{b1}u_{a1} - y_{a1}u_{b1}\right) \tag{3.34}$$

The entrance pupil is the apparent stop as seen from object space, i.e., it is the image of the aperture stop in object space.

In order for the equations to have a solution, the *a* and *b* rays must be linearly independent; that is, one must not be a scalar multiple of the other. There is no other formal requirement for these rays, but it is conventional to choose for the *a* ray, the ray that starts from the vertex of the object plane and passes through the edge (or margin) of the entrance pupil. The *a* ray is also called the axial ray, or sometimes the marginal paraxial ray. It determines the location of the image in image space. The *b* ray is conventionally chosen to be the ray from the edge of the field of view that passes through the center of the entrance pupil. This ray is also called the chief ray, or the principal ray. It determines the location of the exit pupil in image space. There is no difference in meaning between the terms chief ray and principal ray; both are in common use, and refer to the same ray (note that the principal ray does not, in general, go through the principal points).

The matrix approach to paraxial optics gives considerable insight to some general properties of optical systems. The most important of these is just that the transfer matrix exists and can be determined by the trajectories of any two independent paraxial rays. This means that if we trace two such rays through a system, we can express any third ray in terms of the first two. That is, any paraxial data can be expressed in terms of the data for the axial and chief rays.

A property of the transfer matrix that is of considerable importance is that its determinant is unity. Note that

$$\left|\mathbf{R}\right| = 1 \tag{3.35}$$

and

$$\left|\mathbf{T}\right| = 1 \tag{3.36}$$

Since the determinant of the product of several matrices is equal to the product of the determinants of the individual matrices, it follows that

$$\begin{vmatrix} A'_{k1} & B'_{k1} \\ C'_{k1} & D'_{k1} \end{vmatrix} = A'_{k1}D'_{k1} - B'_{k1}C'_{k1} = 1 \tag{3.37}$$

If any two paraxial rays are traced through an optical system, the transfer matrix can be determined using the above equations. In that case, the fact that the determinant of the transfer matrix is unity implies that the quantity L_{ab} is invariant throughout the system. That is

$$L_{ab} = ny_b u_a - ny_a u_b = \text{constant} \tag{3.38}$$

on either side of any surface of the system. This quantity is known as the *Lagrange invariant*.

Lagrange's law

Usually, in addition to knowing the location of the image plane corresponding to a given object plane, one wants to know the size of the image. This is specified by giving the *lateral magnification* of the system. If the object height is *h*, and image height is *h*′, the lateral magnification is defined to be

$$m = \frac{h'}{h} \tag{3.39}$$

A straightforward way to determine the lateral magnification of a system would be to locate the paraxial image plane by tracing an axial ray through the system and then trace the chief ray from the edge of the field of view, find the intersection point with the paraxial image plane, and calculate the required ratio to determine the lateral magnification.

A better way to calculate the lateral magnification of an image is to use *Lagrange's law*. Lagrange's law allows one to find the magnification of an image using only the data for an axial ray. It is a simple consequence of the fact that the Lagrange invariant has the same value on any surface of the optical system. In particular, on the object surface, the axial ray, by definition, has zero ray height. This means that on the object surface the Lagrange invariant is given by

$$L_{ab} = n_1 y_{b0} u_{a1} = h n_1 u_{a1} \tag{3.40}$$

where the height h of the object has been taken to define the height of the chief ray, or b ray. Similarly, by definition, the height of the axial ray on the image plane is also zero, so that the Lagrange invariant there is

$$L_{ab} = n_k y_{bk} u_{ak} = h' n_k u_{ak} \tag{3.41}$$

Equating these two expressions for the Lagrange invariant, we find that

$$h n_1 u_{a1} = h' n_k u_{ak} \tag{3.42}$$

or that

$$m = \frac{h'}{h} = \frac{n_1 u_{a1}}{n_k u_{ak}} \tag{3.43}$$

The lateral magnification is thus equal to the ratio of the optical slope of the axial ray in object space to the optical slope in image space. Thus, both the location and lateral magnification of an image are determined by the axial ray. Moreover, since the magnification depends only on the slope of the ray, it is not necessary to know the precise location of the image plane in order to know the size of the image.

Lagrange's law is one of the most powerful invariance principles of geometrical optics. In addition to its use for finding the magnification of images, it can also be used for example, to find the magnification of pupils, to determine the throughput of an optical system, or the information capacity of a diffraction-limited system having a given aperture.

Paraxial constants

OSLO computes seven numbers, labeled *paraxial constants*, when the **pxc** command is given. Four of these are related to (but are not the same as) the *A*, *B*, *C*, *D* Gaussian constants discussed above.

The effective focal length (*EFL*) is given by

$$EFL = f' = \frac{-eh}{u_b' e + u_a' h} \tag{3.44}$$

where e is the radius of the entrance pupil.

The Lagrange (paraxial) invariant (*PIV*) is

$$PIV = L_{ab} = h n u_a \tag{3.45}$$

Corresponding to the lateral magnification is the longitudinal magnification $LMAG = (n'/n)(m_1 m_2)$. If the refractive indices are equal in object and image space and the object is small, then $LMAG = m^2$.

The lateral magnification (*TMAG*) is computed using Lagrange's law, as described above.

$$TMAG = m = \frac{nu_a}{n'u'_a} \tag{3.46}$$

The Gaussian image height (*GIH*) is computed as

$$GIH = h' = \frac{nu_a}{n'u'_a}h \tag{3.47}$$

Three additional items are printed with the paraxial constants. The numerical aperture (*NA*) of the system in image space is defined by

$$NA = n'\sin U' \tag{3.48}$$

In computing the *NA*, OSLO *assumes* that the system is aplanatic, in which case the *NA* can be found from the Abbe sine condition. Then

$$NA = \frac{NAO}{TMAG} \tag{3.49}$$

where *NAO* is the numerical aperture in object space, which can be determined from

$$NAO = n\sin U = \frac{nu_a}{\sqrt{1+u_a^2}} \tag{3.50}$$

The *f*–number (*FNB*) is defined as

$$FNB = \frac{1}{2\ NA} \tag{3.51}$$

The *f*–number *FNB* reduces to the common focal ratio *f*/*d* for a lens used at infinite conjugates, but for finite conjugates it gives what can be called the *working f*-number. The *f*–number and other quantities that measure aperture are discussed in more detail in the next chapter.

The Petzval radius (*PTZRAD*) is computed from the surface curvatures and refractive indices. The Petzval radius is the radius of curvature of the surface on which an image of an extended plane object would be formed, if the imaging system has no astigmatism (called an *anastigmat*). Although the Petzval radius is not really a paraxial quantity, it is a fundamental performance indicator and is easily computed. If there are *k* surfaces, the Petzval radius is given by

$$PTZRAD = \left[n'_k \sum_{j=1}^{k-1} \frac{\left(n_j - n'_j\right)c_j}{n_j n'_j} \right]^{-1} \tag{3.52}$$

Note that all of the above paraxial constants are independent of the location of the final image surface.

Image equations

The theory of matrix optics can be effectively applied to imaging between an object point at *h* and an image point at *h*′. Consider the case where object and image distances are measured from the principal points, rather than the first and last surfaces of the system. The transfer matrix linking two conjugate planes can be written as

$$\begin{bmatrix} h' \\ n'u' \end{bmatrix} = \begin{bmatrix} m & 0 \\ -\phi & 1/m \end{bmatrix} \begin{bmatrix} h \\ nu \end{bmatrix} \tag{3.53}$$

where *m* is the lateral magnification of the system, and ϕ is the power of the system.

The principal points, as discussed before, are the conjugate points of unit, positive, lateral magnification. It follows that a ray entering the first principal plane will be transferred to the second principal plane by an object-image matrix having a magnification of +1. The transfer matrix between the principal planes must be:

$$\begin{bmatrix} 1 & 0 \\ -\phi & 1 \end{bmatrix} \tag{3.54}$$

Let s be the distance from the object to the first principal plane, and let s' be the distance from the second principal plane to the image. The overall matrix is given by

$$\begin{bmatrix} h' \\ n'u' \end{bmatrix} = \begin{bmatrix} 1 & s'/n' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\phi & 1 \end{bmatrix} \begin{bmatrix} 1 & s/n \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h \\ nu \end{bmatrix} \tag{3.55}$$

Carrying out the indicated multiplication, one finds that

$$\begin{bmatrix} h' \\ n'u' \end{bmatrix} = \begin{bmatrix} 1-(s'/n')\phi & s/n+s'/n'-\left[(ss')/(nn')\right]\phi \\ -\phi & 1-(s/n)\phi \end{bmatrix} \begin{bmatrix} h \\ nu \end{bmatrix} \tag{3.56}$$

The equation determining the image distance in terms of the object distance is found by setting the upper right-hand element to zero:

$$\frac{s}{n}+\frac{s'}{n'}-\frac{ss'}{nn'}\phi = 0 \tag{3.57}$$

which becomes, upon dividing by $(ss')/(nn')$,

$$\frac{n}{s}+\frac{n'}{s'} = \phi = \frac{n'}{f'} = \frac{n}{f} \tag{3.58}$$

The lateral magnification of the system is determined by the diagonal elements of the overall matrix:

$$m = 1-\frac{s'\phi}{n'} = 1-\frac{s'}{f'} \tag{3.59}$$

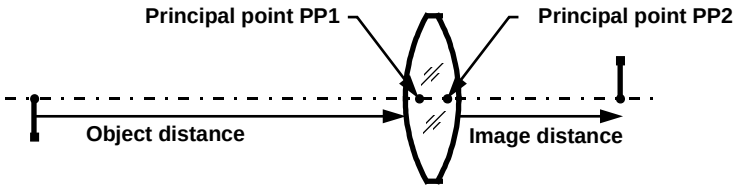
$$\frac{1}{m} = 1-\frac{s\phi}{n} = 1-\frac{s}{f} \tag{3.60}$$

The formulas relating the object and the image thus assume a simple form (often called the Gaussian form) when the object and image distances are measured from the principal planes. If the object and image are both in air, the Gaussian image equation reduces to the well-known equation

$$\frac{1}{s}+\frac{1}{s'} = \frac{1}{f} \tag{3.61}$$

Specification of conjugate distances

In OSLO, it is possible to specify object and image distances referred to cardinal points, rather than physical lens surfaces.



Conjugates	
Object dist	150.000000
Object to PP1	152.190481
Gaus img dist	72.667075
PP2 to image	74.857557
Magnification	-0.491868

The location of the principal points (*PP1* and *PP2*) is calculated using the paraxial properties spreadsheet. The spreadsheet allows you to enter a value for any one of the items shown in the figure. The other items are then calculated automatically. The figure shows typical data for a thick lens having a focal length of about 50 mm. According to the spreadsheet, the first principal point is located 152.19 – 150 = 2.19 mm to the right of the first lens surface, while the second principal point is located 74.86 –72.67 = 2.19 to the left of the last surface.

When a lens is set up using the conjugates spreadsheet, the object and image distances are set automatically if necessary. However, this relationship is not maintained if other data are changed later.

Lens setup

We have seen in the preceding sections that paraxial ray tracing can provide a great deal of information about an optical system. Using a contemporary desktop computer, a complete paraxial analysis of a typical system can be accomplished in a matter of milliseconds. In OSLO, whenever any lens data is entered or updated, a paraxial ray trace is carried out automatically. This process is called *lens setup*. The lens setup routine allows OSLO to integrate paraxial optics with lens data entry. That is, lens data such as curvatures or thicknesses can be specified either directly, or indirectly in terms of paraxial ray data. Whenever the lens setup routine is executed, the indirect data specifications are used to compute the actual values of the lens data.

We have already mentioned one example of indirect data specification in the section on *ynu* ray tracing: the paraxial solves used to specify curvatures and thicknesses according to ray data. OSLO includes both angle and height solves for the axial and chief rays, as well as curvature solves that establish aplanatic points or concentric geometries. Although there are several types of solves available, three types account for the majority of applications.

An axial ray height solve (PY solve) is very commonly used to specify the last thickness (the one before the image surface). By setting PY = 0 on the image surface, the surface is automatically located in the paraxial image plane. Then, no matter how the lens is changed, it is refocused automatically by the solve constraint.

A chief ray height solve (PYC solve) can be used in image space similarly to the axial ray height solve. In this case, the final surface becomes the paraxial exit pupil.

An angle solve on the last surface is used very frequently to constrain the *f*–number of a lens. For example, suppose you want to use the curvature of the last surface to constrain the *f*–number to *f*/2. Assuming that the object is at infinity, this means that the slope of the axial ray in image space must be –0.25. Using an angle solve PU –0.25 to specify the curvature of the last surface will impose the necessary constraint.

Angle solves also have the effect of constraining the focal length of a system. This is because the entrance beam radius is fixed. Thus in the above system, if the entrance beam radius is 12.5, the focal length must be EFL = 12.5/0.25 = 50.0. No matter how the curvatures or thicknesses of the lens are changed, the focal length will always be held to this value. This technique is often used to hold the focal length of a lens during optimization.

A second type of indirect data specification is a link that specifies one data item in terms of another one; this is called a *pickup*. Pickups are used to maintain relationships between elements in systems containing mirrors, or to establish other geometrical constraints (e.g., an equiconvex lens). Although pickups do not involve paraxial ray tracing, they are resolved in the lens setup routine at the same time as solves.

Solves and pickups are examples of what may be called *paraxial constraints,* which, since they are resolved each time that any data are changed, establish OSLO as an *interactive* program that responds automatically to data updates. Paraxial constraints can be distinguished from more general constraints that are based on exact ray data. General constraints are ones included in optimization error functions, and require an optimization cycle to resolve. Paraxial constraints are separated from general constraints because they involve easily computed, linear relationships. Paraxial equations almost always have a solution, unlike exact ray equations, which are nonlinear, and often fail due to rays missing surfaces or total internal reflection.

In actuality, paraxial constraints can fail at a few singular points, such as when an axial ray angle solve is placed on a dummy surface, or a height solve is specified in a region where the paraxial ray travels parallel to the axis. For this reason, it is a good idea to enter lens data in an order that prevents these conditions from happening during data input. For example, you should enter glass data before specifying a surface by an angle solve.

Other actions carried out during lens setup include updating of refractive index data, computation of default apertures, and pre-calculation of global items such as rotation and translation matrices that are used throughout the program. Lens setup is the most commonly executed part of OSLO, and hence is probably its most important routine.

Chapter 4 Stops and Pupils

A pinhole camera is a perfect optical system. If the pinhole is sufficiently small, the resolution will be limited only by diffraction. Unfortunately the image will be rather dim. The angular resolution will be

$$\Delta\theta \approx \frac{\lambda}{d_{pinhole}} \tag{4.1}$$

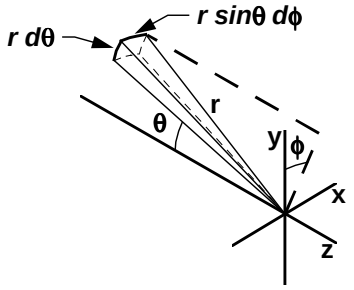
Curiously, this relation indicates that the larger the hole, the better the resolution. Of course, to accommodate a larger pinhole, the back focus must also become large, and the camera soon grows to unwieldy size. However, we can put a lens behind the pinhole to effectively move the image plane closer to the pinhole. Now we can make the hole as large as we please to increase the irradiance of the image, and at the same time increase the resolution!

Obviously the aperture of an optical system is one of its most important attributes. The limit on aperture size is caused by the fact that the image quality will deteriorate if the aperture becomes too large. The optical designer's task is to balance the complexity of the design with the requirements for a specified aperture.

Radiometric concepts and terminology

The amount of light that passes through an optical system is a radiometric issue. Although the overall field of radiometry is extensive, we are concerned only with the basic laws and geometrical relationships that are important for optical design. The fundamental terms and concepts used in radiometry are geometrical in character. When it is necessary to apply these geometric concepts to a particular system, the subject is called *photometry* if the system is visual, or *radiometry* if the system is physical. For simplicity, we use only the radiometric terms.

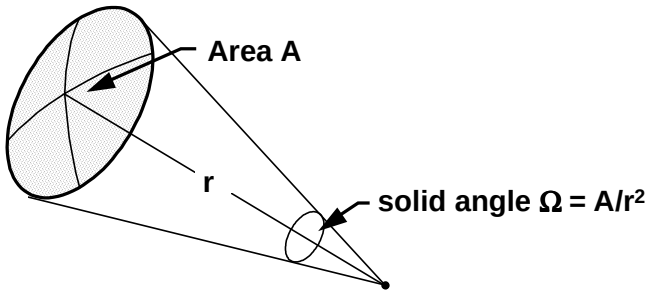
For our discussion, it is convenient to employ a spherical coordinate system.



In the coordinate system shown, the differential element of area is

$$dA = r^2 \sin\theta d\theta d\phi \tag{4.2}$$

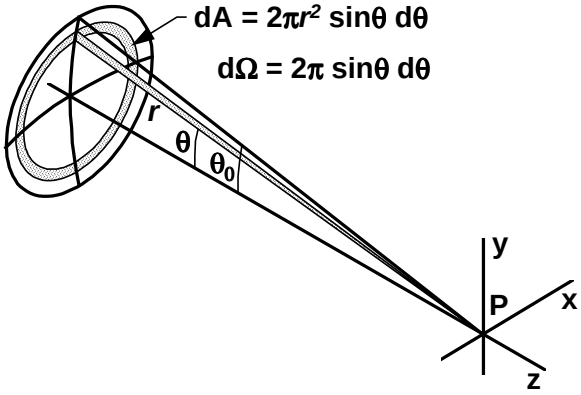
The solid angle Ω, subtended by some area *A*, as seen from a point, is defined to be the ratio of the area of a spherical surface bounded by the edge of the area *A*, divided by the square of the radius of the sphere.



In the spherical coordinate system shown above, the differential element of solid angle is

$$d\Omega = \frac{dA}{r^2} = \sin\theta d\theta d\phi \tag{4.3}$$

Many problems have azimuthal symmetry. In these problems, it is convenient to consider the differential element of solid angle to be an annular cone, as shown below



If the surface is circular, and subtends an angle θ_0 as seen from *P*, then

$$\Omega = 2\pi\int_0^{\theta_0}\sin\theta d\theta = 2\pi\left(1-\cos\theta_0\right) = 4\pi\sin^2\frac{\theta_0}{2} \tag{4.4}$$

Ordinary radiometry is concerned with the propagation and measurement of radiation from incoherent light sources. To discuss light from incoherent sources, we introduce the following terminology:

1. The time rate of change of energy is called the *flux*, denoted by Φ. In MKS units, the unit of flux is the watt.

2. The flux per unit area impinging on a surface is called the *irradiance* of the surface, denoted by *E*. In MKS units, the unit of irradiance is watts/m^2.

3. The total flux per unit solid angle coming from a small source is called the *intensity*, denoted by *I*. It is assumed in the definition of intensity that the size of the source is small compared to the distance to the point where the intensity is measured. In MKS units, it is measured in watts/sr. A uniform point source is one that radiates uniformly in all directions. For such a source, the intensity is

$$I \equiv \frac{d\Phi}{d\Omega} = \frac{\Phi}{4\pi} \tag{4.5}$$

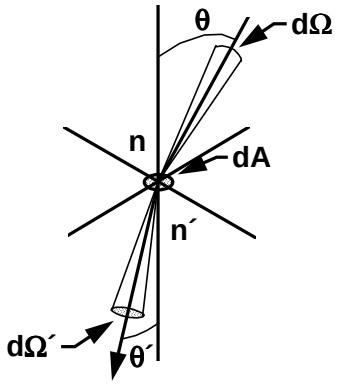
4. To discuss an extended source, we need a term to describe the intensity per unit area of the source. This general quantity is called the *radiance* of the source, defined to be the flux emitted per unit solid angle, per unit area projected in the direction of observation, and denoted by *L*.

$$L \equiv \frac{d^2\Phi}{dA\cos\theta d\Omega} \tag{4.6}$$

A source having a radiance independent of direction is called a *Lambertian* source. Blackbody sources are Lambertian sources, and most diffusing surfaces behave approximately like Lambertian sources.

Radiance conservation theorem

It can be shown that radiance is conserved along any tube of rays propagating through an optical system. Consider a boundary separating two media of refractive index n and n', as shown below



Let dA be an infinitesimal area on the boundary. The flux on dA from the tube of rays contained within the solid angle $d\Omega$ is (cf. Eq. (4.3))

$$\frac{d^2\Phi}{dA} = L\cos\theta d\Omega \tag{4.7}$$

This flux will be transmitted into the second medium, where the radiance will be

$$L' = \frac{d^2\Phi}{dA\cos\theta' d\Omega'} \tag{4.8}$$

It follows that the radiance along the tube of rays in the second medium is related to the radiance along the tube of rays in the first medium by

$$L' = L\frac{\cos\theta d\Omega}{\cos\theta' d\Omega'} \tag{4.9}$$

Now

$$\frac{d\Omega}{d\Omega'} = \frac{\sin\theta d\theta d\phi}{\sin\theta' d\theta' d\phi'} \tag{4.10}$$

Since the incident and refracted rays are co-planar, we have

$$d\phi = d\phi' \tag{4.11}$$

Differentiating Snell's law, we find

$$\begin{aligned} n\sin\theta &= n'\sin\theta' \\ n\cos\theta d\theta &= n'\cos\theta' d\theta' \end{aligned} \tag{4.12}$$

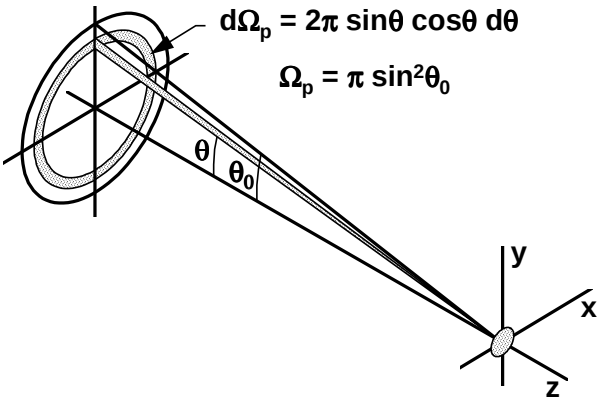
Using these equations in Eq. (4.9) yields

$$L' = L\left(\frac{n'}{n}\right)^2$$
$$\frac{L'}{n'^2} = \frac{L}{n^2} \tag{4.13}$$

so that radiance, divided by the square of the refractive index, is conserved along a tube of rays. This result is called the *Radiance Conservation Theorem*. It implies that it is impossible to increase the radiance of an incoherent source using a (passive) optical system such as a lens. In forming an image, a lens can only increase the apparent solid angle of a source, not its radiance.

Irradiance by a flat circular source

Let us now consider the irradiance of a small flat target due to a flat circular source having a uniform radiance L, and subtending an angle θ_0.



From the definition, Eq. (4.6), of radiance,

$$\frac{d^2\Phi}{dA} = L\cos\theta d\Omega \tag{4.14}$$

so the irradiance is

$$E = \frac{d\Phi}{dA} = 2\pi L\int_0^{\theta_0}\sin\theta\cos\theta d\theta$$
$$= L\pi\sin^2\theta_0 \tag{4.15}$$

The quantity $\pi\sin^2\theta_0$ is sometimes called the projected solid angle Ω_p subtended by the source. Thus

$$E = L\Omega_p \tag{4.16}$$

This result is the basic equation used to calculate the irradiance on a surface. It states that the irradiance is equal to the source radiance times the projected solid angle subtended by the source, as seen from the surface. Thus, to calculate irradiance, we imagine that we are on the surface, and look back at the source to determine the solid angle subtended by it.

In OSLO, the quantities that are used to describe the convergence of light beams are the numerical aperture, abbreviated *NA*, and the *f*–number, or relative aperture, abbreviated *FNB*. The numerical aperture is defined by

$$NA = n\sin\theta_0 \tag{4.17}$$

where n is the refractive index. The *f*–number is defined by

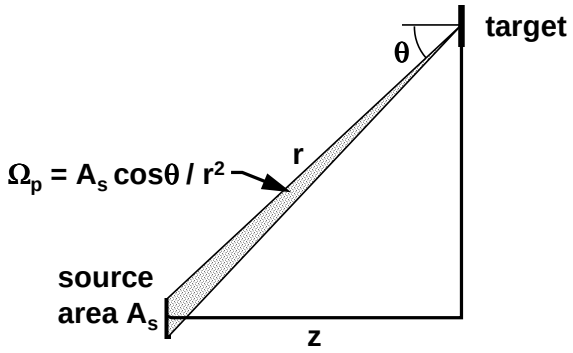
$$FNB = \frac{1}{2\,NA} \tag{4.18}$$

Using these definitions in Eq. (4.15), we find

$$\begin{aligned} E &= \frac{\pi L\, NA^2}{n^2} \\ &= \frac{\pi L}{4n^2 FNB^2} \end{aligned} \tag{4.19}$$

Cos^4 law

For a non-uniform or an off-axis system, the integration required in Eq. (4.15) must generally be done numerically. There is one simple off-axis case to consider, which approximates the situation when the source is small, as shown below.

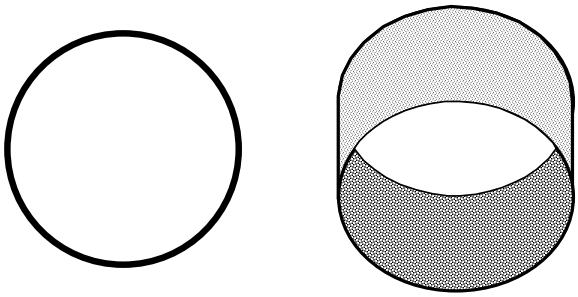


In this situation, the projected solid angle Ω_p of the source decreases as $\cos^3\theta$, since the projected area of the source as seen from the target decreases as $\cos\theta$, and the distance r from the source to the image increases as $\cos^2\theta$. In addition, the projected area of the target decreases as $\cos\theta$, so the overall irradiance of the target becomes

$$I = I_0 \cos^4 \theta \tag{4.20}$$

Vignetting

Many optical systems, used off axis, do not obey the $\cos^4$ law, because they suffer from vignetting. Vignetting is a term that refers to the reduction in aperture of a system caused by the axial separation of two apertures. A simple example of vignetting is shown by a short length of pipe viewed at an oblique angle, as shown in the drawing below. The cat's-eye aperture of the view on the right is typical of lens vignetting. Of course, in a lens, the aperture of each element is imaged by the lenses between it and the observer, often with different magnification.

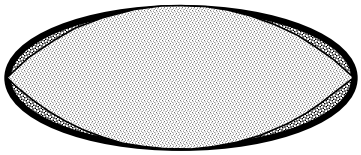


Although vignetting might appear to be undesirable from a radiometric point of view, it is often a useful way for the designer to control aberrated rays that occur near the edges of lenses. Some lenses, such as the Cooke triplet, depend on vignetting to achieve satisfactory performance.

A general estimate of vignetting can be obtained from paraxial ray data. If a chief ray is traced through a system from an off-axis object point, then if the system is not to have any vignetting, the aperture radius of each lens must be at least as big as the sums of the absolute value of the axial and chief ray heights on the lens. OSLO uses this fact to compute the default aperture radii of lenses in the case where no other value is specified.

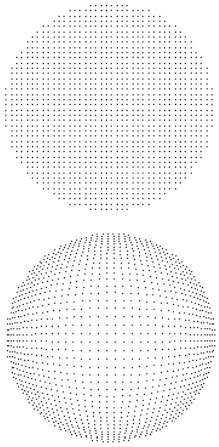
Computation of irradiance

To compute irradiance, OSLO uses the model described above, where the solid angle subtended by the apparent source is determined in image space, and multiplied by the radiance of the source to determine the irradiance. OSLO only computes irradiance relative to the on-axis value, so the absolute value of the source radiance is not used by the program. Two models are used. In the first, the actual pupil is approximated by an ellipse that passes through its top, bottom, and maximum *x* extent.



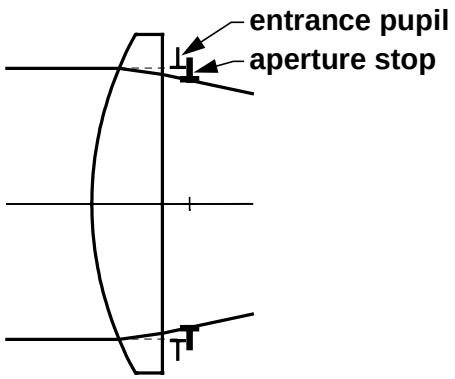
In the second, only available in OSLO Premium, a grid of rays is traced through the optical system. The grid is set up so that each ray subtends an equal amount of solid angle in image space, and is weighted according to the solid angle that it subtends in object space. This algorithm makes the irradiance of a small area in image space proportional to the weighted sum of the number of rays that fall on it.

To achieve accurate results from illumination calculations of the type used in OSLO, it is necessary to use aplanatic ray aiming rather than paraxial ray aiming, at least for systems involving finite conjugates. With aplanatic ray aiming, fractional coordinates are based on numerical aperture, while with paraxial ray aiming, fractional coordinates are based on the entrance beam radius. The figures below compare exit pupil coordinates for a perfect lens having a numerical aperture of 0.8, with default (aplanatic) ray aiming (below), and with paraxial ray aiming (next page).



Stops and pupils

Consider a lens with an iris diaphragm behind it, as shown below. The diaphragm limits the diameter of the beam that can pass through the lens, and is called the *aperture stop*.



The apparent aperture stop, as seen from object space, is called the *entrance pupil*. The apparent aperture stop, as seen from image space, is called the *exit pupil*. In the drawing, the exit pupil is the aperture stop itself, since there are no lenses following the stop. In general, since the entrance pupil is the image of the aperture stop in object space, and the exit pupil is the image of the aperture stop in image space, the exit pupil is the image of the entrance pupil by the entire system.

In the system shown above, if the object is somewhat off the axis, the beam diameter will still be limited by the aperture stop. The ray from an off-axis object point through the center of the aperture stop is called the *chief ray*. If the object is extremely far off axis, the edge of the lens will limit one side of the beam, and the aperture stop the other. Then we will have a vignetted pupil, as described above. With a vignetted pupil, the chief ray is generally not the central ray of an off-axis bundle.

In old optics literature, the term *field stop* is used to describe a surface that limits the field of view of a system, and the *entrance window* and *exit window* are defined to be the image of the field stop in object and image space. Field stops are commonly found in visual relay systems that include field lenses and reticles, or infrared systems where internal baffles are used to control stray light. However, the concept is not useful in many systems where there is no surface within the system that serves as a field stop. Then the field of view is limited by the edge of the object, the edge of the image, or some vignetting condition that occurs within the lens.

The paraxial entrance and exit pupils serve to define the effective aperture of a simple system whose behavior is adequately characterized using paraxial optics. In systems containing tilted and decentered elements, paraxial optics may not exist. Even within the realm of paraxial optics, complications occur for systems in which a single surface does not serve as the aperture stop. For example:

In a system containing cylindrical lenses, the surface that limits the *yz* beam may be different than the one that limits the *xz* beam.

The effective pupil off axis is often determined by one aperture for the lower portion of the beam and a different one for the upper portion of the beam.

In a zoom system, the aperture stop may depend on the magnification of the system.

In real systems, it is often necessary to consider pupil aberrations. For example, in wide-angle systems, aberrations often distort and move the actual pupil so much that real rays launched towards the paraxial pupils can not even pass through the system. The actual pupil may appear to rotate towards the observer, and may grow in size as the field angle is increased.

Specifying aperture and field of view

Although the individual surface apertures determine the system performance, it is preferable to specify the aperture and field of view of the overall system independently of the apertures during the design phase. The reason for this is that the apertures do not change the trajectories of rays that are that are used during automatic design to optimize a system; apertures can only block or transmit rays.

It takes a minimum of two items, which we call paraxial operating conditions, to specify the aperture and field of a system. In OSLO, both the aperture and field can be specified in several ways. The paraxial data spreadsheet can be used to enter the required data. In the spreadsheet, the program continually recalculates all data so that conflicting specifications are automatically resolved. The figure below shows typical data for a Cooke triplet lens set to work at a magnification of –0.5, a numerical aperture of 0.1, and an image height of 18 mm.

Aperture		Field		Conjugates	
Entr beam rad	6.837144	Field angle	13.757350	Object dist	136.571844
Object NA	0.050000	Object height	-36.000000	Object to PP1	150.001623
Ax. ray slope	-0.100125	Gaus image ht	18.000000	Gaus img dist	68.080824
Image NA	0.100000			PP2 to image	75.000812
Working f-nbr	5.000000			Magnification	-0.500000
Overall lens length		17.000000	Total track length		221.652669
Entr pup rad/pos	7.361115	10.466307	Ext pup rad/pos	7.824886	-10.070166
Lagrange invariant		-1.802254	Petzval radius		-149.381547
Effective focal length		50.000541			

In the spreadsheet, the top five rows contain data entry fields, while the bottom four rows contain calculation fields. The items in the conjugates column have been described in the previous chapter. The items in the aperture and field columns, and their values, require some explanation.

The specification of aperture in OSLO is somewhat complicated by the fact that OSLO assumes that all systems are aplanatic rather than paraxial, even when setting up the paraxial data. This assumption allows OSLO to more accurately describe systems that work at finite conjugates than it would if aperture specifications were strictly paraxial. In the current system, the image *NA*, defined as $n'\sin\theta'$, is set to 0.1. The magnification of the system is –0.5. Since the system is assumed to be aplanatic, we have

$$m = \frac{n\sin\theta}{n'\sin\theta'} \tag{4.21}$$

which implies that the *NA* in object space must be 0.05. Now if the *NA* in object space is 0.05, the actual angle of the axial ray entering the system must be $\theta = \sin^{-1}(0.05) = 2.865984$ degrees. The slope of the entering ray must therefore be PU = tan(2.865984°) = 0.050063. Since the magnification is –0.5, the slope of the emerging axial ray must be PU = –0.100125. As described before, the focal ratio, or *f*–number, used in OSLO is the working *f*–number, defined as 1/(2 *NA*), or 5.0, in the present case.

Note that the aperture can be specified by the “entrance beam radius,” but not the entrance pupil radius. The entrance beam radius is the radius of the beam on the first surface of the system. If the object is at infinity, the entrance beam radius is equal to the entrance pupil radius, but at finite conjugates they are different. The reason for using the entrance beam radius is that, if the aperture stop is inside the lens, the entrance pupil moves and changes size as the lens curvatures and thicknesses are changed. Moreover, the entrance pupil is often ill-defined in a real system because of aberrations or vignetting. The entrance beam radius, on the other hand, is a known quantity that is readily understood. The entrance pupil location and size are automatically computed and displayed in the spreadsheet as an information item.

Although any of the five items on the spreadsheet can be used to specify the aperture of a system used at finite conjugates, the default specification is the numerical aperture in object space. On the other hand, this item cannot be used when the object is at infinity, and the entrance beam radius is the default for infinite-conjugate systems.

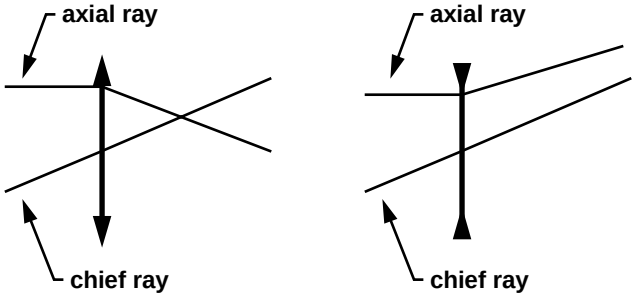
In OSLO the field of view of an optical system can be specified by the object height, the image height, or the field angle. The field angle is the angle of a ray from the edge of the object to the center of the entrance pupil (not the center of the first surface). The field angle is not useful for finite-conjugate systems, since it changes as the entrance pupil moves.

Optical system layout

Optical system layout is too big a subject to be covered in detail in this manual. However, it is important to understand a few fundamental principles in order to work sensibly with OSLO, particularly those that concern the trajectories of the axial and chief rays. Hence we review a few common systems.

Thin lens

A thin lens is one having zero thickness. Naturally such a lens does not actually exist, but the effects of the thickness of a lens are often negligible, and the elimination of thickness as a lens parameter simplifies many optical equations. Thin lenses are often drawn as double-headed arrows, with the arrows pointing out to indicate a positive lens, or pointing in to indicate a negative lens.



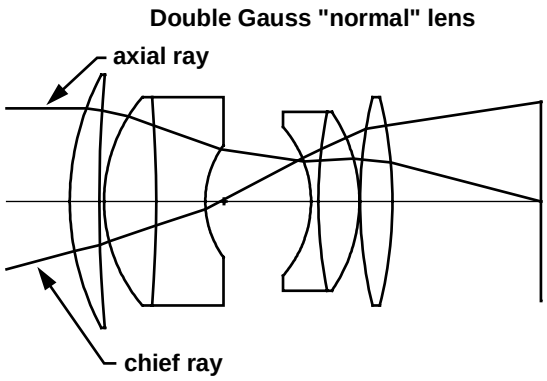
For a single thin lens, the aperture stop is coincident with the lens, so that the axial ray is the ray from the vertex of the object that passes through the edge of the lens, and the chief ray is the ray from the edge of the field that passes through the center of the lens. More complicated thin-lens systems may contain several thin lenses, so that the image of the stop by all of the lenses must be found to determine the locations of the pupils, prior to determining the trajectories of the axial and chief rays.

Photographic objective

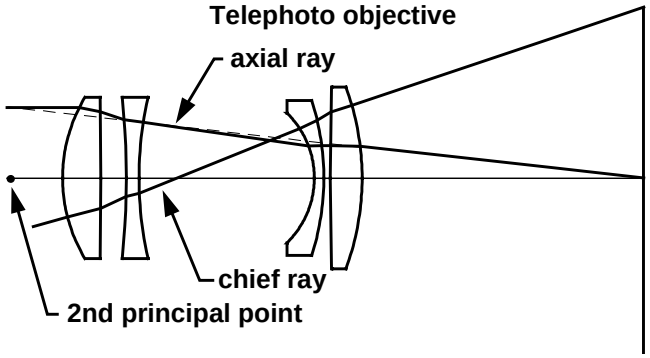
The photographic objective is a type that includes an enormous number of examples from relatively simple lenses such as the Cooke triplet, to very complicated photolithographic systems that may contain dozens of elements. An important characteristic of a photographic objective is that it has $\tan\theta$ mapping. That is, the image height is proportional to the tangent of the incoming chief ray, so that the image will have the same shape as the object. The difference between the actual mapping and $\tan\theta$ mapping is called *distortion*.

An example of a typical photographic lens is the double Gauss objective. This design is the most common type of high-speed lens used for normal focal length 35 mm cameras, typically working at a speed of about *f*/2. Like other lenses, it works by balancing aberrations introduced by positive and negative surfaces. The double Gauss lens is approximately symmetrical, with the aperture stop between the two negative central elements.

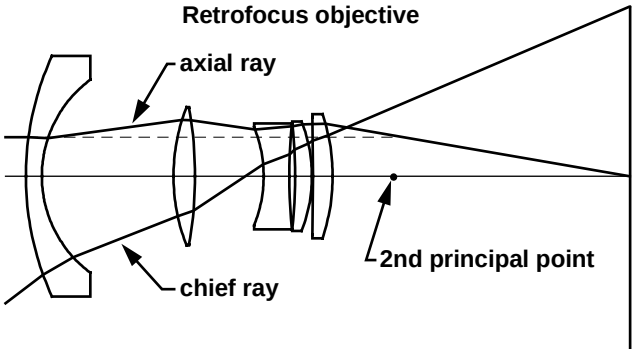
The trajectories of the axial and chief rays are important for balancing aberrations. To a first approximation the axial ray goes through the edge of the overall lens, and the chief ray goes through the center. In detail, the ratio of the height of the chief ray to the height of the axial ray on a particular surface is an indicator of the effectiveness of the surface in controlling off-axis aberrations.



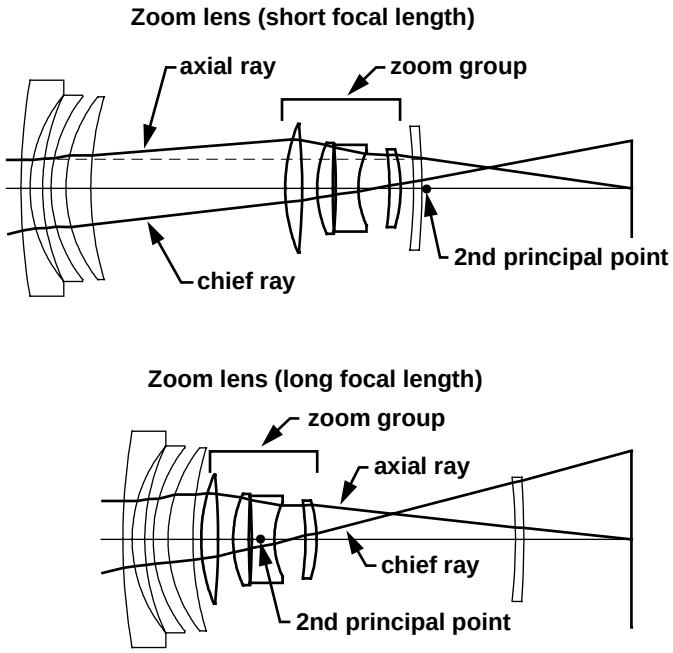
If the positive power in a lens is concentrated more towards the front of the lens than the rear, the principal points will be shifted towards the front, and the lens becomes a *telephoto* lens, in which the effective focal length is longer than the distance from the center of the lens to the focal point, as shown below.



On the other hand, if the positive power is concentrated towards the rear of the lens, the focal length will be shorter than the distance from the center of the lens to the focal point, and the lens will be called a *retrofocus lens.*



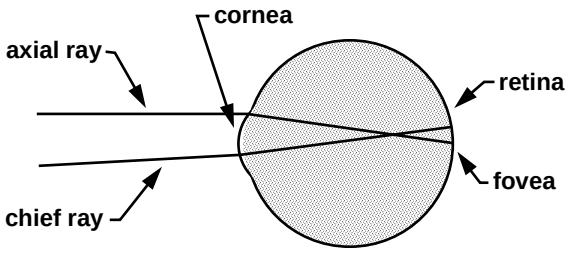
Many, if not most, photographic objectives are *zoom* lenses, in which the focal length is varied by moving one or more groups of elements within the lens, as shown below.



In the first view, the central four-element zoom group is moved towards the back of the lens, concentrating the power towards the rear, and making the lens behave as a retrofocus design. In the second view the zoom group is moved towards the front, shifting the power towards the front, making the lens more like a telephoto. Note that both the internal location of the zoom group and the back focal length are changed as the lens is zoomed. As drawn, the image height is variable, but in use it would be more common to fix the image height and vary the field of view.

Magnifier

The human eye, although it cannot be designed itself, forms an important part of any overall visual system. The refraction of light in the eye occurs largely at its outer surface, called the cornea. The image is formed on the back of the eye, on an array of light-detecting rods and cones that make up the retina. The outer portions of the eye, which have mostly rods as detectors, do not contribute to high-acuity vision. The portion of the retina that has the highest acuity is called the fovea, and the eye normally pivots in its socket to bring the image of the point of visual interest into the foveal region.

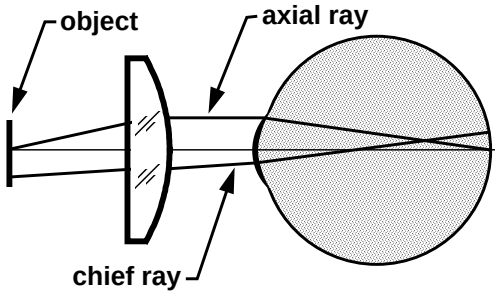


Because of the eye's ability to rotate in its socket, the axial ray and the chief ray for the eye are not precisely defined. The axial ray is the ray from the vertex of the object through the edge of the entrance pupil of the eye, which is the image of the iris of the eye by the cornea. The chief ray is sometimes taken to be a ray from the intended field of view through the center of rotation of the eye, since the eye will automatically pivot to view an off-axis image. Care must be taken in the design of any visual instrument to ensure that the pupil is large enough to accommodate the motion of the iris that occurs when the eye pivots.

A simple magnifier is a lens that is used in conjunction with the eye, to extend the accommodation range of the eye to permit close-range viewing of small objects. The near focus point of the human

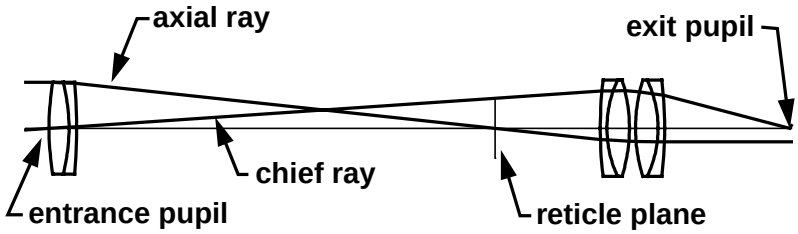
eye ranges from approximately 10 cm for children to several meters for people past the age of 60; a value of 25 cm is used as a reference in computing the power of a magnifier. Thus a magnifier with a focal length of 25 mm would be called a 10× magnifier.

The drawing below shows typical use of a magnifier. As shown, the axial ray emerges from the magnifier parallel to the axis, indicating that the focal point is at infinity. Such a system is sometimes termed *afocal* on the image side. In actual use, the layout is up to the user, who adjusts it for maximum visual comfort, usually keeping the object somewhat inside the focal point.



Telescope

The inverting telescope is one form of a family of optical systems known as compound magnifiers. Such systems include an objective that forms an aerial image, and an eyepiece that serves as a magnifier to view the image. The objective serves as the aperture stop. The entrance pupil is thus located in the plane of the objective, and the exit pupil is located in the plane where the objective is imaged by the eyepiece. The eye of the observer should be placed in the exit pupil of the instrument, so that the entire field can be viewed without vignetting. The distance between the rear surface of the eyepiece and the exit pupil plane is called the eye relief.

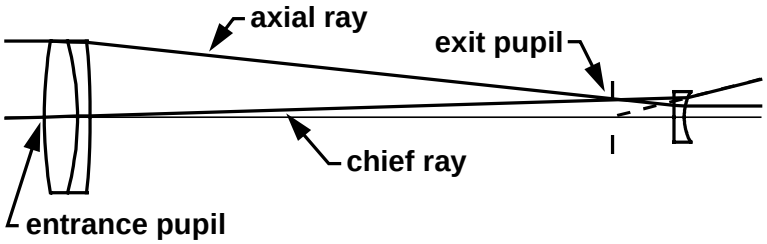


The axial ray enters the objective parallel to the axis and emerges from the eyepiece parallel to the axis, so that the image can be viewed with the relaxed eye. The chief ray passes through the center of the objective and the edge of the eyepiece. From the figure, it can be seen that the field of view of the telescope is limited by the diameter of the eyepiece. The chief ray trajectory shows that the telescope forms an inverted image. From Lagrange's Law, it follows that the angular magnification of the telescope is proportional to the ratio of the entrance and exit pupil diameters.

The above telescope is termed *afocal* because both the principal planes and focal points are at infinity. OSLO has a special mode of evaluation that displays ray data for afocal systems in angular rather than linear format. In afocal mode, the paraxial constants are displayed as follows.

```
*PARAXIAL CONSTANTS
   Angular magnification:     -3.39831    Paraxial invariant:        -1.04890
   Eye relief:                46.01604    Petzval radius:           -44.23070
```

An alternate form of telescope is the Galilean telescope shown below.



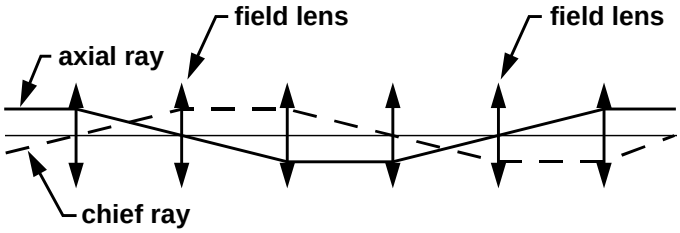
The Galilean telescope differs from the inverting telescope, in that it forms an erect image. The paths of the axial and chief rays are shown in the figure. The axial ray enters the system parallel to the axis, and leaves the system parallel to the axis, so the system is afocal.

The objective serves as the aperture stop, but since the eyepiece has a negative focal length, the chief ray diverges after passing through the eyepiece, placing the exit pupil inside the telescope. Since it is impossible to put the observer's eye in the exit pupil of the instrument, the field of view will be limited by the diameter of the pupil.

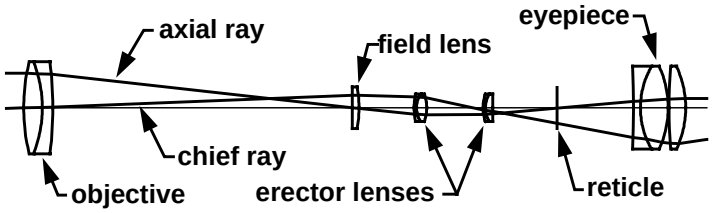
Galilean telescopes are occasionally used for visual instruments having low power (e.g. opera glasses), but the negative eye relief is a severe shortcoming that restricts the usefulness of the system at high powers. The Galilean system has considerably shorter overall length than the inverting system, and is often used for laser beam expanders, where its narrow field is of no consequence.

Relay system

Telescopes are often coupled to relay systems that transmit images to a different location for viewing. The paraxial layout for a typical relay system is shown below. The axial and chief ray trajectories are the same in image space as in object space, but are shifted to the right by the length of the relay. Relay systems consist of alternate sequences of objectives and field lenses that bend the chief ray but not the axial ray. In an actual relay system, curvature of field is often a significant problem.



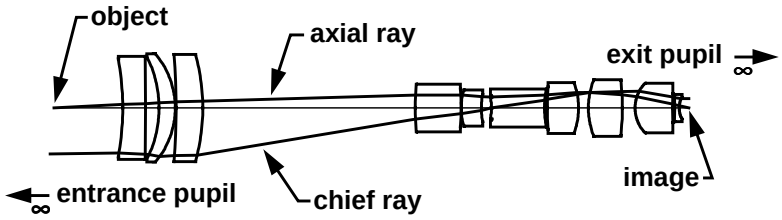
The figure below shows a zoom telescope that contains a relay/erector system. In this system, a field lens at the front of the erector system constrains the chief ray height, and the two small doublets relay the primary image near the field lens to the reticle plane. The position and spacing of the erector lenses can be varied to change the magnification of the relay.



Telecentric lens

An increasing number of lenses are telecentric on the object, image, or both sides. Telecentric means that the chief ray is parallel to the axis, or equivalently that the pupil is at infinity. Such

lenses are useful in various metrology applications, because they have the property that the image height does not change as the lens is focused. The lens below is a typical example that is telecentric on both sides.



Other examples of telecentric lenses include scanning lenses, which are often required to be telecentric to preserve geometrical relationships, and Fourier transform lenses, in which the axial ray in one direction is the same as the chief ray in the other direction.

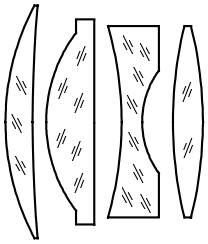
Specifying lens apertures

The amount of light that passes through a real system depends on the actual physical apertures present, not an arbitrary specification of paraxial aperture and field. At the same time, the apertures predicted by paraxial ray tracing can provide an approximate indication of the necessary aperture sizes for many systems that have no vignetting. To accommodate these requirements, OSLO uses two methods for specifying apertures. Apertures may be either *directly specified* by the user, or *solved* using paraxial optics.

The term *aperture* in OSLO is used in a general sense to describe features that limit the extent of an optical surface. Thus a hole in a lens is called an aperture, as is a mask used to block light from passing through some part of a surface. These aperture types are called special apertures and are discussed later. For the present discussion, apertures are considered to be circular and to bound the edges of optical surfaces.

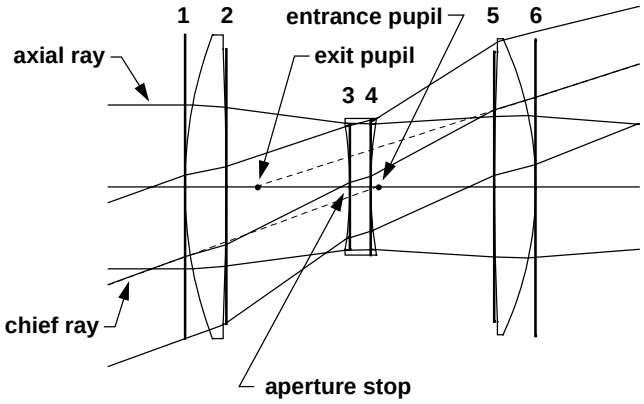
The aperture radius required to pass a beam traveling along the axis is equal to the axial ray height on the lens. If the beam is from an off-axis point, the required aperture radius is the sum of the magnitudes of the axial and chief ray heights. The paraxial algorithm is very fast to evaluate, and is used in OSLO to compute the aperture of solved apertures every time the lens setup routine is carried out.

The paraxial apertures will generally differ on each surface. An actual fabricated lens will normally have edges that are parallel to the optical axis. For drawing pictures, the radius of an element is made equal to the larger of the two apertures. A line perpendicular to the axis is drawn between the smaller aperture and the larger aperture. This leads to elements that appear as follows.



In the drawing, the second lens is impossible to fabricate using traditional grinding and polishing technology, but can be made by molding or other modern methods, such as diamond turning. The third lens has a flat region that is commonly used to reduce the weight of thick negative elements.

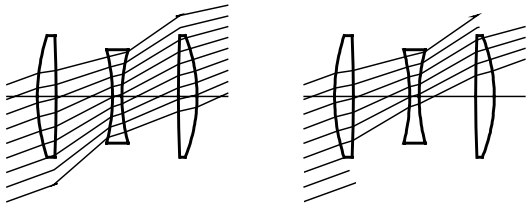
The figure below shows the way that OSLO assigns apertures for a Cooke triplet lens. The paraxial surfaces are shown as straight lines whose length is equal to twice the paraxial aperture, and the actual surfaces are shown as curved lines, with apertures assigned using the solving algorithm described above. The ray trajectories are for real rays. The pupil locations, but not sizes, are shown by dots.



By looking at the figure, we can see which rays are responsible for the apertures, as well as the differences between the paraxial apertures and the actual apertures required to pass the beams. The lower rim ray is responsible for the apertures before the stop, and the upper rim ray for the apertures after the stop, a common, but not necessary, condition. The real chief ray misses the center of the paraxial entrance pupil by a little, indicating some pupil aberration. The upper rim ray misses the paraxial apertures of the last two surfaces by a significant amount, showing that this lens must be made larger than the expected value to pass the entire real beam from the edge of the field.

It should be stressed that aperture values *per se* do not affect ray tracing. In the left-hand figure below, rays are shown passing through a lens outside of the drawn apertures. This is because there is nothing in the ray trace equations that mathematically prevents this from happening. Moreover, for the lower and upper rim rays in the drawing, small glitches can be seen in the ray trajectories at the first and last surfaces. This is caused by the fact that at that distance from the axis, the front surface crosses behind the back surface, a so-called “feathered edge” that is not physically realizable. Again, this does not prevent a solution of the ray trace equations.

In order to exclude rays that fall outside defined apertures, OSLO provides a *checked aperture* type that causes the ray trace to check the ray heights on each surface to see if the ray is outside the aperture, and terminate the trace if it is. The figure on the right below shows what happens when checked apertures are put on the first and last surfaces.



Generally, it is not desirable to put checked apertures on more surfaces than required, since it slows down the ray trace needlessly. There are parts of the program, such as optimization, where checked apertures can inhibit derivative computations and prevent the program from working properly. In fact, OSLO is set up so that apertures are never checked in the DLS optimization routines. There is a general operating condition called **apck** that controls whether rays will be blocked by checked apertures. The action of **apck** is different in different parts of the program, as shown in the following table.

Ray blocking at checked apertures	Apck Off	Apck On
Paraxial routines	no	no
Ray trace evaluation	no	yes
Lens drawings	no	yes
Spot Diagram Setup	yes	yes
DLS Optimization	no	no
ASA Optimization	no	yes

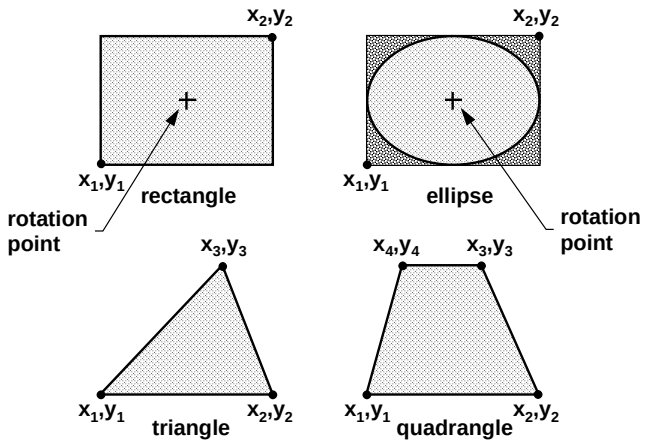
Special apertures

The discussion above concerns only the circular apertures that define the edges of lenses. There is a need for more complicated apertures to describe complex shapes, or other conditions such as central obstructions or holes that affect the amount of light that is transmitted through an optical system. OSLO has what are called special apertures to meet this need.

A special aperture is an aperture that is built up from ellipses, rectangles, triangles, or quadrangles. Rectangles and ellipses are specified by two points that define the bounding box surrounding the shape. Triangles and quadrangles are defined by *x* and *y* coordinates that give their vertices. Rectangular and elliptical apertures can be rotated around their centroids, as shown below.

Obviously, squares and circles are special cases of rectangles and ellipses. Complex aperture shapes can be constructed by grouping and combining several primitive shapes on a single surface. In a complex aperture, the primitive shapes are combined using logical operations to establish the effect of the overall aperture. Each primitive shape can be transmitting or obstructing. Any number of primitives can be placed in a group, and each surface can have any number of groups.

The logical rules for combining primitives within a group are that a ray passes through the group if it passes inside every transmitting primitive, and outside every obstructing primitive. This corresponds to the intersection, or *anding*, of the primitives. The logical rule for a ray passing through an overall surface aperture is that it pass through any group defined on the surface. This corresponds to the union, or *oring* of the groups.

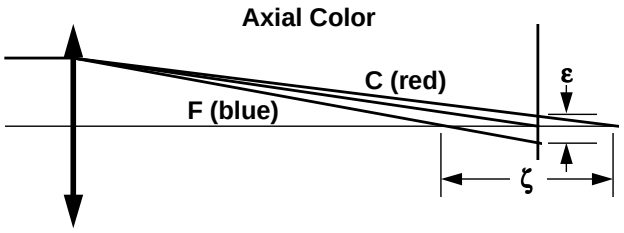


The above primitives and rules allow the construction of almost any aperture. Note that special apertures in OSLO are always checked apertures. Whether or not they block rays depends on the setting of the **apck** operating condition, according to the table above.

Chapter 5 Aberrations

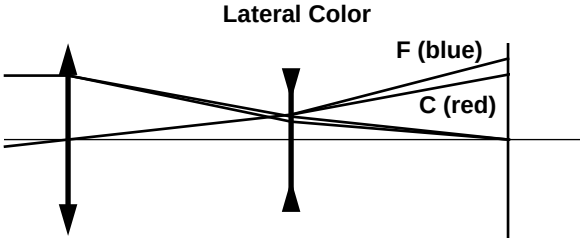
Axial and lateral chromatic aberration

The dispersion of optical glass causes the properties of a lens to be dependent on the wavelength of the light transmitted by it, i.e., to have *chromatic aberration*. There are two types of first-order, or paraxial, chromatic aberrations. One is called *axial chromatic aberration*, or *axial color*, and relates to the ability of a lens to bring light of all wavelengths to a focus in the same plane. Axial chromatic aberration is illustrated in the figure below, where the difference in the position of the focal point of a lens is shown (exaggerated). In the example shown, the blue focus is closer to the lens than the red focus; this is called undercorrected axial chromatic aberration, and is typical of a positive singlet lens.



The magnitude of the axial chromatic aberration of a lens can be specified by either the longitudinal shift in the focal position, ζ, or by the difference in height, ε, on the nominal image plane, between axial rays having long and short wavelengths. This last specification is called *primary axial color* (PAC). If the blue focus is closer to the lens than the red focus, so that the blue ray intersects the image plane at a lower ray height than the red ray as in the figure above, the primary axial color is negative, or undercorrected.

The figure below shows a two-lens system that has been corrected for primary axial color, but which possesses a different type of paraxial chromatic aberration. In the lens shown, the red and blue focus have been made to coincide, but the effective focal length of the lens is different in red and blue light. This difference in focal length causes a chromatic difference in magnification of the lens, called *primary lateral color* (PLC). Lateral color can be measured by giving the difference in ray height on the image plane of chief rays traced through the system in short and long wavelengths. If the red image is larger than the blue image, the aberration is undercorrected and is negative.



In addition to primary axial and lateral color, there is another paraxial chromatic aberration that is often of considerable importance, called *secondary spectrum*. Secondary spectrum arises from the fact that the refractive index of glass does not vary linearly with wavelength. An *achromatic* lens is corrected for chromatic aberration at two wavelengths, but has residual aberration, or secondary spectrum at other wavelengths. In OSLO, secondary spectrum is displayed as secondary axial color (SAC) and secondary lateral color (SLC), which are computed using the refractive index

difference for wavelengths 1 and 2, rather than 2 and 3. Lenses that are corrected for three (or more) wavelengths are called *apochromatic*.

In addition to the paraxial chromatic aberrations, there is a chromatic variation of the normal (monochromatic) aberrations. Of these, the most important is usually the chromatic variation of spherical aberration called *spherochromatism*.

Calculation of chromatic aberration

The amount of chromatic aberration in an optical system can be determined by paraxial ray tracing. Below, we show a paraxial ray trace in three colors for a simple lens, made of BK7 glass.

```
SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    1.7633e+19             AIR

  1     50.731000 F    6.500000 F   15.000000 AF          BK7 F *
  2    -50.731000 F   47.986599     15.000000 F           AIR

  3        --            --          8.847573 S

*REFRACTIVE INDICES
 SRF     GLASS            RN1         RN2         RN3         VNBR        TCE
  1      BK7           1.516800    1.522376    1.514322   64.166410   71.000000

*PARAXIAL TRACE
 SRF      PY          PU          PI          PYC         PUC         PIC
  3       --       -0.298941   -0.298941    8.847573    0.168629    0.168629

*PARAXIAL TRACE (WAVELENGTH 2)
 SRF      PY          PU          PI          PYC         PUC         PIC
  3    -0.157135   -0.302119   -0.302119    8.842187    0.168575    0.168575

*PARAXIAL TRACE (WAVELENGTH 3)
 SRF      PY          PU          PI          PYC         PUC         PIC
  3     0.069839   -0.297529   -0.297529    8.849979    0.168654    0.168654
```

Subtracting the axial ray height in color 3 from the axial ray height in color 2, we find that the axial color is PAC = –0.157135 – 0.069839 = –0.226974. Repeating the calculation for the chief ray shows PLC = 8.842187 – 8.849979 = –0.007792.

Although the direct tracing of paraxial rays through lenses provides a straightforward procedure for calculating chromatic aberration, it is often more useful to consider the individual effects of the various surfaces of a lens on the total axial or lateral chromatic aberration. The total chromatic aberration of a lens can be expressed as a sum of surface contributions. In a system with k surfaces, the transverse aberration of the axial and chief rays can be written as

$$h'_y = mh_y \quad \text{and} \quad h'_x = mh_x \tag{5.22}$$

The surface contributions are given in terms of the paraxial ray data (c.f. Eqs. 1.19) and the dispersions $dn = n_F - n_C = (n-1)/V$ on either side of the surface by

$$C_{aj} = n_j y_{aj} i_{aj} \left(\frac{dn_j}{n_j} - \frac{dn'_j}{n'_j} \right) \tag{5.23}$$

$$C_{bj} = \frac{i_{bj}}{i_{aj}} C_{aj}$$

In OSLO, the surface contributions for the above lens are displayed by the **chr** command:

```
*CHROMATIC ABERRATIONS
 SRF      PAC         SAC         PLC         SLC
  1    -0.078779   -0.054544   -0.046980   -0.032527
  2    -0.148222   -0.102624    0.039172    0.027122

 SUM   -0.227001   -0.157168   -0.007807   -0.005406
```

Evaluating the image of an optical system can be carried out using either direct ray tracing or aberration theory. Ray tracing gives exact results for the rays traced. Aberration theory gives approximate results that are valid over the entire field and aperture of the system.

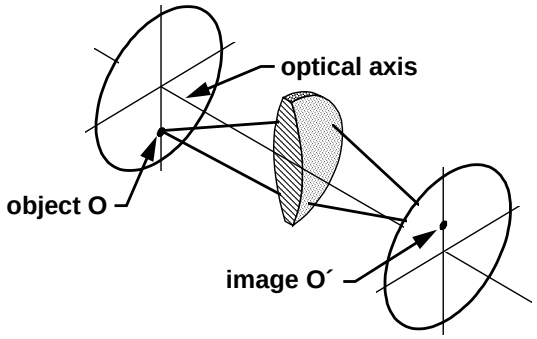
Aberration theory considers defects in optical imagery to be described by a polynomial expansion in which the ray displacement from an ideal image point, or the wavefront displacement from an ideal reference sphere, is written in terms of object and pupil coordinates. The expansion coefficients can be computed using only paraxial ray data. The accuracy of predictions made on the basis of aberration theory depends on the order of the expansion polynomial and the particular system under study. For monochromatic light, OSLO computes both third-order (Seidel), and fifth-order (Buchdahl) aberrations. A single term, the spherical aberration, is computed to seventh order.

Aberration theory uses the symmetry of an optical system to reduce the number of terms that must be included in the aberration polynomial. In this chapter, we explore this aspect of image evaluation in some detail.1 Many of the results are also applicable to ray trace analysis.

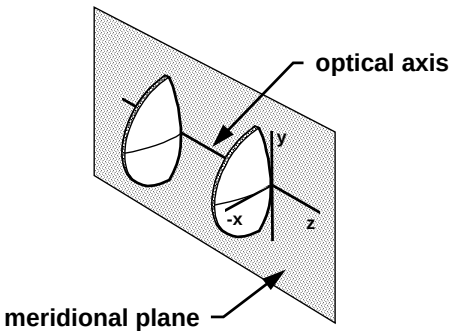
Symmetry properties of centered systems

A centered optical system consists of rotationally symmetrical surfaces whose symmetry axes coincide and thereby define an *optical axis.* Such a system is not only invariant under an arbitrary rotation about the optical axis, but also under a reflection in any plane containing the optical axis.

Consider two planes normal to the optical axis. The ideal state of imagery for a centered system is that whereby the system forms a sharp and undistorted image of an object in one plane, in the other plane. The two planes are then said to be *conjugate*.



Consider any point *O* in the object space of a centered system. If all rays from *O* pass through a unique point *O′* in the image space, imagery of the point *O* is said to be *stigmatic*. The definition of a centered system implies that the following theorem is valid: *If O′ is the stigmatic image of some point O then the two points O and O′ lie in a plane containing the optical axis, called the meridional plane.*



Next suppose that the point *O* is confined to lie in the object plane and that the images of all such points are stigmatic. According to the definition of a centered system, it follows that the stigmatic image of a plane surface *P*, normal to the optical axis, is a surface of revolution about the optical

axis. If the surface is not a plane, we say that the image suffers from *field curvature*, and this is regarded as a defect, or *aberration*, of the image.

Consider the case where the stigmatic image of the object plane is also a plane. The image of some curve in the object plane will be a curve in the image plane, which will in general not be geometrically similar to the original curve. In this case one says the image suffers from *distortion*. If the image is not distorted, then the ratio of some characteristic length in the image to the corresponding length in the object is the magnification *m*, a constant characterizing the image.

If the coordinates of the object point O are (h_y, h_x) and those of the stigmatic image point O' (h_y', h_x'), imagery of O into O' will be free of distortion if

$$h_y' = mh_y \quad \text{and} \quad h_x' = mh_x \tag{5.24}$$

where m is the magnification of the image and is independent of h_y and h_x.

The ideal image of a plane object normal to the optical axis satisfies three conditions: it is stigmatic, free of field curvature, and free of distortion.

An important property of any centered system is that paraxial imagery is ideal in the sense just discussed. The problem of optical design is to manipulate the optical system so that the actual image of a non-paraxial object is identical with the ideal image. In general some arbitrary ray from O will pass through a point O_1' with coordinates (h_y', h_x') in the neighborhood of O'. The extent to which (h_y', h_x') fail to satisfy the above equation is defined to be the *displacement* (ε_y, ε_x) of the ray, where

$$\varepsilon_y = h_y' - mh_y \quad \text{and} \quad \varepsilon_x = h_x' - mh_x \tag{5.25}$$

The displacement depends on the ray and vanishes if imagery is ideal. Actual systems, however, have aberrations. Our discussion here will be devoted primarily to the qualitative features of the displacement (ε_y, ε_x) that are a result of the symmetry properties of a centered system.

The specification of rays

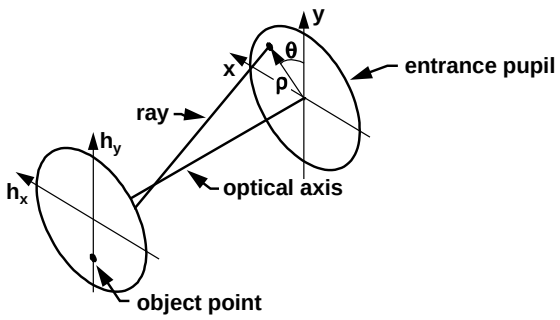
Rays traversing an optical system are conveniently specified by the coordinates of their points of intersection with two surfaces that are not conjugate. Since we are interested in the imagery of objects lying in a plane, the object plane is a natural choice for one of these two surfaces.

We have already seen in Chapter 2, that for perfect imagery by real rays, the Abbe sine condition indicates that the magnification is given by the ratio of the sines of angles in object and image space. A natural choice for the second surface is thus a sphere centered on the object point. This leads to what Hopkins calls *canonical coordinates* for specifying rays, and this is what OSLO uses (by default) for exact ray tracing. We call this *aplanatic* ray aiming.

For the development of aberration theory, which is a generalization of paraxial optics, it is preferable to choose a plane surface, since paraxial rays refract at the tangent planes to spherical surfaces. The normal choice, and the one used here, is the paraxial entrance pupil. We call this *paraxial* ray aiming.

The difference between aplanatic and paraxial ray aiming is only observed at finite object distances, and can usually be neglected when the numerical aperture in object-space is less than about 0.5. Aplanatic ray aiming results from taking direction cosines as the unit of measure, while paraxial ray aiming results from using direction tangents.

If we define coordinates (y, x) in the entrance pupil, then a ray can be specified by four coordinates (h_y, h_x) and (y, x). Since our discussion here is restricted to centered systems, all our results are invariant to a rotation of the system about the optical axis. We use this fact to eliminate h_x. That is, we can always rotate the system to make $h_x = 0.0$. We are left with three numbers (h_y, y, x) that uniquely specify a ray.



Often it is convenient to specify the pupil coordinates by cylindrical coordinates (ρ, θ) where

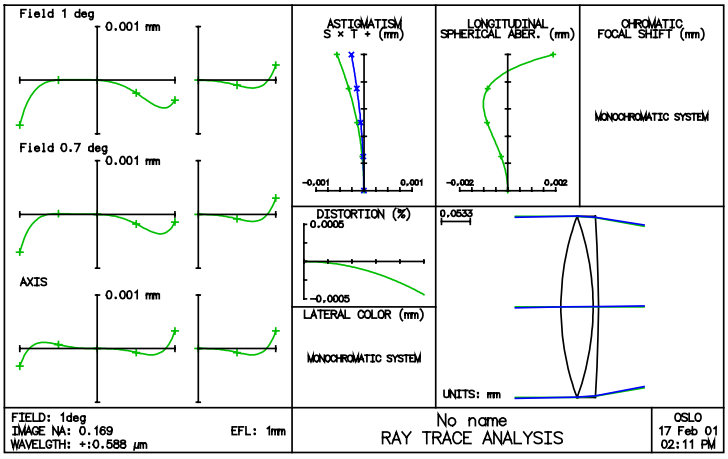
$$y = \rho\cos\theta \quad \text{and} \quad x = \rho\sin\theta \tag{5.26}$$

A ray in the *yz* plane, i.e., with θ = 0 or θ = π, is called a *meridional* ray. A ray with θ = π/2 or θ = 3π/2 is a *sagittal* ray.

Regardless of the coordinate system chosen, the actual values of the coordinates are normalized to unity at the edge of the aperture and field. That is, the value of ρ is between 0.0 and 1.0, as is the value of *h*. In OSLO, the entrance beam radius EBR gives the maximum aperture extent, and the maximum object height is called OBH. Thus to convert fractional aperture or field coordinates to world coordinates, ρ or *h* must be multiplied by EBR or OBH.

Ray-intercept curves

A number of graphical techniques for representing the aberration function are used by lens designers. These are all based on the fact that a necessary condition for ideal imagery is that the image of a point be stigmatic, i.e., the aberration function (ε_y, ε_x) be independent of the ray coordinates (ρ, θ, *h*). The most common representation used in optical design is the *ray-intercept curve*, a plot of ray displacement as a function of aperture coordinates. Although ray-intercept curves are ordinarily generated by ray tracing, they are equally useful for describing aberrations. In this section, we consider the ray- intercept curves for a cemented doublet lens designed to illustrate aberrations, rather than to have best performance. The lens is shown on the lower-right portion of the figure below, and the ray-intercept curves for three field points are shown on the left side.



Consider first the meridional curve corresponding to an object on the optical axis. A ray is said to be meridional if it lies entirely in the meridional (*yz*) plane of the system. If the object lies in the

meridional plane, a ray contained in the meridional plane in the object space of the system will remain in that plane as it traverses through the system.

It should be clear that the meridional ray-intercept curve of a system that forms a perfect geometrical image is just a straight line along the abscissa. Moreover, for a system having a perfect image that is displaced along the optical axis from the nominal image plane, the ray-intercept curve is a straight line that passes through the origin with a non-zero slope.

In view of the symmetry of a centered system, the aberration of a ray from an on-axis object point, with aperture $-\rho$, is opposite in sign but of the same magnitude as the aberration of the ray with aperture ρ. Consequently, the curve obtained by plotting the aberrations of meridional rays from an axial object point is always anti-symmetrical about the origin. In OSLO, standard formatting shows ε_y (called DY) vs. meridional fractional aperture coordinate y (called FY) with $x = 0$, i.e., $\rho\cos\theta$ for $\theta = 0$, and also ε_x (called DX) vs. sagittal fractional aperture coordinate x, (called FX) with $y = 0$, i.e., $\rho\sin\theta$ for $\theta = \pi/2$. For an on-axis point, only half of the meridional curve is independent.

The ray-intercept curve for the on-axis point in the above example shows three aspects of the image. First, the image plane is evidently at the paraxial focus, since the displacements of rays coming from aperture coordinates near the axis are very small. Second, the system evidently has negative (undercorrected) low-order aberration and positive (overcorrected) high-order aberration. This can be understood by observing that the ray displacement initially becomes negative as the aperture coordinate is increased, but turns around at an aperture of about 0.8, and is positive at the edge. Finally, with the exception of rays coming from the edge of the aperture, all rays have displacements less than about 0.2 μm, so we expect the axial image to consist of a core of about this radius, with a flare patch of about 1 μm.

The meridional curves (DY vs. FY) for off-axis points at the 0.7 zone and the edge of the field show increasing aberration and lack of symmetry about the y axis. The fact that the curves are neither symmetrical or anti-symmetrical about the y axis means that the image at these field points has both coma and astigmatism, as discussed in the next section.

Assuming that the object lies in the meridional plane, a sagittal ray is defined to be a ray that intersects the entrance pupil in points for which $\theta = \pm\pi/2$. The symmetry of a centered system does not imply that in the image space these rays will lie in a plane normal to the meridional plane and containing the ideal image. On the contrary, both components of the aberration of a sagittal ray will in general be non-zero, although the component ε_y is usually an order of magnitude less than ε_x (In OSLO, only the DX vs. FX sagittal curve is normally plotted, although there is an option to plot DY vs. FX). However, the symmetry properties of a centered system do imply that if the displacement of the sagittal ray $(0, x)$ is $(\varepsilon_y, \varepsilon_x)$, then the displacement of the sagittal ray $(0, -x)$ must be $(\varepsilon_y, -\varepsilon_x)$, because the image must be symmetrical about the meridional plane. Consequently we need only consider the aberrations of sagittal rays over half the aperture.

The sagittal curves (DX vs. FX) for off-axis points are remarkably similar to the on-axis curves. The reason for this is that the sagittal cross section of the lens seen from the object point does not change so rapidly as the meridional section as the object point is moved off axis.

Comatic and astigmatic aberrations

The on-axis image in a centered system must have complete radial symmetry about the ideal image point. Because of this, it is possible to precisely locate the ideal image even if the actual image is far from stigmatic. In the off-axis case, all one can say about the ideal image is that it must lie in the meridional plane. As a further example, suppose that in addition to having its meridional section as a line of symmetry, the image is symmetrical about a line normal to the meridional plane. Then it must also be symmetrical about the point of intersection O' of these two lines of symmetry. Since an optical system produces only a single ideal image of a given object, the point O' defined above must in fact be the ideal image, assuming for the moment that there is no distortion.

In view of the above discussion, locating the precise position of the ideal image is a simple task if the image is symmetric about a point but will in general be quite difficult, if not impossible. It proves convenient when discussing the aberrations of a point object to decompose the aberration into two distinct parts according to how they transform under a change of sign of the aperture coordinate ρ. Note that a change in the sign of ρ is equivalent to changing the angle θ by π.

We have already seen that because of the symmetry of centered systems, the *x*-displacement of a ray is symmetric about the *y* axis. The *y* displacement of a ray (ρ, θ, h) will be written as the sum of two parts:

$$\varepsilon_y(\rho,\theta,h) = \varepsilon_s(\rho,\theta,h) + \varepsilon_c(\rho,\theta,h) \tag{5.27}$$

where

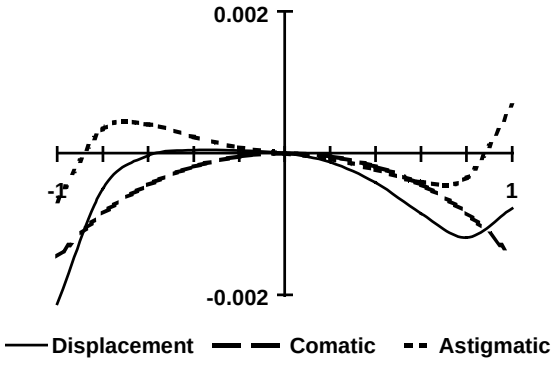
$$\varepsilon_s(\rho,\theta,h) = \tfrac{1}{2}\left[\varepsilon_y(\rho,\theta,h) - \varepsilon_y(-\rho,\theta,h)\right] \tag{5.28}$$
$$\varepsilon_c(\rho,\theta,h) = \tfrac{1}{2}\left[\varepsilon_y(\rho,\theta,h) + \varepsilon_y(-\rho,\theta,h)\right]$$

The two components ε_s and ε_c are respectively known as the *astigmatic* and the *comatic* aberrations of the ray (ρ, θ, h). Their significance and the reasons for decomposing the aberration as in Eq. (5.27) are most readily seen by considering the image for ε_s and ε_c separately.

First note that ε_s changes sign under a change in the sign of ρ. As a result, the image produced for ε_s will be symmetric about the ideal image, i.e., corresponding to the point $(\varepsilon_y, \varepsilon_x)$ in the image there is a point $(-\varepsilon_y, -\varepsilon_x)$ and if the aperture coordinates of the ray producing one of these are (ρ, θ, h), those of the ray producing the second are $(-\rho, \theta, h)$. In view of this high degree of symmetry the aberration ε_s is sometimes called the symmetrical aberration of the ray (ρ, θ, h).

Next consider the comatic aberration ε_c. This is invariant under a change in sign of ρ and hence two rays will pass through each point in the image. As a result the image will in general be asymmetrical about the origin and is accordingly sometimes called the asymmetrical aberration of the ray (ρ, θ, h).

It is interesting to apply the above decomposition to the ray-intercept curves provided in the example on p. 108. The figure below shows the decomposition of the meridional curves for the 0.7 field point into comatic and astigmatic components.



The decomposition of ε_y into the astigmatic and comatic components shows one very important property: since ε_s is of necessity symmetric about the ideal image any asymmetry in the image must come about from the asymmetry of the comatic aberration.

Defocusing

In the introduction to this section, we noted that in order for an image to be ideal it had to satisfy three conditions. The first of these was that it be stigmatic. The second was that there be no

curvature of field. This section is devoted to a detailed discussion of this aberration and the related astigmatism. These two aberrations are intimately associated with the effects of defocusing the image, and accordingly this topic will be considered first.

Consider a plane in image space displaced by an amount ε_z from the paraxial image plane. On this plane, the height of the ray can be written as

$$\tilde{h}'_y = \varepsilon_y + \varepsilon_z u'_a y + \left(m + \varepsilon_z u'_b\right) h_y \qquad (5.29)$$
$$\tilde{h}'_x = \varepsilon_x + \varepsilon_z u'_a x + \left(m + \varepsilon_z u'_b\right) h_x$$

Note that Eqs. (5.29) assume that ε_z is small so that the paraxial approximation for the directions tangents (i.e., u_a' and u_b') is permissible. Inspection of these equations shows that the image in the displaced image plane will be stigmatic provided the quantities

$$\tilde{\varepsilon}_y = \varepsilon_y + \varepsilon_z u'_a y \qquad (5.30)$$
$$\tilde{\varepsilon}_x = \varepsilon_x + \varepsilon_z u'_a x$$

vanish. If this is indeed the case, the image height on the shifted plane is proportional to the object height (h_y, h_x) and the constant of proportionality is

$$\tilde{m} = \left(m + \varepsilon_z u'_b\right) \qquad (5.31)$$

It therefore seems natural to call the quantities given in Eqs. (5.30) and (5.31) the aberration in the displaced image plane and the magnification in the displaced image plane, respectively. From Eq. (5.30) it is seen that the effect of defocusing by the amount ε_z is to increase the aberration by the amount ($\varepsilon_z u_a' y$, $\varepsilon_z u_a' x$) and accordingly this term in Eq. (5.30) is called the *defocusing term*. Certain things should be immediately noticed about the defocusing term. First of all it depends on the aperture coordinates but not on the object height. Second, it is clearly an astigmatic aberration. Together these imply that defocusing can be used to improve an astigmatic aberration, but asymmetry in the image cannot be controlled by defocusing. In fact it is easy to show that if the aberration in the ideal image plane is purely comatic, the images obtained in the displaced images planes situated a distance ε_z on each side of the ideal image plane are identical. In other words, a comatic image is symmetric about the ideal image plane.

There is a simple method for determining the affects of a focal shift on the meridional curve. This is based on Eq. (5.30) and consists simply of drawing a straight line through the origin having slope $\varepsilon_z u_a'$ on the ray-intercept curve. The distance parallel to the ε_y axis from this line to the meridional curve is then the meridional aberration in the displaced image plane.

Curvature of field and astigmatism

Suppose that the image in the ideal image plane is stigmatic. It follows from Eq. (5.30) that in some displaced image plane the aberration is strictly proportional to the aperture (y, x). The image formed in the displaced plane is circular and of uniform intensity, and the radius of the image is proportional to the aperture ρ and the amount of defocus ε_z. One can of course turn these observations around and say that if the image in the ideal image plane has these properties, it is possible to find a displaced image plane in which the image is stigmatic.

To pursue this further, suppose that in the ideal image plane the aberration is given by

$$\varepsilon_y = \kappa y \qquad (5.32)$$
$$\varepsilon_x = \kappa x$$

Where κ is a constant. It follows from Eq. (5.30) that the displacement ε_z required to obtain a stigmatic image is

$$\varepsilon_z = \frac{-\kappa}{u'_a} \qquad (5.33)$$

If the constant of proportionality κ is a function of h, we have a stigmatic image of the plane object formed on a rotationally symmetric surface whose sag ε_z is given by Eq. (5.33). In this case, the image suffers from *curvature of field*.

If the aberration is given by Eq. (5.32), the slopes of the meridional and sagittal ray-intercept curves will be equal. This need not be the case. Consider the situation where the aberration in the ideal image plane is given by

$$\varepsilon_y = \kappa_y y \qquad (5.34)$$
$$\varepsilon_x = \kappa_x x$$

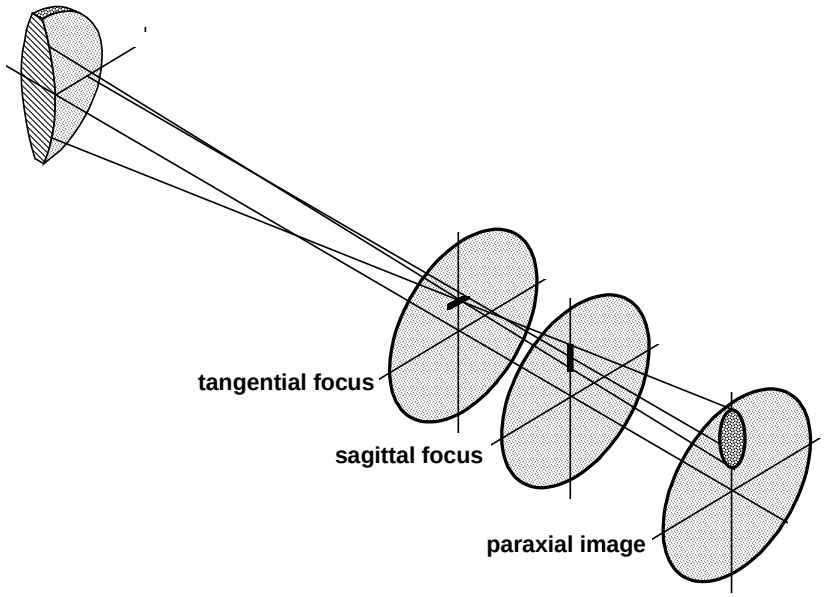
The slope of the meridional curve is κ_y and that of the sagittal curve is κ_x and these slopes may depend on h. We could define two quantities analogous to Eq. (5.33) that would give the displacement required to obtain a stigmatic image for either the meridional rays or the sagittal rays. The fact that the meridional and sagittal field surfaces are distinct means that the image suffers from *astigmatism*. (If $\kappa_x = \kappa_y$, we have pure curvature of field.) The location of the meridional image is sometimes called the tangential focus (ε_{zT}), and the location of the sagittal image is sometimes called the sagittal focus (ε_{zS}).

The astigmatism and curvature of field of a system are usually represented by the so called field curves. The sags ε_{zT}, and ε_{zS} are plotted as a function of h, with h as ordinate. An example of these is given in the upper right portion of the figure on p. 108.

To complete this discussion, something should be said about the appearance of the image in the case when the aberration is due solely to astigmatism and curvature of field. If we temporarily write y for ε_y and x for ε_x, we obtain a parametric equation for the image in a displaced image plane for a particular object height h.

$$y = \left(\kappa_y + \varepsilon_z u'_a\right)\rho\cos\theta \qquad (5.35)$$
$$x = \left(\kappa_x + \varepsilon_z u'_a\right)\rho\sin\theta$$

This pair of equations indicates that the aperture curves corresponding to astigmatism and curvature of field are a set of ellipses centered on the ideal image, with their axes parallel to the coordinate axes. The image patch is thus elliptical and symmetric about the ideal image. As the position of the displaced image plane is varied, the lengths of the axes of the ellipses will vary. In particular, at the tangential focus ($\varepsilon_z = \varepsilon_{zT}$), y vanishes identically; the image is a straight line perpendicular to the meridional plane and passing through the meridional focus. This is the tangential focal line. Similarly, at the sagittal focus ($\varepsilon_z = \varepsilon_{zS}$) x vanishes identically and the image reduces to a straight line in the meridional plane and passing through the sagittal focus. This is the sagittal focal line. Midway between the two foci the image is circular.



Distortion

The one remaining requirement for ideal imagery that has not been discussed in detail is that the image must be geometrically similar to the object. Let it be assumed that the image is stigmatic and is formed in the ideal image plane. However, in order that imagery is ideal, the stigmatic image height must satisfy Eq. 5.24. In general this will not be the case and the ray displacement, although independent of the aperture, will depend on the object height *h*. As a result the stigmatic image of an object will be displaced from the ideal image and the image will not be geometrically similar to the object. The distortion of the image is defined to be the aberration of the ray for which ρ is zero; that is, distortion is the aberration of the chief ray. Distortion is often expressed as a percentage of the ideal image height and plotted as a function of *h*. This has been done for the doublet on p. 108 in the upper right portion of the figure. For this system, the distortion is extremely small, since the field angle is small, and the stop is in contact with the lens. Because of the resulting image of a square grid, positive distortion is called *pincushion* distortion, while negative distortion is referred to as *barrel* distortion.

Aberration polynomials

The detailed computation of the magnitude of aberrations involves a substantial amount of tedious algebra, but the general form of the aberration polynomial for a centered system can be derived from symmetry conditions alone. In a system with rotational symmetry, the optical path along a ray must not change when the system is rotated about the optical axis. We describe a ray by its pupil coordinates (y, x) and object coordinates (h_y, h_x). At this point we do not restrict the object point to lie in the *yz* plane. Because of the rotational symmetry, any polynomial approximation to the optical path must be a function of combinations of y, x, h_y, and h_x that are rotation invariant. These combinations are

$$x^2 + y^2 \qquad xh_x + yh_y \qquad h_x^2 + h_y^2 \tag{5.36}$$

We now use the rotational symmetry of the system to note, just as on p. 107, that we only need to consider object points on the h_y axis. Thus, we set $h_x = 0$, and define the rotation invariant variables as

$$\xi = \rho^2 = x^2 + y^2 \qquad (5.37)$$
$$\eta = \rho h \cos\theta = y h_y$$
$$\zeta = h^2 = h_y^2$$

Aberrations are usually designated according to ray displacements, which are one order less than the wavefront retardations. Thus a fourth-order wavefront term corresponds to a third-order aberration, etc.

The most general expansion of the optical path will involve all combinations of the above terms, taken one at a time, two at a time, three at a time, etc. We write the overall wavefront as

$$W = W_2 + W_4 + W_6 + \cdots \qquad (5.38)$$

The terms taken one at a time give rise to paraxial imagery. We have

$$W_2 = a_1\rho^2 + a_2\rho h\cos\theta + a_3 h^2 \qquad (5.39)$$

The first term represents a curvature of the wavefront, which accounts for focusing. The second represents a tilt proportional to the object height, which accounts for magnification. The third is a constant called piston error (a term that is independent of pupil coordinate), which has no effect on imagery.

The lowest order aberrations involve the rotational invariants taken two at a time, i.e.

$$\xi^2 \quad \eta^2 \quad \zeta^2 \quad \xi\eta \quad \xi\zeta \quad \eta\zeta \qquad (5.40)$$

Expressing these in cylindrical coordinates, we obtain the third-order contribution to the wavefront (omitting the piston error term in ζ^2).

$$W_4 = b_1\rho^4 + b_2\rho^3 h\cos\theta + b_3\rho^2 h^2\cos^2\theta + b_4\rho^2 h^2 + b_5\rho h^3\cos\theta \qquad (5.41)$$

It is convenient to express the third-order wavefront aberration polynomial, Eq. (5.41), in terms of the five *Seidel sums*, $S_\mathrm{I} - S_\mathrm{V}$.

$$W_4 = \frac{1}{8}S_\mathrm{I}\rho^4 + \frac{1}{2}S_\mathrm{II}\rho^3 h\cos\theta + \frac{1}{2}S_\mathrm{III}\rho^2 h^2\cos^2\theta + \frac{1}{4}\left(S_\mathrm{III} + S_\mathrm{IV}\right)\rho^2 h^2 + \frac{1}{2}S_\mathrm{V}\rho h^3\cos\theta \qquad (5.42)$$

The ray displacements are proportional to the derivatives of the optical path. Thus if W is the optical path, the ray displacements are

$$\varepsilon_y \propto \frac{\partial W}{\partial y} \qquad \varepsilon_x \propto \frac{\partial W}{\partial x} \qquad (5.43)$$

If we differentiate Eq. (5.42), we find that the lowest order transverse ray aberrations have the form

$$\varepsilon_{3y} = \sigma_1\rho^3\cos\theta + \sigma_2\left(2+\cos 2\theta\right)\rho^2 h + \left(3\sigma_3 + \sigma_4\right)\rho h^2\cos\theta + \sigma_5 h^3 \qquad (5.44)$$
$$\varepsilon_{3x} = \sigma_1\rho^3\sin\theta + \sigma_2\rho^2 h\sin 2\theta + \left(\sigma_3 + \sigma_4\right)\rho h^2\sin\theta$$

The are called the third-order aberration polynomials, because the sum of the powers of ρ and h all add to 3. If we take the rotational invariants (in W) three at a time and perform a similar differentiation, we obtain the fifth-order aberration polynomials

$$\begin{aligned}\varepsilon_{5y} &= \mu_1\rho^5\cos\theta + (\mu_2 + \mu_3\cos 2\theta)\rho^4 h + (\mu_4 + \mu_6\cos^2\theta)\rho^3 h^2\cos\theta \\ &\quad + (\mu_7 + \mu_8\cos 2\theta)\rho^2 h^3 + \mu_{10}\rho h^4\cos\theta + \mu_{12}h^5 \\ \varepsilon_{5x} &= \mu_1\rho^5\sin\theta + \mu_3\rho^4 h\sin 2\theta + (\mu_5 + \mu_6\cos^2\theta)\rho^3 h^2\sin\theta \\ &\quad + \mu_9\rho^2 h^3\sin 2\theta + \mu_{11}\rho h^4\sin\theta\end{aligned} \tag{5.45}$$

One can in principle extend the same procedure to find the seventh, ninth, etc., order aberration polynomials. The σ and μ terms in the above equations are called *aberration coefficients*. Although it might appear that there are 12 fifth-order aberrations, in fact only 9 are independent. In a system where the third-order aberrations are corrected, the following identities exist:

$$\begin{aligned}\mu_2 &= \frac{3}{2}\mu_3 \\ \mu_4 &= \mu_5 + \mu_6 \\ \mu_7 &= \mu_8 + \mu_9\end{aligned} \tag{5.46}$$

The table below shows various representations of the third-order aberrations (n.b. H = Lagrange invariant, n' = refractive index in image space, u_a' = axial ray angle in image space, u_b' = chief ray angle in image space, $\overline{S}_{\mathrm{I}}$ = first Seidel sum evaluated for chief ray).

Name	Wavefront	Transverse	Longitudinal
Spherical	$S_{\mathrm{I}}/8$	$S_{\mathrm{I}}/(2n'u_a')$	$-S_{\mathrm{I}}/(2n'u_a'^2)$
Coma	$S_{\mathrm{II}}/2$		
Sagittal Coma		$S_{\mathrm{II}}/(2n'u_a')$	
Tangential Coma		$3S_{\mathrm{II}}/(2n'u_a')$	
Astigmatism	$3S_{\mathrm{III}}/4$		
T–*S* distance			$S_{\mathrm{III}}/(n'u_a'^2)$
Petzval	$S_{\mathrm{IV}}/4$		
Image Plane to Petzval Surface			$-S_{\mathrm{IV}}/(2n'u_a'^2)$
Curvature of Petzval Surface			$-n'S_{\mathrm{IV}}/H^2$
Sagittal Field Curvature		$(S_{\mathrm{III}} + S_{\mathrm{IV}})/(2n'u_a')$	$-(S_{\mathrm{III}} + S_{\mathrm{IV}})/(2n'u_a'^2)$
Tangential Field Curvature		$(3S_{\mathrm{III}} + S_{\mathrm{IV}})/(2n'u_a')$	$-(3S_{\mathrm{III}} + S_{\mathrm{IV}})/(2n'u_a'^2)$
Distortion	$S_{\mathrm{V}}/2$	$S_{\mathrm{V}}/(2n'u_a')$	
Fractional Distortion		$S_{\mathrm{V}}/(2H)$	
Entrance Pupil Spherical		$\overline{S}_{\mathrm{I}}/(2n'u_b')$	$-\overline{S}_{\mathrm{I}}/(2n'u_b'^2)$

Aberration types

Aberration types (transverse ray aberrations) are characterized by their dependence upon ρ and *h*. Those aberrations that depend on an odd power of ρ are astigmatic, and those that depend on an even power of ρ are comatic. The aberration depending on ρ and *h* in the combination $\rho^{n-s}h^s$ is said to be of the type n^{th} order, s^{th} degree coma if $(n - s)$ is even, of n^{th} order, $(n - s)^{th}$ degree astigmatism if $(n - s)$ is odd. For example, the aberration in ε_5 that depends on ρ and *h* in the combination ρ^3h^2 is fifth order, cubic astigmatism.

An inspection of the aberration polynomials shows that two new aberration types are introduced in each order. One of these is astigmatic, the other comatic. There are four distinct aberration types represented by the third order aberrations, six in the fifth order, eight in the seventh order and so on. Some of the aberration types, especially those represented in the third and fifth orders, have simple names. These are:

Simple Name	Aberration Type	*Third Order*	*Fifth Order*
Spherical Aberration	n^{th} order, n^{th} degree astigmatism	σ_1	μ_1
Coma or Linear Coma	n^{th} order, first degree or linear coma	σ_2	μ_2, μ_3
Astigmatism	n^{th} order, first degree or linear astigmatism	σ_3, σ_4	μ_{10}, μ_{11}
Distortion	n^{th} order, n^{th} degree coma	σ_5	μ_{12}
Oblique Spherical Aberration	n^{th} order, $(n - 2)^{th}$ degree astigmatism		μ_4, μ_5, μ_6
Elliptical Coma	n^{th} order, $(n - 2)^{th}$ degree coma		μ_7, μ_8, μ_9

Each of the above six aberration types is governed by a number of coefficients, the number varying from type to type. In the next few sections, we consider the relationship between the aberration types and coefficients on the one hand, and the form of the image patch on the other hand. Only the six types present in the third and fifth orders will be considered in detail. The somewhat artificial assumption will be made that only a single aberration type is present.

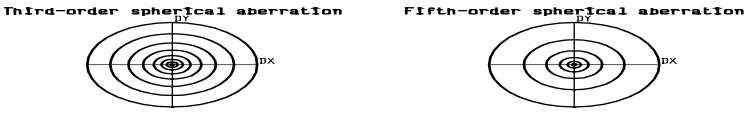
To represent aberrations, we use both ray-intercept curves, as described previously, and annular aperture curves, which are plots of ε_y vs. ε_x, at a fixed field point and aperture radius, with the angle θ being varied through a complete circle.

Spherical aberration

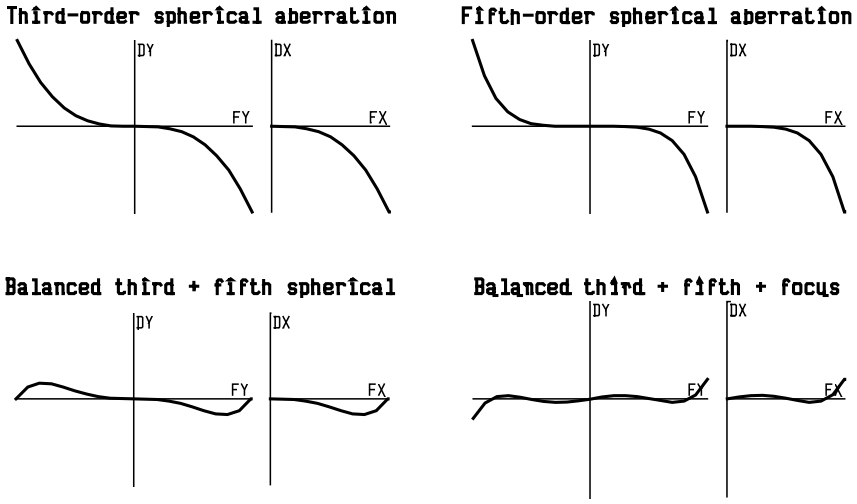
The simplest aberration to study is spherical aberration, the aberration present in the image of an object on the optical axis. It is evident from the symmetry of the system that the image of an axial object is radially symmetric about the optical axis, and is given by

$$\varepsilon = \sigma_1\rho^3 + \mu_1\rho^5 + \tau_1\rho^7 + \cdots \qquad (5.47)$$

The definition of an astigmatic aberration implies that spherical aberration is astigmatic, in fact n^{th} order spherical aberration is n^{th} order, n^{th} degree astigmatism. The aperture curves for spherical aberration of any order are circles centered on the ideal image, as shown below.



The meridional ray-intercept curve for third-order spherical aberration is simply a cubic, flat near the origin and increasing rapidly for large positive and negative apertures. The only significant difference between this case and that of the higher orders is that the higher the order, the flatter is the meridional curve. In other words, the high order aberrations do not become effective until a sufficiently high aperture is reached. The ray-intercept curves below illustrate this effect. The two curves at the bottom show balancing equal amounts of third and fifth-order spherical, and also the balance achieved by also adding a focus shift, which causes a tilting of the ray-intercept curve at the origin.



Linear coma

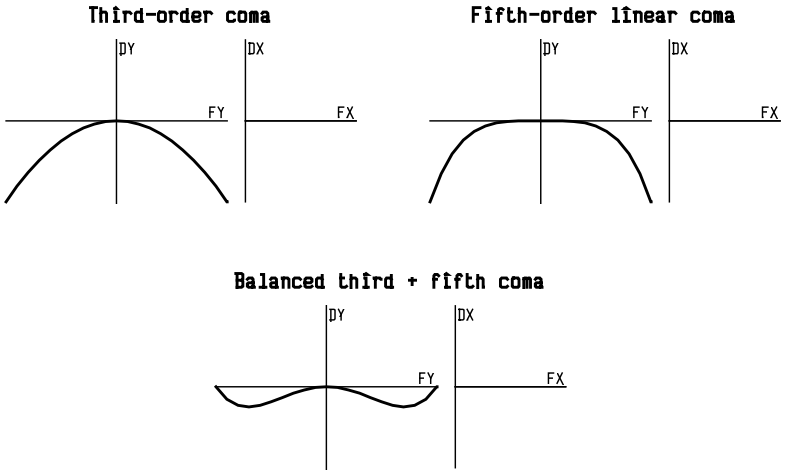
Of the aberration types that depend on the object height, perhaps the most important are those that depend linearly on *h*, i.e., linear coma. Linear coma is the simplest comatic aberration and is the classical aberration from which the term “comatic” is derived. Linear coma is given to the fifth order by

$$\varepsilon_y = \sigma_2\left(2+\cos 2\theta\right)\rho^2 h + \left(\mu_2 + \mu_3 \cos 2\theta\right)\rho^4 h \qquad (5.48)$$

$$\varepsilon_x = \sigma_2 \rho^2 h \sin 2\theta + \mu_3 \rho^4 h \sin 2\theta$$

Insight into the nature of linear coma can be gained by assuming that all aberration coefficients except those of a single order of coma vanish. The annular aperture curves then consist of a family of circles, one for each value of ρ.

Considering third-order coma, if σ_2 is positive, the image patch is directed radially away from the optical axis whereas if σ_2 is negative, it is directed radially toward from the optical axis. These two cases are referred to as positive (overcorrected) and negative (undercorrected) coma, respectively. The image is highly asymmetrical, lying entirely on one side of the paraxial image, and as such the presence of coma in an image is very undesirable. The meridional curve for third order coma is a parabola; the sagittal curve is a straight line along the *x* axis. The aberration curves for higher order coma are similar to those of third order coma, but the meridional curves are much flatter, as shown below.



The pair of meridional rays of aperture ρ intersect the image in the same point (as they must since any comatic aberration is a periodic function of θ of period π). Moreover, the pair of sagittal rays of aperture ρ (i.e., $\theta = \pi/2, 3\pi/2$) also intersect the image in a single point, and this point lies on the meridional section of the image. By taking into account the identities between the coefficients of linear coma, the following important result is established:

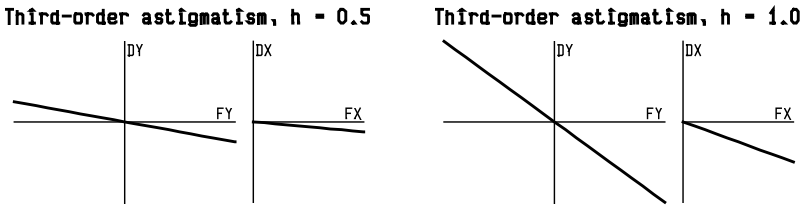
For a given aperture, the maximum displacement of the meridional rays is three times that of the minimum displacement for third order linear coma, and five times for fifth order linear coma.

Linear astigmatism

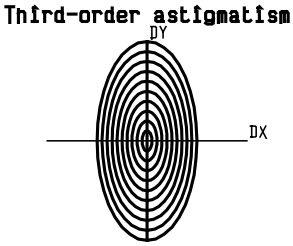
The third terms in the third order aberration polynomial, Eq. (5.44), are linear in the aperture variable ρ. In accordance with our definition, these terms are third-order linear astigmatism. Astigmatism and curvature of field were previously discussed in general terms. From Eqs. (5.44) and (5.45), linear astigmatism is given to fifth order by

$$\varepsilon_y = (3\sigma_3 + \sigma_4)\rho h^2 \cos\theta + \mu_{10}\rho h^4 \cos\theta \qquad (5.49)$$
$$\varepsilon_x = (\sigma_3 + \sigma_4)\rho h^2 \sin\theta + \mu_{11}\rho h^4 \sin\theta$$

The significance of the third order coefficients σ_3 and σ_4 should be considered. Since astigmatism causes the tangential and sagittal images to be sharp, but displaced longitudinally from each other, the ray-intercept curves are straight lines, tilted differently for meridional and sagittal rays. Moreover, the tilt of the lines depends on the field angle, as shown below.



The aperture curves for third-order astigmatism are ellipses on the paraxial image plane, as shown below. At the tangential and sagittal foci, the curves become straight lines, as shown in the figure on p. 113.



The astigmatic focal difference is proportional to σ_3. If this coefficient vanishes, a stigmatic image is formed, for a narrow bundle of rays near the principal ray, on a surface called the *Petzval surface*, whose sag is given by

$$\varepsilon_{zP} = -\frac{\sigma_4}{u'_a} h^2 \tag{5.50}$$

The coefficient σ_4 is proportional to the curvature of this surface and is sometimes called the *coefficient of Petzval curvature*. Comparison of the sags of the third order tangential and sagittal field surfaces with that of the Petzval surface shows that the two field surfaces always lie (in the third order) on the same side of the Petzval surface; to the right if σ_3 is positive and to the left if σ_3 is negative. Moreover it follows from Eqs. (5.49) that the distance, measured parallel to the optical axis, from the Petzval surface to the third order tangential field surface is three times the corresponding distance from the Petzval surface to the third order sagittal field surface.

In Chapter 1 (c.f. Eq. (1.52)), the expression for the Petzval radius (the reciprocal of the Petzval curvature) was shown to be a function only of the curvatures and refractive indices of the lens. Accordingly, for a given system, the Petzval curvature of field is independent of:

the position of the object

the position of the aperture stop

the separations of the surfaces

the bending of any pair of adjacent surfaces (assuming that the two surfaces define a lens in a medium).

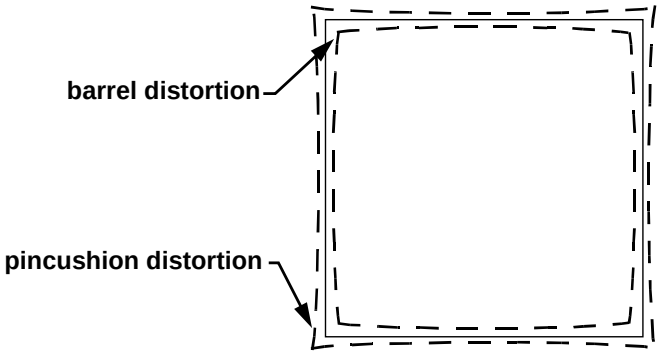
Because of these properties, Petzval curvature must be controlled at a very early stage in the design of a system.

Distortion

Distortion is the aberration type classified as n^{th} order n^{th} degree coma and was discussed previously from the context of ray trace data. Correct to the fifth order, the aberration polynomial for distortion is

$$\varepsilon_y = \sigma_5 h^3 + \mu_{12} h^5 \qquad (5.51)$$
$$\varepsilon_x = 0$$

The coefficients σ and μ appearing in this are the coefficients of third and fifth-order distortion, respectively. If a particular coefficient is positive, the corresponding distortion is said to be *pincushion distortion*; if negative, the corresponding distortion is said to be *barrel distortion*. Distortion could be controlled, in principle, by balancing third-order barrel distortion, say, with fifth-order pincushion distortion, and so on. In practice distortion is often controlled by making the system longitudinally symmetrical about the aperture stop, which reduces all comatic aberrations.



Oblique spherical aberration

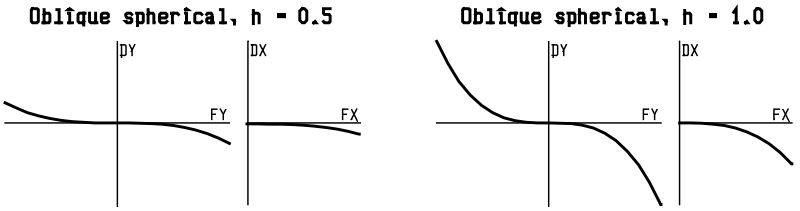
The four previous sections have been devoted to a discussion of the third-order aberrations. As the order is increased, so are the number of distinct aberration types represented in that order. For instance, there are four aberration types in the third order, six in the fifth order, eight in the seventh order, and so on. Each order introduces two new aberration types. One of these is an astigmatic type, the other is comatic. The higher the order, the greater the number of independent coefficients that characterize these new aberrations. Accordingly the aberrations become progressively more and more complicated as the order is increased.

Of the two new types of aberration introduced in the fifth order, the astigmatic type is fifth-order cubic astigmatism, more commonly called oblique spherical aberration, which is given by

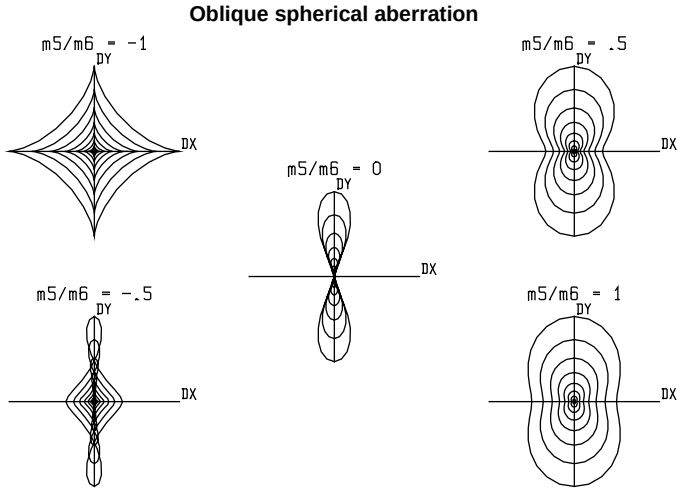
$$\varepsilon_y = \left(\mu_4 + \mu_6 \cos^2\theta\right)\rho^3 h^2 \cos\theta \qquad (5.52)$$
$$\varepsilon_x = \left(\mu_5 + \mu_6 \cos^2\theta\right)\rho^3 h^2 \sin\theta$$

As with the other aberration types that appear in the fifth and higher orders, oblique spherical aberration is much more difficult to control than spherical aberration, linear coma, linear astigmatism or distortion. The coefficient μ_5 governs the oblique spherical aberration of sagittal rays and μ_4 and μ_6 govern the meridional oblique spherical aberration.

That three coefficients appear in Eq. (5.52) is to some extent misleading. In view of the fact that $\mu_4 = \mu_5 + \mu_6$, (assuming that the third-order aberrations are zero; c.f. Eqs. (5.46)), oblique spherical aberration of any order is governed by only two independent coefficients. The ray intercept curves for oblique spherical aberration are similar to those for ordinary spherical aberration, except that the magnitude of the aberration is proportional to the square of the field angle, as shown in the following plots for the case in which $\mu_5 = \mu_6$.



In general, the discussion of the aperture curves for oblique spherical aberration is not so straight forward as for the cases of spherical aberration or linear coma, since two independent coefficients are involved. For example, the figure below shows annular aperture curves plotted for various values of the ratio μ_5/μ_6.



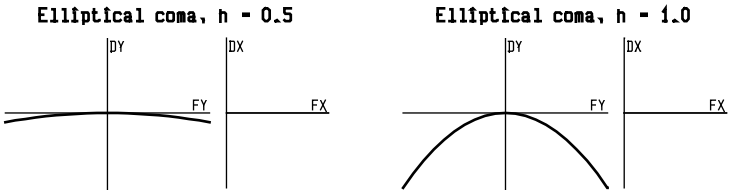
For large values of μ_5/μ_6 we have oval shaped curves. For μ_5/μ_6 between 1 and 0 the sides of the oval are pulled in and actually touch each other along the *y*-axis for $\mu_5/\mu_6 = 0$. For μ_5/μ_6 less than 0 the curves intersect themselves at two points along the *y*-axis, and as μ_5/μ_6 approaches –1 these points of intersection move away from the origin and the outer loops become tighter. In the limit when $\mu_5/\mu_6 = -1$ the curves have cusps at the points where they cross the axes.

Elliptical coma

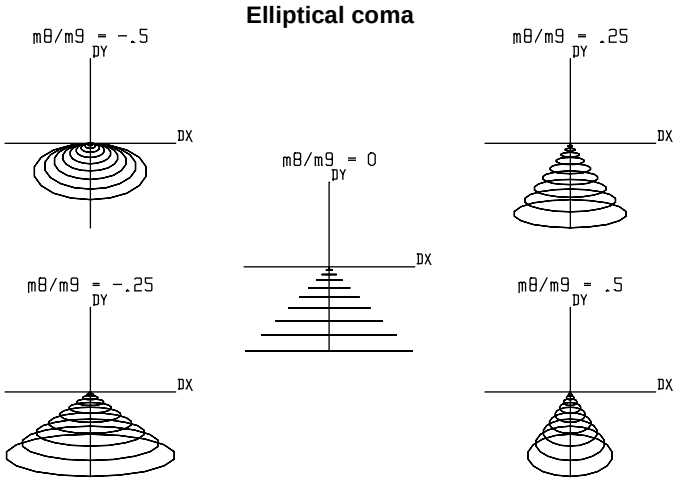
The remaining fifth-order aberration to be discussed is elliptical coma, i.e., the aberration proportional to $\rho^2 h^3$. This aberration is typical of the type classified as n^{th} order, $(n-2)^{\text{th}}$ degree coma, that is, fifth-order cubic coma. Correct to the fifth order, elliptical coma is given by

$$\varepsilon_y = (\mu_7 + \mu_8 \cos 2\theta)\rho^2 h^3 \qquad (5.53)$$
$$\varepsilon_x = \mu_9 \rho^2 h^3 \sin 2\theta$$

The ray-intercept curves for elliptical coma are similar to those for ordinary coma, except that the magnitude of the aberration increases as the cube of the field angle, instead of linearly, as shown below.



Although three coefficients appear in each order in the expression (5.53) for elliptical coma, from Eqs. (5.46) we see that $\mu_7 = \mu_8 + \mu_9$ (assuming zero third-order aberration), so again there are only two independent coefficients. The aperture curves below are plotted for various values of the ratio μ_8/μ_9.



For large negative values of the ratio μ_8/μ_9, the curves are ellipses tangent to the *x*-axis at the point of intersection of the paraxial ray. As shown, the ellipses are below the axis, but a corresponding series exists with the ellipses above the axis when μ_7 is positive. For large positive values of the ratio μ_8/μ_9, the aperture curves become similar to those for ordinary third-order coma. For very small values of the ratio μ_8/μ_9, the eccentricity of the ellipses increases greatly, and the ellipses degenerate into straight lines when $\mu_8 = 0$.

Pupil aberrations

The discussion so far has been concerned with the displacements of rays in the ideal image plane. The various aberrations depend on where rays leave the object and pass through the entrance pupil. There is an entirely analogous set of aberrations that pertain to imagery of the entrance pupil into the exit pupil. The aberrations can be computed by interchanging the roles of the axial and chief rays. In OSLO, they are denoted as PSA3, PCM3, PAS3, and PDS3. The pupil aberration that corresponds to PTZ3 is in fact identical to it, and hence is not separately computed.

In a typical lens system, pupil aberrations tend to be smaller than image aberrations, because the pupils are located in the vicinity of the lens, and the pupil magnification is near unity. However, in lenses such as eyepieces or scanning lenses, which have external pupils, pupil aberrations can play a major role in limiting performance.

Computation of aberration coefficients

In OSLO, the aberration coefficients are computed by commands that are accessed through the Calculate>>Aberration Analysis menu. The third-order aberrations are computed and displayed using the **sei** command. The numerical output depends on the aberration mode setting, which can be unconverted, transverse, or angular. For systems evaluated in focal mode, the default output is transverse, which provides transverse aberrations in the paraxial image plane. For afocal systems, the default output is in angular form. Unconverted output provides the Seidel sum S_i coefficients, which can be converted to various representations using the table on page 5-115.

The table below compares the nomenclature used in OSLO and Smith's *Modern Lens Design* book.

Name	OSLO	Smith
Spherical Aberration	SA3	TSC
Coma	CMA3	CC
Astigmatism	AST3	TAC
Petzval Blur	PTZ3	TPC
Distortion	DIS3	DC

There are two commands used to compute fifth-order aberrations in OSLO. The normal **fif** command computes fifth-order analogs to the third-order aberrations. The command **buc** causes the fifth-order aberrations to be presented in the form used by Buchdahl, with the notation MU1, … , MU12 corresponding to the μ_1, … , μ_{12} in Eq. (5.45). The nomenclature for fifth-order aberrations is shown in the table below. OSLO also computes the seventh-order spherical aberration, which is printed with the fifth-order coefficients using the **fif** command.

The oblique spherical aberration and elliptical coma are not computed explicitly by OSLO, although they can be formed from the Buchdahl coefficients, as shown.

Name	OSLO	Buchdahl
Spherical Aberration	SA5, MU1	μ_1
Coma	CMA5, MU3	μ_3
Astigmatism	AST5, (MU10 – MU11)/4	$(\mu_{10} - \mu_{11})/4$
Petzval Blur	PTZ5, (5MU11 – MU10)/4	$(5\mu_{11} - \mu_{10})/4$
Distortion	DIS5, MU12	μ_{12}
Tangential Oblique Spherical Aberration	MU4 + MU6	$\mu_4 + \mu_6$
Sagittal Oblique Spherical Aberration	MU5	μ_5
Tangential Elliptical Coma	MU7 + MU8	$\mu_7 + \mu_8$
Sagittal Elliptical Coma	MU9	μ_9

Aldis theorem

The Aldis theorem is an example of a “finite aberration formula,” i.e., an expression for the entire aberration of a ray. This can be contrasted to the familiar Seidel (third-order), fifth-order, etc., aberrations, which are truncations of a power series expansion of the aberration. This theorem was derived by H. L. Aldis, but not published until A. Cox’s book “A System of Optical Design”, The Focal Press, London (1964). Perhaps the clearest derivation of the Aldis theorem is given by W. T. Welford in his book “Aberrations of Optical Systems”, Adam Hilger, Ltd., Bristol (1986), Section 9.2. Before giving the actual form of the theorem, we need to define some notation. As usual, quantities measured after refraction or reflection will be indicated by a prime. The Lagrange invariant for the chosen paraxial field and aperture is denoted by *H*. We will assume a system of *k*

centered, spherical surfaces. The refractive index is denoted by *n*, the angle of the paraxial axial ray by *u*, and the angle of incidence of the paraxial axial ray by *i*. Then, the refraction invariant for the paraxial axial ray at surface *j* is denoted by $A_j = n_j i_j = n_j' I_j'$. A real ray intersects surface *j* at the point (x_j, y_j, z_j) with direction cosines (K_j, L_j, M_j). Welford gives the following form for the total transverse ray aberration $(\varepsilon_x, \varepsilon_y)$.

$$\varepsilon_x = \frac{-1}{n_k' u_k' M_k'} \sum_{j=1}^{k} \left[A_j z_j \Delta K_j + \frac{A_j x_j}{M_j' + M_j} \Delta \left(K_j^2 + L_j^2 \right) \right] \tag{5.54}$$

$$\varepsilon_y = \frac{-1}{n_k' u_k' M_k'} \sum_{j=1}^{k} \left[A_j z_j \Delta L_j + \frac{A_j y_j - H}{M_j' + M_j} \Delta \left(K_j^2 + L_j^2 \right) \right] \tag{5.55}$$

In the above expressions, Δ means, as usual, the change in the indicated quantity upon refraction or reflection. Cox and Welford give extensions of the above equations that can be used for aspheric surfaces. Obviously, a real ray must be traced to evaluate the above expressions, and ε_x and ε_y can be computed from the ray data on the image surface (given a reference point). The value of the Aldis theorem is that it gives the contributions to the aberrations on a surface-by-surface basis (as the Seidel aberration coefficients do) so that the source of the aberration can be located.

In OSLO, the SCP command "***aldis** " can be used to compute the surface contributions to the total ray aberration. The computations are valid for centered systems of refracting or reflecting surfaces. The supported surface types are spheres, conics, standard aspheres (**ad**, **ae**, **af**, **ag**) and **asr** surfaces up to tenth order. The syntax of the command is

***aldis** *fby fy fx*

where *fby* is the fractional object height and *fy* and *fx* are the fractional pupil coordinates. This command is also available from the General Analysis pull-right menu of the User main menu. The contribution of each surface to the total aberration of the ray will be displayed. For comparison purposes, the Seidel values of the ray aberration will also be displayed.

Note that the Aldis theorem, like the Seidel aberration coefficients, gives the ray aberration with respect to the *paraxial* image point, which is not necessarily the same as the reference ray intersection with the image surface. Thus, the values of DELTA Y and DELTA X may not be the same as DY and DX computed by the OSLO ray trace if the image surface is not at the paraxial focus. The values labeled "SUM" in the output are the total ray aberration and total Seidel aberration with respect to the paraxial image point. The value labeled "Dyref" is the sum of the paraxial image height for the chosen field plus the SUM value for DELTA Y, minus the intersection height of the reference ray with the image surface. The difference between the SUM value for DELTA Y and Dyref is indicative of how much of the total ray aberration is due to distortion.

Zernike analysis

The power series expansions used in the previous sections are only one way to represent the aberrations of an optical system. Another common approach is the use of *Zernike polynomials* to express the wavefront. The Zernike representation is widely used in interferometry and optical testing. The Zernike polynomials are one of the infinite number of complete sets of polynomials that are orthogonal over the interior of the unit circle. Because of the orthogonality of the polynomials, a Zernike decomposition results in coefficients that are independent of the expansion order used. This is in contrast to a power series expansion, where the coefficients can be strongly influenced by the maximum order of the expansion polynomial that is used. The reader is warned that the terminology, normalization, and notation used for Zernike polynomials is far from uniform. In this discussion, we follow the article by Wyant and Creath.2

As mentioned above, the Zernike polynomials are orthogonal in two real variables, ρ and θ, in a continuous fashion over the interior of the unit circle. The polynomials are not orthogonal over

domains other than the entire unit circle, or, in general, for discretely represented data. The reader is cautioned to be aware of the conditions under which the orthogonality of the Zernike polynomials (and, hence, the uniqueness of a Zernike representation) is assured. As an additional caution, Wyant and Creath note that there situations for which Zernike polynomial representations are not well-suited (e.g., air turbulence, alignment of conical elements). As with all numerical techniques, the user of Zernike polynomials should be aware of the assumptions and limitations governing their use.

The Zernike polynomials have three distinguishing characteristics.

5. The polynomials $Z(\rho, \theta)$ separate into the product of a function $R(\rho)$ of the radial coordinate ρ and a function $G(\theta)$ of the angular coordinate θ.

$$Z(\rho,\theta) = R(\rho)G(\theta) \tag{5.56}$$

The angular function $G(\theta)$ is a continuous, periodic function, with a period of 2π. It also has the property that a rotation of the coordinate system by an angle α does not change the form of the polynomial, i.e., a rotation by angles θ and α must be equivalent to a rotation by $\theta + \alpha$. Hence,

$$G(\theta+\alpha) = G(\theta)G(\alpha) \tag{5.57}$$

The general solution for the function $G(\theta)$ is the complex exponential

$$G(\theta) = e^{\pm im\theta} \tag{5.58}$$

where m is zero or a positive integer.

2. The radial function $R(\rho)$ is a polynomial in ρ of degree n and contains no power of ρ less than m.

3. The radial function $R(\rho)$ is an even function of ρ if m is an even integer and is an odd function of ρ if m is an odd integer.

The radial polynomials are denoted by $R_n^m(\rho)$ and are a special case of the Jacobi polynomials. The orthogonality condition for the radial polynomials is

$$\int_0^1 R_n^m(\rho) R_{n'}^m(\rho)\rho d\rho = \frac{1}{2(n+1)}\delta_{nn'} \tag{5.59}$$

In the above equation, $\delta_{nn'}$ is the Kronecker delta, i.e., $\delta_{nn'} = 1$ if $n = n'$ and $\delta_{nn'} = 0$ if $n \neq n'$. There are several normalizations that are found in the literature. The normalization we shall use is

$$R_n^m(1) = 1 \tag{5.60}$$

The reader is warned that the normalization given by the above equation is common but by no means universal.

When computing the explicit forms of the radial polynomials, it is convenient to factor the polynomial into

$$R_{2n-m}^m(\rho) = Q_n^m(\rho)\rho^m \tag{5.61}$$

where the polynomial $Q_n^m(\rho)$, of order $2(n - m)$, is given by

$$Q_n^m(\rho) = \sum_{s=0}^{n-m}(-1)^s \frac{(2n-m-s)!}{s!(n-s)!(n-m-s)!}\rho^{2(n-m-s)} \tag{5.62}$$

Usually, real polynomials (using sines and cosines) are used in place of the complex exponential. In this case, the Zernike expansion of the wavefront aberration $W(\rho, \theta)$ takes the form

$$W(\rho,\theta) = \bar{W} + \sum_{n=1}^{\infty}\left[A_n Q_n^0(\rho) + \sum_{m=1}^{n} Q_n^m(\rho)\rho^m \left(B_{nm}\cos m\theta + C_{nm}\sin m\theta \right) \right] \quad (5.63)$$

where $\bar{W}$ is the mean value of the wavefront aberration, and A_n, B_{nm}, and C_{nm} are expansion coefficients. Since the "0th" polynomial is the constant (or piston) term 1 and the average value of all of the other Zernike polynomials (over the unit circle) is zero, the average value of W is the coefficient of this 0th term, A_0. With this ordering, the above equation is equivalent to

$$W(\rho,\theta) = A_0 + \sum_{n=1}^{\infty}\left[A_n Q_n^0(\rho) + \sum_{m=1}^{n} Q_n^m(\rho)\rho^m \left(B_{nm}\cos m\theta + C_{nm}\sin m\theta \right) \right] \quad (5.64)$$

For a rotationally symmetric system, the object lies in the meridional plane, so the wavefront aberration is symmetric about the *yz* plane. In this case, only the even functions of θ, i.e., the cosine terms, are non-zero.

Because of the orthogonality of the Zernike polynomials, it is easy to compute the variance of the wavefront aberration, σ^2.

$$\sigma^2 = \sum_{n=1}^{\infty}\left[\frac{A_n^2}{2n+1} + \frac{1}{2}\sum_{m=1}^{n}\frac{B_{nm}^2 + C_{nm}^2}{2n+1-m} \right] \quad (5.65)$$

The rms (root-mean-square) wavefront error, or rms OPD, is just the square root of the variance, i.e., σ. The above expression for the wavefront variance is indicative of another property of Zernike polynomials: each Zernike term minimizes the rms wavefront aberration to the order of that term. The addition of lower order aberrations can only increase the rms wavefront aberration value. In other words, each Zernike term represents an aberration that is optimally balanced, in the sense of the addition of lower order aberrations to minimize the rms value.

The set of 37 Zernike polynomials that is used by OSLO when performing a Zernike analysis of a the Zernike expansion of the wavefront below. The reader is again warned that although this ordering is common, other orderings are also used.

With reference to the table below, Zernike term 0 is seen to be a constant (or piston error) term. Since the average over the unit circle of all of the other Zernike polynomials is zero, the coefficient of term 0 is the average OPD of the wavefront. Terms 1 – 3 represent the paraxial properties of the wavefront (*y* tilt, *x* tilt, and focus respectively). Terms 4 – 8 represent third-order aberrations; terms 9 – 15 represent fifth-order aberrations; terms 16 – 24 represent seventh-order aberrations; terms 25 – 35 represent ninth-order aberrations; term 36 represents eleventh-order spherical aberration.

There is a common misconception: that the low-order Zernike polynomials have a one-to-one correspondence to the common Seidel aberrations. For example, the Zernike coefficient Z8 is compared to the primary spherical aberration SA3. This is misleading, because it is not generally possible to compute Z8 from SA3, nor vice versa. If the aberration function, defined continuously over the unit circle, has no aberrations higher than third order and completely describes the system, then there is such a correspondence. But if fifth or higher-order aberrations are present, then the equality of the third-order relationships is not maintained.

N	*m*	No.	Zernike polynomial
0	0	0	1
1	1	1	$\rho\cos\theta$
		2	$\rho\sin\theta$
	0	3	$2\rho^2-1$
2	2	4	$\rho^2\cos 2\theta$
		5	$\rho^2\sin 2\theta$
	1	6	$\left(3\rho^2-2\right)\rho\cos\theta$
		7	$\left(3\rho^2-2\right)\rho\sin\theta$
	0	8	$6\rho^4-6\rho^2+1$
3	3	9	$\rho^3\cos 3\theta$
		10	$\rho^3\sin 3\theta$
	2	11	$\left(4\rho^2-3\right)\rho^2\cos 2\theta$
		12	$\left(4\rho^2-3\right)\rho^2\sin 2\theta$
	1	13	$\left(10\rho^4-12\rho^2+3\right)\rho\cos\theta$
		14	$\left(10\rho^4-12\rho^2+3\right)\rho\sin\theta$
	0	15	$20\rho^6-30\rho^4+12\rho^2-1$
4	4	16	$\rho^4\cos 4\theta$
		17	$\rho^4\sin 4\theta$
	3	18	$\left(5\rho^2-4\right)\rho^3\cos 3\theta$
		19	$\left(5\rho^2-4\right)\rho^3\sin 3\theta$
	2	20	$\left(15\rho^4-20\rho^2+6\right)\rho^2\cos 2\theta$
		21	$\left(15\rho^4-20\rho^2+6\right)\rho^2\sin 2\theta$
	1	22	$\left(35\rho^6-60\rho^4+30\rho^2-4\right)\rho\cos\theta$
		23	$\left(35\rho^6-60\rho^4+30\rho^2-4\right)\rho\sin\theta$
	0	24	$70\rho^8-140\rho^6+90\rho^4-20\rho^2+1$
5	5	25	$\rho^5\cos 5\theta$
		26	$\rho^5\sin 5\theta$
	4	27	$\left(6\rho^2-5\right)\rho^4\cos 4\theta$
		28	$\left(6\rho^2-5\right)\rho^4\sin 4\theta$
	3	29	$\left(21\rho^4-30\rho^2+10\right)\rho^3\cos 3\theta$
		30	$\left(21\rho^4-30\rho^2+10\right)\rho^3\sin 3\theta$
	2	31	$\left(56\rho^6-105\rho^4+60\rho^2-10\right)\rho^2\cos 2\theta$
		32	$\left(56\rho^6-105\rho^4+60\rho^2-10\right)\rho^2\sin 2\theta$
	1	33	$\left(126\rho^8-280\rho^6+210\rho^4-60\rho^2+5\right)\rho\cos\theta$
		34	$\left(126\rho^8-280\rho^6+210\rho^4-60\rho^2+5\right)\rho\sin\theta$
	0	35	$252\rho^{10}-630\rho^8+560\rho^6-210\rho^4+30\rho^2-1$
6	0	36	$924\rho^{12}-2772\rho^{10}+3150\rho^8-1680\rho^6+420\rho^4-42\rho^2+1$

1 Portions of this section are based on “Analysis of the aberrations of a symmetric system”, notes prepared by P.J. Sands, CSIRO, Australia.
2 J.C. Wyant and K. Creath, “Basic Wavefront Aberration Theory for Optical Metrology,” pp. 2-53, in *Applied Optics and Optical Engineering*, Vol. XI, R.R. Shannon and J.C. Wyant, Eds. (Academic Press, Boston, 1992).

Chapter 6 Ray tracing

Ray tracing is the essence of an optical design program. Although aberration theory provides many insights into the behavior of an optical system, exact ray trace data provide reassurance to lens designers that a design will work as predicted. In fact, this reassurance is sometimes unwarranted, because the data only describe the rays that are traced, not all rays. Nevertheless, practical optical design is almost always carried out using ray tracing, at least in its final stages.

The complexity of ray tracing is twofold. One part involves the development of algorithms that accurately compute the trajectories of rays through optical systems that are increasingly complex. The other involves the specification of the rays to be traced, and the interpretation of the resulting data.

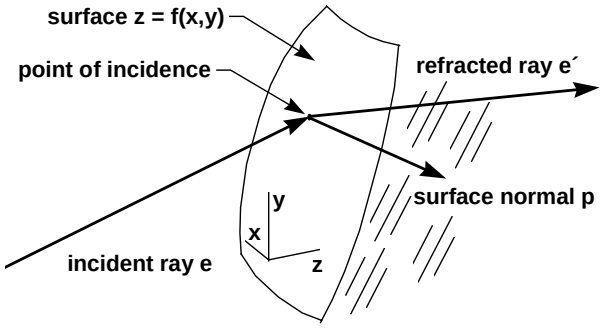
Ray trace algorithms

The basic algorithm for tracing rays involves translating rays from one surface to the next, then refracting them at the surface, starting from the object surface, and ending at the image surface. The translation step involves computing the intersection of a line and a surface, while the refraction step involves applying Snell's law at the point of incidence. If the medium is isotropic and homogenous, rays propagate along straight lines.

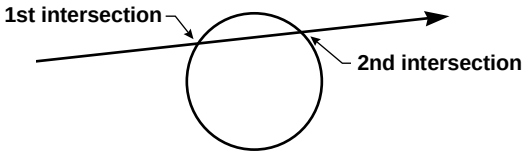
A ray can be described by a vector in the direction of propagation. The magnitude of the ray vector is equal to the refractive index of the medium in which it propagates. At the interface between two refracting (not diffracting) media, the tangential component of the ray vector is continuous, or equivalently, the change in the ray vector is perpendicular to the surface. Let $n\mathbf{e}$ be the ray vector in an initial medium, and $n'\mathbf{e}'$ be the ray vector in a second medium. If $\mathbf{p}$ is a unit vector perpendicular to the surface, Snell's law states that

$$n'\mathbf{e}' \times \mathbf{p} = n\mathbf{e} \times \mathbf{p} \qquad (6.1)$$

Once the point of incidence on the surface defined by $z = F(x, y)$ is established, it is straightforward to differentiate $F(x, y)$ to find the surface normal, whence Eq. (6.1) can be used to determine the emergent ray.



Of the two steps needed to propagate a ray through a surface, translation is usually the more difficult, particularly if the equation of the surface is of some general form $F(x, y, z)$ that cannot be solved analytically. Then an iterative procedure must be followed to find the point of incidence. In many cases, (e.g., a sphere) there may be more than one intersection, and it is not always possible to know which point is correct.



In some systems, it is not possible to enumerate the surfaces according to the order in which rays strike them. Examples include roof prisms, light pipes, and lens arrays. For these, when a ray leaves a surface, all possible surfaces that a ray might hit must be tested to see which one the ray actually intersects. This is called *non-sequential* ray tracing, and is of course a much slower process than ordinary ray tracing.

Ray tracing is complicated by the need to specify the apertures associated with surfaces. OSLO uses the term *aperture* to indicate a region on a surface where some particular action on a ray is defined. For example, rays that strike a surface outside its edge would ordinarily be considered to be *blocked* in sequential ray tracing, or *passed without deviation* in non-sequential ray tracing. In addition, a surface might have one or more interior apertures bounding holes, obstructions, or reflecting spots. These must be accounted for in ray trace algorithms.

There is a question of how to handle apertures during design. For example, although rays that miss the edge of a lens will clearly fail in a finished lens, during the design phase the possibility exists to make the lens aperture bigger to avoid this situation. Thus in optimization, the effects of apertures on rays are not considered.

The amount and type of aperture checking to do is defined by the program and/or the user. However, if a ray fails to intersect a surface, it obviously cannot refract, regardless of the aperture specification. Similarly, if a ray strikes a surface beyond the critical angle, it must be reflected. If a sequential system is set up under the assumption that rays striking the surface are refracted, then rays that undergo total internal reflection (TIR) are effectively blocked.

The above discussion indicates that there are a number of issues related to ray tracing that go beyond simple mathematics. Some of these relate to the types of surfaces involved, others to the way that the system is set up. To make optimum use of OSLO, you should understand the basic rules it uses for tracing rays.

Paraxial ray tracing

Paraxial ray tracing is restricted to systems that have an optical axis. The paraxial equations take no account of surface tilts or decentration, nor surface types such as splines, holograms, gradient index, or diffractive elements. Special coordinate data such as return coordinate data or global coordinates, are not recognized. OSLO is programmed to provide a warning message in most cases when paraxial data is not valid. However, because the data may provide useful information (with the proper interpretation), paraxial ray tracing is not prohibited.

The principal discussion of paraxial ray tracing is given in Chapter 3. OSLO has special commands to compute paraxial-like data for systems that are not recognized by the normal paraxial equations. The ***pxt** command carries out an effective paraxial ray trace by tracing a real ray and differential rays around it, then scaling the resulting data to the given aperture and field. A related command, ***pxc**, computes the paraxial constants.

Real ray tracing

Exact rays, more commonly known as real rays, are ones that obey Snell's law exactly. OSLO uses two types of real rays to evaluate optical systems. The first is called an *ordinary*, or *Lagrangian* ray. This is a ray that starts from a given point on the object surface, in a prescribed direction, that is traced through the system. The second is called a *reference*, or *Hamiltonian* ray. This is a ray that starts from a given point on the object surface, but in a direction that is initially unknown, the direction being determined by the requirement that the ray pass through some prescribed interior point in the system, as mentioned above. A typical example of a reference ray is the real chief ray, which is defined by the requirement that it emanate from the edge of the field and pass through the

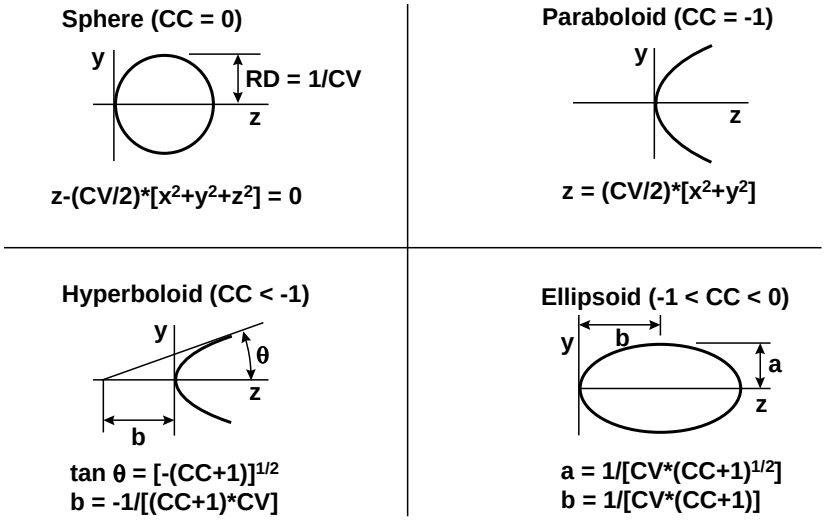
center of the aperture stop. (The names *Lagrangian* and *Hamiltonian* are used in analogy with their use to describe the trajectories of particles in classical mechanics.)

To compute the trajectory of a Hamiltonian ray, a trial direction is assumed, then a Lagrangian ray is traced up to the interior surface of interest. Next, two differential rays are traced (displaced in the *x* and *y* directions) to determine the directional derivatives of the ray with respect to aperture, on the interior surface. A correction in direction is then applied, based on the differential ray data, and a new Lagrangian ray is traced. This process is repeated until the ray passes acceptably close to the prescribed point. Typically, three iterations may be required, which implies that Hamiltonian rays are traced about an order of magnitude more slowly than Lagrangian rays.

Once an exact ray has been traced, it is often possible to trace a ray that is only slightly displaced in aperture or field using simplified equations. Such a ray is called a *differential ray*, and is useful for a variety of purposes. Differential rays are used to compute the tangential and sagittal field sags around a real chief ray, to compute paraxial constants of systems when there is no paraxial ray trace, and to compute derivatives needed to iterate rays that must pass through prescribed interior points in an optical system.

Centered conic surfaces

The vast majority of lenses have spherical surfaces, so the speed and stability of the spherical surface ray trace is of paramount importance. Spherical surfaces are considered a special case of conic surfaces. Since conics can be described by quadratic equations, the ray translation equation can be solved analytically, which significantly speeds up ray tracing.



The various forms of conic surfaces are shown in the above figure. In addition to the forms shown, a hyperboloid with an extremely small curvature and large conic constant can be used to approximate a cone (axicon). As shown, conic surfaces are defined in a coordinate system whose origin is at the point where the symmetry axis intersects the surface. This leads to equations that have a different form from those found in standard mathematics texts. In addition, the so-called conic constant (cc) is used to describe the type of conic, rather than the eccentricity *e* (cc = $-e^2$). In terms of the curvature cv (1/rd) and the rotational invariant $r^2 = x^2 + y^2$, the sag of a conic surface is

$$z = \frac{\mathbf{cv}r^2}{1 + \sqrt{1 - \mathbf{cv}(\mathbf{cc}+1)r^2}} \qquad (6.2)$$

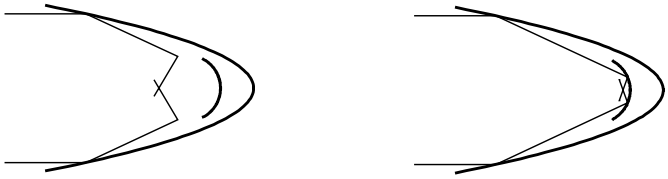
The following code shows a very simple ray trace, written in OSLO's CCL language.

```
#define LENSIZE 4
static double  curv[LENSIZE]  = {0.0,0.01,-0.01,0.0}; // lens data
static double  thik[LENSIZE]  = {0.0,1.0,0.0,0.0};    // lens data
static double  rindx[LENSIZE] = {1.0,1.523,1.0,1.0};  // lens data
static double  n1[LENSIZE], n2[LENSIZE]; // index-related setup items
static double  fx_pupil = 0.5;           // fractional pupil coordinates
static double  fy_pupil = 0.5;           // fractional pupil coordinates
static double  ebr = 8.0;                // entrance beam radius
static double  tho = 1.0e20;             // object distance
static double  oby = -1.0e19;            // object height
static double  kray,lray,mray;           // direction cosines of ray
static double  xray,yray,zray;           // coordinates of ray
static void setup();
static void rayinit();

void raytrex()
{
       int    ray,srf;
       double ray_incr,tl,ts;
       double r1,s1,ncosi,ncosi2,p1,u1,w1;

       setup();
       rayinit();
       for (srf = 1; srf <= 3; srf++)
       {
              zray = zray - thik[srf - 1];
              r1 = mray - curv[srf]*(kray*xray + lray*yray + mray*zray);
              s1 = curv[srf]*(xray*xray + yray*yray + zray*zray) - zray - zray;
              ncosi2 = r1*r1 - n1[srf]*s1;
              ncosi = sqrt(ncosi2);
              u1 = sqrt(n2[srf] + ncosi2) - ncosi;
              w1 = curv[srf]*u1;
              p1 = s1/(r1 + ncosi);
              xray = xray + p1*kray;
              yray = yray + p1*lray;
              zray = zray + p1*mray;
              kray = kray - w1*xray;
              lray = lray - w1*yray;
              mray = mray - w1*zray + u1;
       }
       printf("\nAccuracy test:\n");
       printf("x, y, z = %16.12f %16.12f %16.12f\n",xray,yray,zray);
       printf("k, l, m = %16.12f %16.12f %16.12f\n",kray,lray,mray);
       printf("sumsq   = %16.12f\n\n",kray*kray + lray*lray + mray*mray);
}

static void setup()
{
       int i;

       for (i = 1; i < LENSIZE; i++)
       {
              n1[i] = rindx[i - 1]*rindx[i - 1]*curv[i];
              n2[i] = (rindx[i] + rindx[i - 1])*(rindx[i] - rindx[i - 1]);
       }
}

static void rayinit()
{
       double  b1;

       xray = fx_pupil*ebr;
       yray = fy_pupil*ebr;
       zray = 0.0;
       b1 = (yray - oby)/tho;
       kray = -xray/tho;
       mray = sqrt(1.0/(kray*kray + b1*b1 + 1.0));
       lray = mray*b1;
       kray = -mray*kray;
}
```

It is probably not necessary for you to know the details of algorithms used for ray tracing, but you will find it helpful to understand the basic procedure(1). Although the above code is very simple, it is in fact a working program that traces skew rays through a singlet lens. It consists of an initialization section, a loop, and an output section. The initialization section consists of two parts, one that pre-computes some data related to refractive index, and the other that sets up the data for a ray entering the system. The loop traces the ray from one surface to the next, and includes 19 add/subtract operations, 17 multiplications, and 2 square roots. This is approximately the minimum number of operations needed for real ray tracing. A general-purpose ray trace would need several additional steps for error checking and additional features.

Each surface is defined in a local coordinate system whose origin is at the surface and whose *z*-axis points along the symmetry axis. The sign conventions used to describe conic surfaces are determined by the sag of the surface near the vertex. If the sag is in the positive *z* direction, the curvature is positive, otherwise it is negative.

As mentioned above, it is not always possible to predict the intersection point of a ray with a surface. In the case of a conic surface, an *alternate surface intersection* (ASI) flag can be attached to a surface, instructing the ray trace to choose a different intersection point from the one normally used. The figure below shows a typical application of the ASI flag.



In the figure, rays are reflected by the parabolic mirror, but continue to propagate in the +*z* direction (Normally, rays are expected to propagate in the –*z* direction after a single reflection; rays such as the above are sometimes called *strange* rays). As shown, the ray trace computes the ray intersection with the wrong side of the inner spherical surface. The situation is remedied by turning on the ASI flag, as shown to the right.

Cylinders and toroids

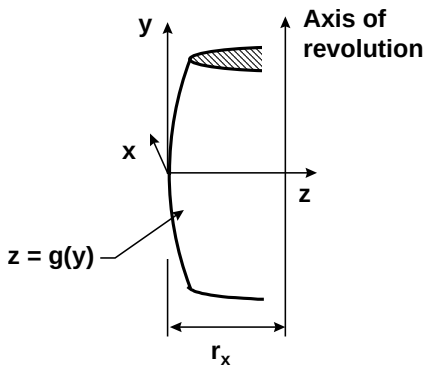
Cylindrical and toroidal surfaces are handled in a special way in OSLO to provide efficient ray tracing. A toroidal lens has different curvatures in the *yz* and *xz* planes. Moreover, a toroidal surface is generated by sweeping one of the curvatures around a point, and the other around a line. A toroidal surface is named according to the direction of the line around which it is generated. The figure below shows a *y* toroid. Because the sections in the two planes are generated differently, a *y* toroid cannot be converted into an *x* toroid by interchanging the curvatures in the two planes. To produce an *x* toroid, a *y* toroid must be rotated 90 degrees about the optical axis. (OSLO has another type of surface, defined according to ISO 10110, which permits a surface to be designated as either a *y* toroid or an *x* toroid without rotation).

In OSLO, the profile in the *yz* plane can be aspheric, of the form

$$z = g(y) = \frac{\mathbf{cv}y^2}{1+\sqrt{1-\mathbf{cv}^2(\mathbf{cc}+1)y^2}} + \mathbf{ad}y^4 + \mathbf{ae}y^6 + \ldots \qquad (6.3)$$

The profile in the *xz* plane is circular, with curvature

$$\mathbf{cvx} = \frac{1}{r_x} \qquad (6.4)$$



A cylindrical lens is one for which **cvx** is zero. For such a lens, the generating axis is in the *x* direction. To define a cylindrical lens with the generating axis in the *y* direction, you must rotate the lens about the optical axis as described above. It is possible, of course, to define a lens where **cv** is zero, and **cvx** is non-zero. However, this surface will be traced as a toroid by OSLO, which means it will use an iterative ray trace that is generally slower than the cylindrical trace.

There are some issues concerning the paraxial optics of systems containing cylindrical lenses that are important to understand. During lens setup, OSLO traces two paraxial rays through the lens to resolve solves and set apertures. For maximum speed, this trace is carried out in the *yz* plane, which means that solves in the *xz* plane are not allowed. In the evaluation routines, however, separate traces are provided in the *yz* and *xz* planes, so the paraxial properties of systems containing cylindrical optics can be readily determined. In order to impose a paraxial constraint (e.g., a solve) in the *xz* plane, you should define a suitable constraint operand in the optimization error function.

Polynomial aspherics

There are a variety of forms of aspheric surface (more complicated than conic sections) used in optical design. Most of these are written as polynomials, and are described in the OSLO Help system. The most common is an aspheric surface of the form

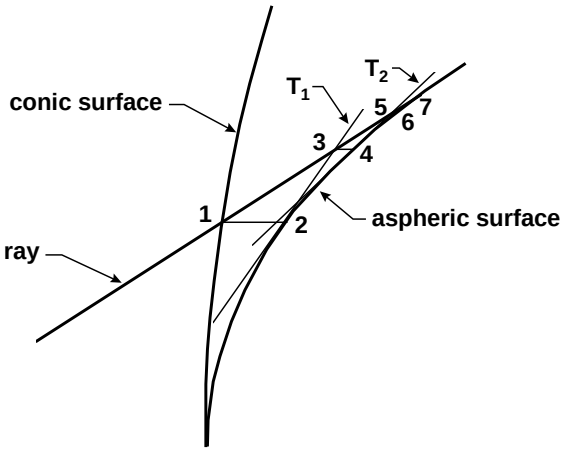
$$z = \frac{\mathbf{cv}r^2}{1+\sqrt{1-\mathbf{cv}(\mathbf{cc}+1)r^2}} + \mathbf{ad}r^4 + \mathbf{ae}r^6 + \mathbf{af}r^8 + \mathbf{ag}r^{10} \qquad (6.5)$$

which OSLO calls a standard asphere.

In a standard asphere the aspheric constants give the sag of the surface in terms of its departure from a conic, not necessarily a sphere. If the conic constant is –1, then

$$z = \frac{1}{2}\mathbf{cv}r^2 + \mathbf{ad}r^4 + \mathbf{ae}r^6 + \mathbf{af}r^8 + \mathbf{ag}r^{10} \qquad (6.6)$$

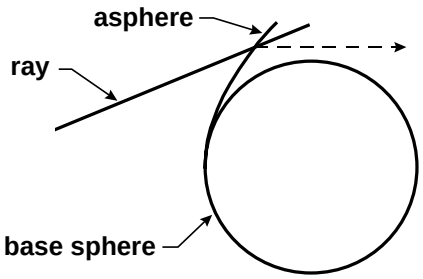
To trace a ray through a polynomial aspheric surface, it is necessary to employ an iterative procedure to find the point of incidence of the ray on the surface. Several schemes have been proposed for such a procedure. One typical procedure uses successive approximations to the intersection of the ray and the tangent plane to the surface. This is used in the sample code provided for the User ray trace. The scheme works as shown in the figure below.



The ray is first translated from the preceding surface to the conic surface tangent to the asphere at the vertex. In the figure, this yields point 1. The sag of the asphere relative to the conic surface is computed, and point 2 is generated as a trial point. The surface normal is computed, which determines the tangent plane T_1. The intersection of the ray with this tangent plane is then computed, which yields point 3. The relative sag is then found to determine point 4, a new tangent

plane T_2 is computed, and so forth. When the intersection points differ by less than some prescribed tolerance, the iteration is terminated.

Typically, the iteration procedure described above converges in 2 or 3 iterations. However, in pathological cases, iteration may be much slower, or even fail to converge at all. For example, consider the case shown below.

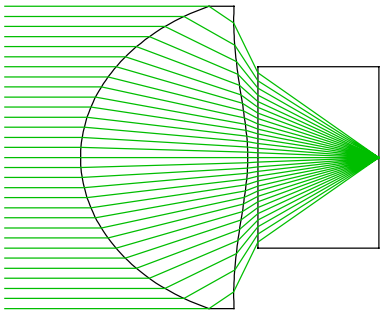


Here, the incident ray never intersects the base sphere, so the iteration fails on the very first step. Various more sophisticated iteration procedures can be used to circumvent this problem.

Spline surfaces

Although the representation of an aspheric surface by a polynomial is convenient when the deviation of the surface from a conic is small, when the aspheric deviation is large, a large number of terms are typically needed to adequately characterize the surface. In such a case, it is often preferable to represent the surface by a spline function. A spline function is a series of cubic polynomials, each valid in a region of the surface, so that there is no single function that represents the entire surface. Often, spline surfaces are represented by a series of point-value pairs, but ray tracing through spline surfaces is simplified if the spline is expressed as a series of point-slope pairs, the scheme used in OSLO. The reason for this is that both the surface sags and curvatures can be computed directly, without solving any equations, if the data are given in this form.

Spline surfaces are useful for describing surfaces that cannot be represented by polynomials that are rotationally symmetric about an optical axis. An example is the CD pickup lens shown below, which works at a numerical aperture of 0.9, and whose surfaces are markedly different from spheres.



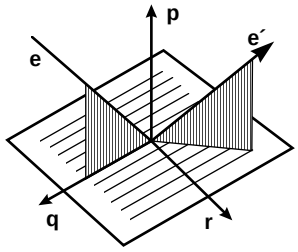
In OSLO, the spline ray trace uses a parabola as the starting point for the translation iteration. The parabola is determined by its vertex curvature, which is used to specify the second derivative of the spline function at the vertex of the surface. In general, setting up spline surfaces that deviate strongly from their base parabolas may involve considerable experimentation by the user to achieve a stable iterative ray trace.

Diffractive surfaces

Many people think of diffraction as physical optics, and optical design as geometrical optics, and conclude that there must be some conflict. In fact, there is none. Ray tracing through diffractive surfaces has been an established procedure for more than fifty years, and has been incorporated in optical design software since about 1960.

In Eq. (6.1), Snell's law was presented in vector form, showing that the tangential component of the ray vector is continuous across a refracting (or reflecting) surface.

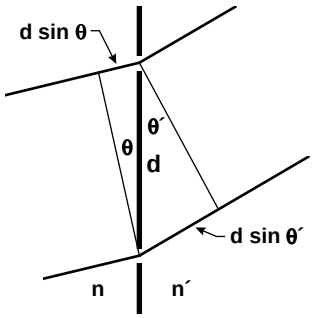
Consider a section of an optical surface small enough to be considered plane, and suppose that this section contains a grating. It doesn't make any difference whether the grating is in amplitude or phase. If the section is small enough, the grating grooves can be considered straight and equally spaced with spacing *d*. Consider a coordinate system on the surface, as shown in the figure below.



The effect of the grating is to add a component in the *r*-direction to the refracted ray vector **e′**. Snell's law, including diffraction, becomes

$$n'\mathbf{e}'\times\mathbf{p} = n\mathbf{e}\times\mathbf{p} + \frac{m\lambda}{d}\mathbf{q} \tag{6.7}$$

If **e** and **e′** lie in the *rp* plane, Eq. (6.7) reduces to the standard diffraction equation, derived with reference to the figure below.



Consider a plane wave incident on a surface separating two media, and suppose that the surface is covered by a double slit. Light incident on each slit will be diffracted into a large angle. Because of the presence of two slits, light propagating in directions where there is constructive interference will be reinforced. These directions are given by

$$n'\sin\theta_d - n\sin\theta_i = \frac{m\lambda_v}{L} \tag{6.8}$$

where *m* is an integer called the order number. When there are more slits, the direction of the diffracted light becomes more precisely defined, the uncertainty in direction being determined by the diffraction angle associated with the size of the region in which the grating spacing can be considered uniform.

The above figure shows that a diffractive surface is a phase surface. Light emerges in directions corresponding to an integer number of wavelengths path difference between adjacent grooves. The shape of the grooves is not important, nor does it matter whether the grooves are amplitude or phase objects; these only affect the *amount* of light that is diffracted into a given order. The amount of diffracted light, or the diffraction efficiency, is discussed in Chapter 10.

Ray tracing serves to determine the direction of propagation of diffracted rays, regardless of whether light is actually propagated in that direction. During the computation, in addition to the square root needed to refract a ray passing from one medium to another, another square root is needed to renormalize the ray vector **e′** as given by Eq. (6.7). Failure of this square root indicates

that the effective grating spacing for the order considered is less than a wavelength, so there is no diffracted ray.

A phase surface can be simulated by a thin prism of artificially high refractive index (~ 10^4), which permits optical software to handle diffractive surfaces without modification; this is called the Sweatt model. Although the OSLO real ray trace is set up to handle diffractive surfaces directly, the Sweatt model is also useful because it enables the computation of aberrations and other items based on paraxial ray data.

There are three major classes of diffractive elements that are available in OSLO. As a result of their method of manufacture, they differ in the way that the grating periods are specified and computed. Although any diffractive surface produces, in general, multiple diffracted waves, OSLO only ray traces one order at a time, so you must specify the diffraction order in which the diffractive optic is to be evaluated. You are free, of course, to change the diffraction order so that the behavior of other, non-design, diffraction orders may be studied.

Linear grating

The linear grating models a classical ruled diffraction grating. The grating lines are parallel to the local *x*-axis, thus the grating spatial frequency only has a *y* component. If the grating substrate is not planar, the grating lines are equally spaced along the chord to the surface, not on the surface itself. In other words, the grating lines are parallel to the local *x*-axis and are equally spaced in *y* on the local $z = 0$ plane.

Optical hologram

The hologram or holographic optical element (HOE) is the recorded interference pattern of two point sources, i.e., two spherical waves. The local grating spacing is determined by the location and orientation of the resultant interference fringes. In OSLO, you specify the locations of the two sources, the wavelength of the source beams, and whether the source beams are real (rays traveling from the point source to the hologram surface) or virtual (rays traveling from the hologram to the point source). The point source locations are used directly in the process of ray tracing through the hologram; the interference pattern is not explicitly computed. See Welford(2) for details on ray tracing through holographic optical elements.

Phase model

The third type of diffractive surface is the general class of computer generated hologram (CGH) type surfaces. These surfaces are specified by the additional phase that is added to a ray when it strikes the diffractive surface. Since each "grating groove" means that another 2π of phase is added to the ray, the phase function $\Phi(x, y)$ at a point (x, y) is 2π times the total number (i.e., the integral) of grating grooves crossed, or equivalently, the effective grating spatial frequencies $f_x = 1/L_x$ and $f_y = 1/L_y$ in x and y are

$$f_x x, y = \frac{1}{2\pi}\frac{\partial \Phi x, y}{\partial x}$$
$$f_y x, y = \frac{1}{2\pi}\frac{\partial \Phi x, y}{\partial y} \qquad (6.9)$$

Ray tracing through a diffractive surface is performed by using Eq. (6.9) to find the effective grating spacing at the intersection point of the ray with the surface and then applying the grating equation to find the direction of the diffracted ray. The optical path length (or phase) increment along the ray is calculated from the phase function. The different phase models available in OSLO are different series expansions of the phase function $\Phi(x, y)$. Rotationally symmetric and asymmetric power series and axicon-like expansions may be used, or the phase may be expanded as a series of Zernike polynomials.

An alternative to the use of the phase model is the so-called *Sweatt model*. Sweatt(3) and Kleinhans(4) showed that a diffractive lens is mathematically equivalent to a thin refractive lens, in the limit as the refractive index goes to infinity and the curvatures of the two surfaces of the thin

lens converge to the substrate curvature of the diffractive surface. The designer can treat a diffractive lens as just a thin lens with a very high refractive index, e.g., 10,001. To model the chromatic properties of the diffractive lens, the refractive index is proportional to the wavelength, i.e.,

$$n_{Sweatt}(\lambda) = \frac{\lambda}{\lambda_0}\left[n_{Sweatt}(\lambda_0) - 1\right] + 1 \tag{6.10}$$

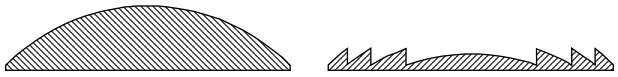
The two curvatures of the equivalent Sweatt model lens of nominal power ϕ_0 and substrate curvature c_s are

$$\begin{aligned} c_1 &= c_s + \frac{\phi_0}{2\left[n_{Sweatt}(\lambda_0) - 1\right]} \\ c_2 &= c_s - \frac{\phi_0}{2\left[n_{Sweatt}(\lambda_0) - 1\right]} \end{aligned} \tag{6.11}$$

Aspheric surface terms are used to model higher order terms in the diffractive surface phase polynomial. OSLO has two SCP commands, ***dfr2swet** and ***swet2dfr**, that can be used to convert a rotationally symmetric diffractive surface to a Sweatt model lens, and vice versa.

Fresnel surfaces

Some Fresnel surfaces bear an outward similarity to diffractive surfaces, but in fact work according to a different principle. Fresnel lenses are constructed to contain several facets that simulate a curved surface by a flat surface, as shown in the figure below.



Plano-convex lens **Fresnel lens**

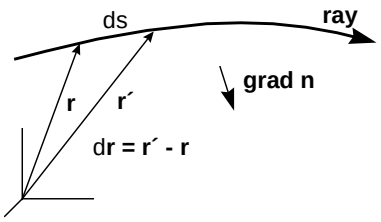
Fresnel lenses have been traditionally used where the design requirement is to minimize weight, such as in searchlights,. More recently, plastic Fresnel lenses have been used for a variety of low-resolution applications, such as the design of condenser lenses for overhead projectors. Unlike diffractive lenses, Fresnel lenses operate by incoherent superposition of light from adjacent facets.

OSLO allows aspheric Fresnel surfaces to be placed on any conic substrate, in addition to a plane substrate. The ray trace translates rays to the base conic substrate, then refracts them as though the surface had a slope at that point equivalent to the aspheric Fresnel surface. Thus no account is taken of the actual facet height, which introduces some error in the refracted ray trajectory. If the facet dimensions are large enough that the error is significant, the model should not be used. Instead, for such a case the lens can be modeled as a non-sequential surface.

Gradient index surfaces

Snell's law, as presented above, gives the change in the direction of rays that cross boundaries separating homogeneous materials. If the materials are inhomogeneous, rays no longer propagate in straight lines. Inhomogeneous, or *gradient-index* materials are finding increased application in optical design, both as lenses in micro-optical systems, and as a substitute for aspherics in conventional imaging systems.

To trace rays through gradient index systems, it is necessary to generalize Snell's law. The figure below shows the propagation of a ray through such a system.



The equation of the ray is

$$\frac{d}{ds}\left[n(\mathbf{r})\frac{d\mathbf{r}}{ds} \right] = \text{grad } n(\mathbf{r}) \qquad (6.12)$$

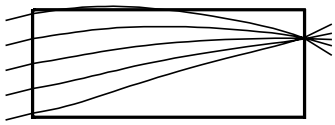
which can be solved by a variety of numerical techniques, the one used in OSLO being the so-called Sharma method, which is based on the Runge-Kutta method for solving differential equations. The numerical solution of eq. (6.12) involves the choice of a step size *ds*, as shown in the figure above. Tracing a ray through a gradient index material proceeds step by step, until the ray intersects the next surface.

The choice of step length determines both the accuracy and speed of gradient-index ray tracing. The default step size in OSLO is set to 0.1 (lens units), which produces acceptable accuracy for typical gradients. It is a good idea to test the step size by decreasing it until you are convinced that the trace will yield acceptable accuracy for the particular system you are working with. In addition, it may be that an acceptable accuracy for ray displacement analysis is not acceptable for OPD analysis.

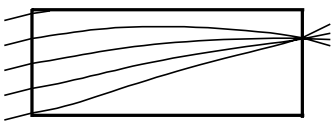
Gradient index materials are specified in a variety of ways. If the gradient is in the direction of the optical axis, it is an *axial gradient*. If the gradient is perpendicular to the axis, it is a *radial gradient*. If the gradient is spherically symmetrical about some point, it is a *spherical gradient*. OSLO contains ray trace routines for handling ten different forms of gradient. In each case, the gradient is given by a power-series expansion, similar to an aspheric surface. If the standard gradient types are not sufficient, it is possible to specify a *user-defined gradient*, in which the quantities $n(\mathbf{r})$ and grad $n(\mathbf{r})$ (c.f. Eq. (6.12)) are specified by a CCL command.

The dispersion of gradient index materials is specified by giving values for the gradient coefficients at each of the defined wavelengths. During optimization, the ν-number and partial dispersion can be varied, but these quantities are subject to heavy restriction during the fabrication process. A special Gradium™ material can be used for axial gradients. Unlike the other gradient materials, Gradium materials have dispersive properties that are specified by the manufacturer and can not be varied during optimization.

Aperture checking for gradient index surfaces is more complicated than for ordinary surfaces, because rays can be blocked between surfaces, as shown in the figure below. A general operating condition (**grck**) controls whether each individual step is aperture checked during transfer through a gradient medium.



Aperture check GRIN segs off



Aperture check GRIN segs on

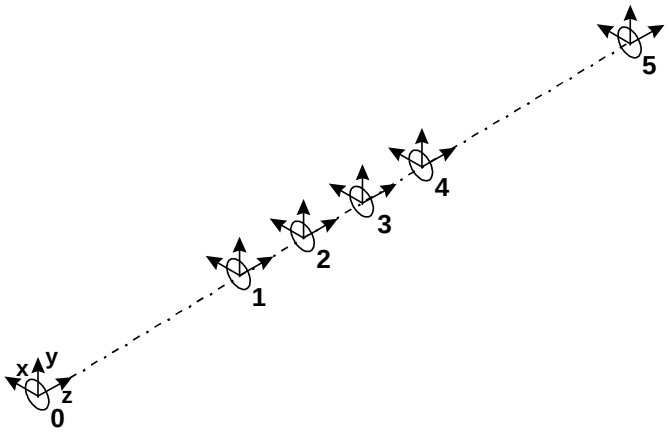
Specification of ray data

For the designer, the most important aspect of ray tracing is to specify the correct rays, and to understand the data that the program computes. This involves a knowledge of the coordinate systems in which systems are described, and the quantities that are used to describe rays in those systems.

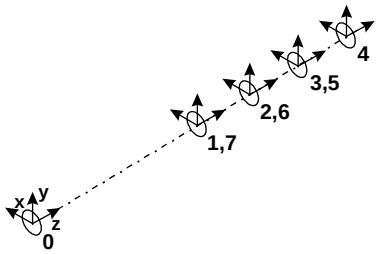
Surface coordinates

OSLO describes optical systems using a surface-based model. Each surface is described in a local coordinate system, whose origin is relative to a base coordinate system for that surface. The base coordinate system, in turn, is located on the *z*-axis of the previous local coordinate system, if local coordinates are being used, or at some point (*x*, *y*, *z*) in a global coordinate system, if global coordinates are being used. Each coordinate system is right-handed and the *z*-axis is the symmetry axis of the surface (if there is one).

The case where there is a centered system with local coordinates is shown in the figure below. For such a system, each local coordinate system is congruent with its base coordinate system. The location of the next vertex is called the thickness of the surface. The surfaces are numbered sequentially, starting with 0 for the object surface. The highest numbered surface is called the image surface, whether or not there is an "image" on that surface.



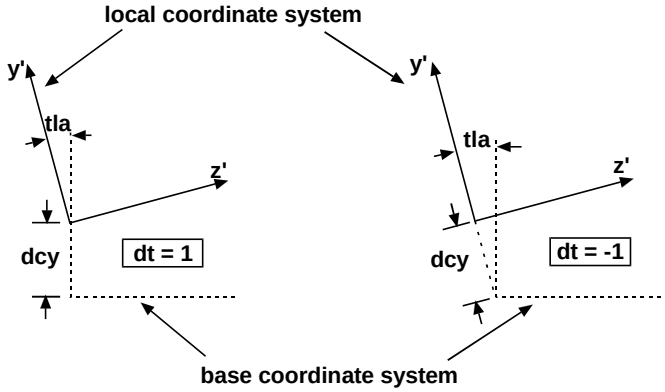
The orientation of the surfaces does not depend on whether there are reflecting surfaces. In the above system, if surface 4 were reflecting, the system would appear as follows (the image surface is not shown). The thicknesses following the reflecting surface are negative because each vertex lies on the negative *z* axis of the previous surface.



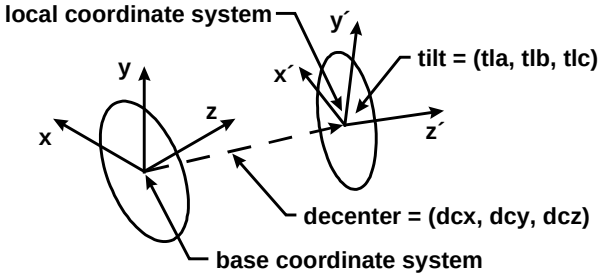
Many optical systems are designed using tilted and decentered surfaces, and all fabricated systems have tilted and decentered surfaces due to manufacturing errors. In local coordinates, the vertex of the base coordinate system normally lies on the *z*-axis of the previous local coordinate system. In global coordinates, the vertex of the current base coordinate system is relative to the base coordinate system of the global reference surface.

Since tilting and de-centering are non-commutative, the program must be told which comes first. OSLO uses a special datum (called **dt**) to accomplish this. If tilting is carried out after de-centering, tilt operations are carried out in the order **tla**, **tlb**, **tlc**. If tilting is carried out before de-

centering, tilt operations are carried out in the order **tlc**, **tlb**, **tla**. This method of specifying tilted and de-centered surfaces allows one to restore the original position and orientation of a surface that has been tilted and de-centered by picking up all the tilt and de-centering data with negative signs.

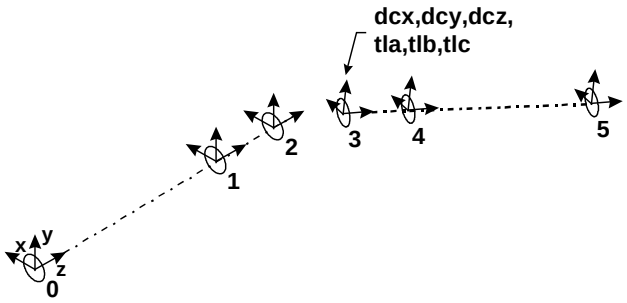


When a surface is tilted and/or decentered, its local coordinate system is separated from its base coordinate system, as shown in the figure below. The origin of the local coordinate system is specified by three coordinates dcx, dcy, and dcz measured from the origin of the base coordinate system. The orientation of the local coordinate system is given by tilt angles tla, tlb, and tlc.

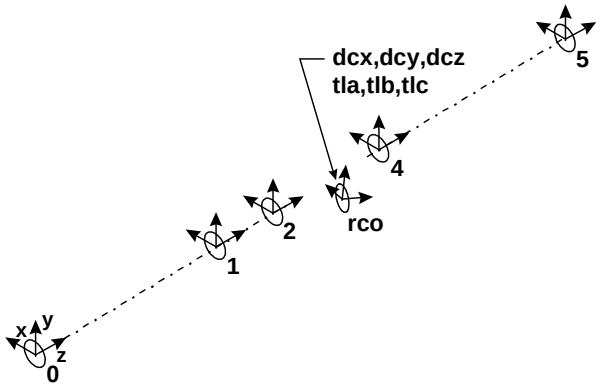


In addition, the surface tilt can be made relative to an offset tilt point specified by three coordinates tox, toy, and toz relative to the nominal location of the local coordinate system vertex. For a surface in local coordinates, if the decenter-tilt order is to decenter, then tilt, the tilt vertex offset (tox, toy, toz) is measured in the coordinate system that results from applying the decentration coordinates to the base coordinate system of the surface. If tilts are applied before decentration, tox, toy, and toz are given in the base coordinate system. For a surface in global coordinates, the coordinate system for tox, toy, and toz is the system of the global reference surface (decentered, if decentration is performed before tilts).

In local coordinates, since the vertex of the coordinate system following a tilted surface lies on the new z-axis, tilting a surface effectively redefines the optical axis for all of the following surfaces. In the following figures, the base coordinate system for each surface is not shown.



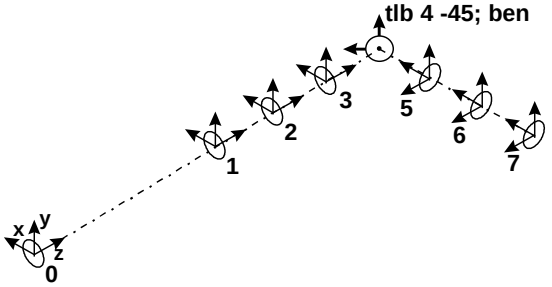
OSLO contains a special return-coordinates command (**rco**) that overrules the above convention. If no argument is given, or the argument is equal to the current surface number, the vertex of the next surface is placed on the *z*-axis of the current *base* coordinate system. If the argument is a previous surface number, the vertex of the next surface is placed on the *z*-axis of the previous *local* coordinate system, unless the surface is specifically designated to always use the base coordinate system for **rco**. In both cases the thickness of the current surface gives the distance along the *z*-axis to the next vertex.



In a normal system, the next surface is expressed in the local coordinate system of the current surface, that is, the thickness is measured along the *z*-axis of the local coordinate system. However, OSLO also has a return_coordinates command (**rco**) that causes the thickness to be measured along the *z*-axis of the base coordinate system of the current surface. An option to the command allows you to measure the thickness along the *z*-axis of any previous local coordinate system. Thus you can, for example, refer the coordinates of any surface to some base surface (e.g. surface 1). This provides a way to specify coordinates globally. Return_coordinates are not the same as global coordinates; you cannot transform back and forth between local and global coordinates when using **rco**.

A convenient way to work with systems that have tilted mirrors is to use the bend (**ben**) command. The bend command automatically sets the optical axis after reflection to be coincident with the ray that connects the previous vertex with the current vertex, i.e. it in effect propagates the optical axis through the system. The new coordinate system is adjusted, if necessary, so that the meridional plane remains the meridional plane after reflection.

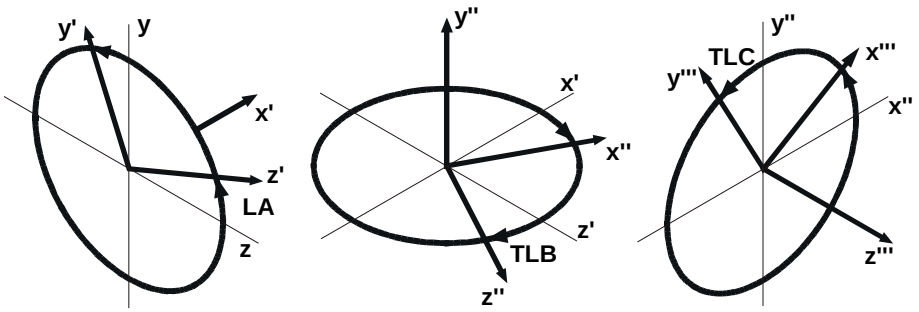
If a tilted surface is reflecting, normally it will be marked with a flag that is set by the OSLO bend (**ben)** command, as shown below. This command rotates the base coordinate system of the current surface (after applying the transformation to the local coordinate system) by twice the tilt, and places the vertex of the next surface on the rotated base coordinate system *z*-axis. This accounts for the physical reflection of the beam at the vertex of the current surface. If necessary, an azimuthal tilt is applied so that the meridional plane of the next surface is its *yz* plane.



Tilt conventions

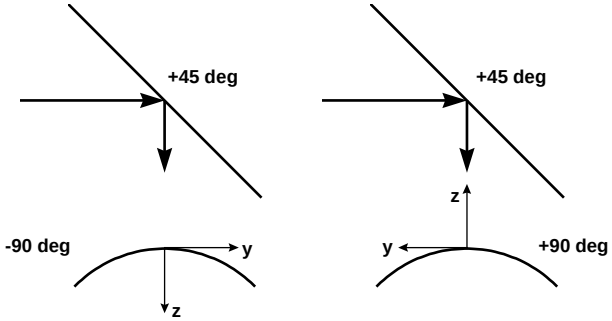
Tilted coordinate systems are handled using an Euler angle system, in which each of the three tilt angles tla, tlb, and tlc takes place in the tilted coordinate system determined by the previous tilts. The tilt angles tla, tlb, and tlc are sometimes called meridional, sagittal, and azimuthal tilts,

respectively. Tilt angles are measured in degrees, according to the figure below. Note that the tla and tlb tilts are left-handed, while the tlc tilt is right-handed. Although not common in other disciplines (e.g. mechanical design), this convention is widely used in optical design software.



Since tilting is a non-commutative operation, it is necessary to undo tilts in the opposite order to that in which they were applied. OSLO provides a pickup command (**pk tdm**) that does this automatically.

A common error is using a tilt angle that is wrong by 180 degrees. The reason for the error is that there are two ways to express the same surface. Suppose you have a system where a beam, initially traveling from left to right, reflects from a mirror that is tilted at 45 degrees. Then the beam is traveling perpendicular to the original beam, as shown in the following figure.



The question then arises about whether the next surface should be tilted by +90 degrees or -90 degrees with respect to the original system, since it doesn't seem to make any difference. If the surface is tilted by -90 degrees, the curvature as shown is positive, whereas if the surface is tilted by +90 degrees, the curvature is negative.

Actually, it does make a difference, and the correct answer is +90 degrees. While the two cases are geometrically identical, for the purpose of ray tracing they are not. A line intersects a sphere in two places, so in solving the ray trace equations, there must be a convention as to which intersection point to choose.

In OSLO, the ray trace equations are set up to use the normal intersection points predicted by the standard sign convention. This convention, when generalized to handle tilted surfaces, states that the *z*-axis of each local coordinate system should point in the same direction as the beam propagation after an even number of reflections, and in the opposite direction after an odd number of reflections. In the present case, the beam has undergone one reflection by the 45 degree mirror, so the *z*-axis of the next local coordinate system should point against the direction of beam propagation, as shown in the figure to the right.

In OSLO, there is a special flag on each surface called the ASI (alternate surface intersection) flag. If you turn it on, the ray trace intersection points will be opposite from the normal, so you could handle a -90 degree tilt by using the ASI flag. But then you would have to deal with the next surface in a different way from normal, so it is simpler to set your system up according to standard conventions.

Global coordinates

As mentioned above, the base coordinate system of a surface is obtained by translating the local coordinate system of the previous surface along its *z*-axis by a distance equal to the thickness assigned to the previous surface. If tilt angles and/or decentered coordinates are specified at the surface, they are applied to reorient and/or reposition the local coordinate system of the surface with respect to this base coordinate system.

In situations involving a number of surfaces that are tilted and decentered with respect to one another, it is sometimes more convenient to specify the position and orientation of a surface with respect to a global coordinate system. The **gc** (global coordinates) command specifies the surface number of a surface within the system whose *local* coordinate system will be used in place of the base system of another surface, when that surface is tilted and/or decentered. Note that gc is a surface data item, so that some surfaces in a system may be specified in global coordinates and some in local coordinates. A global tilt/decenter specification can be deleted with the gcd command.

When global tilt angles and decentered coordinates are used at a surface, the thickness assigned to the surface preceding the surface is not used. The *z* position of the surface is controlled solely by the *z* decentration dcz on the surface.

Whenever the **gc** or **gcd** command is used on a surface, a new set of tilt angles and decentered coordinates is calculated automatically so that the position and orientation of the current surface remains unaltered. Thus, for example, tilt angles and decentered coordinates for one or more surfaces may be set up initially in global coordinates and then changed to a local coordinate representation simply by changing the coordinates specification.

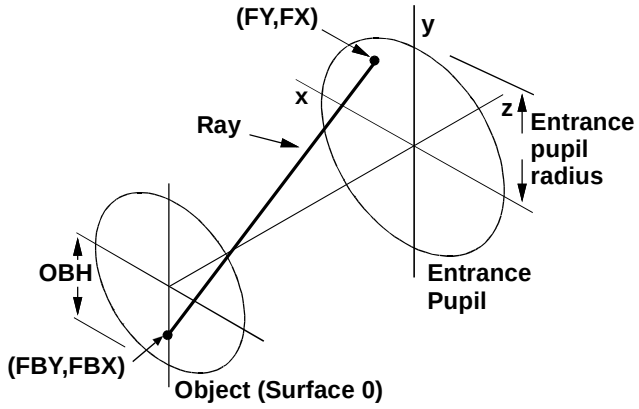
The set of tilt angles defining the orientation of a surface is not unique. For example, a tla of 180 degrees followed by a tlc of 180 degrees is equivalent to a tlb of 180 degrees. When tilt angles are calculated, the following limits are imposed in order to eliminate ambiguity:

$$\begin{aligned} -180^\circ \le \mathbf{tla} \le +180^\circ \\ -90^\circ \le \mathbf{tlb} \le +90^\circ \\ -180^\circ \le \mathbf{tlc} \le +180^\circ \end{aligned} \qquad (6.13)$$

When switching to local coordinates, the *z*-decentered coordinate of the current surface is set to zero and a thickness is calculated and assigned to the preceding surface. When switching from local to global coordinates, the thickness assigned to the preceding surface becomes undefined and an appropriate *z*-decentered coordinate is assigned to the current surface.

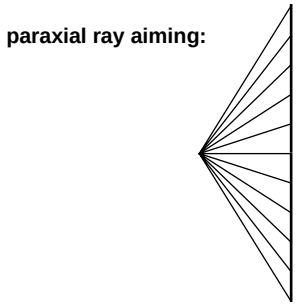
Fractional coordinates

It has become common to specify an input ray by its fractional coordinates on two separated surfaces. The usual scheme uses the ratio of the object height of a ray to the overall object height, and the ratio of the pupil height of the ray to the overall pupil size, as shown in the figure below. When the object is at infinity, the aperture and field of view are modest, and paraxial ray tracing is accurate, this approach is adequate to define fractional coordinates. For more complicated systems, a more elaborate scheme is required.

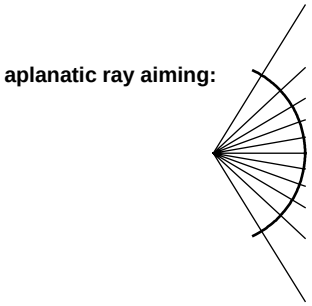


The meaning of fractional coordinates is a matter of definition. Fractional coordinates should be chosen so that they produce reasonable results when applied to practical problems. To specify the field of view, OSLO by default uses the field angle when the object is at infinity, or the object height when the object is at a finite distance. In either case, the fractional object heights FBY and FBX are chosen to be proportional to the tangent of the field angle. This is because a perfect lens is defined to be one that preserves shapes on a plane image surface.

The fractional coordinates FY and FX used to specify aperture are more complicated, at least when dealing with objects at finite distances. OSLO provides two basic options, called *paraxial* ray aiming and *aplanatic* ray aiming. When paraxial ray aiming is used, the fractional coordinate used to specify a ray is proportional to the tangent of the ray angle. This is the convention used in paraxial optics. For small apertures, this is satisfactory, but when the aperture becomes large, as shown in the figure below, rays towards the edge of the aperture are compressed in angular space, just the opposite of what is desirable from a radiometric point of view.



When aplanatic ray aiming is used, fractional coordinates are chosen to be proportional to the direction cosines of ray angles. The is equivalent to aiming the rays at a sphere centered on the object point, as shown below. Now the rays towards the edge of the aperture are spread out, in concurrence with the Abbe sine condition for aplanatic imaging.



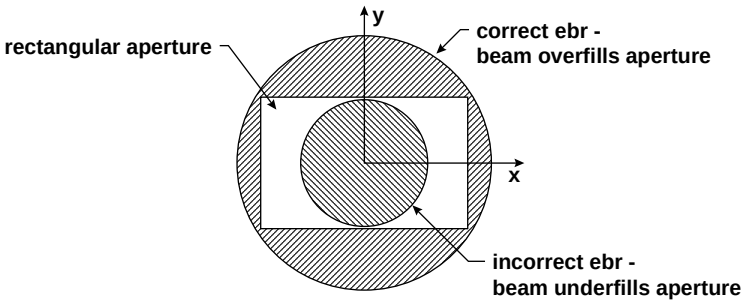
Since most large aperture systems more closely obey the sine condition than the paraxial condition, OSLO uses aplanatic ray aiming by default. The difference is not noticeable at small

apertures, and in fact when the object numerical aperture (**nao**) is less than 0.1, OSLO actually uses paraxial ray aiming.

The definition of fractional coordinates should accommodate motion of the pupil gracefully, both in the longitudinal and transverse directions. OSLO uses **nao** to specify the aperture of a system having an object at finite distance. The aperture can then be considered to be defined by a cone of rays emanating from the origin of the object surface, symmetrical around the *z*-axis. The limits of this cone define the ±1 fractional aperture coordinates for an on-axis beam.

When the object distance becomes infinite, the object numerical aperture drops to zero. Because of this, OSLO uses the *entrance beam radius* (**ebr**), defined as the radius of the beam on the plane tangent to surface 1, to describe the aperture of a system with an object at infinity. When the object is at infinity, the entrance beam radius is equal to the entrance pupil radius.

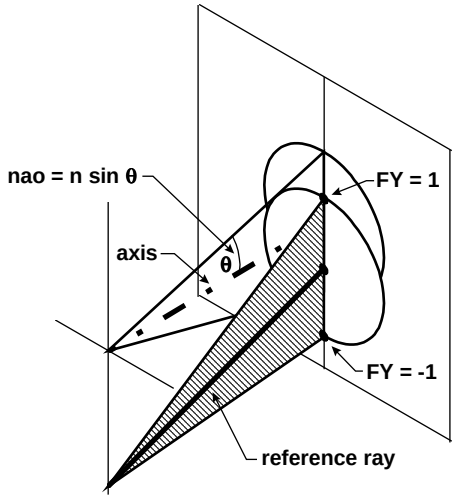
It is important to understand that **ebr** and **nao** define the size of the illuminating beam. No Lagrangian rays are traced outside of the cone defined by these quantities, which corresponds to a fractional aperture of ±1. This is particularly important to remember when dealing with off-axis or odd-shaped apertures. A simple example is a rectangular aperture, as shown in the figure below. If the beam underfills the aperture, fractional coordinates of ±1 will not represent the edge of the aperture. When the beam overfills the aperture, aperture checking can be used to block unwanted rays.



Note that if **ebr** is adjusted to compensate for an unusual shaped aperture, it may be necessary to adjust paraxial quantities such as angle solves to maintain the correct focal length.

Central reference ray-aiming

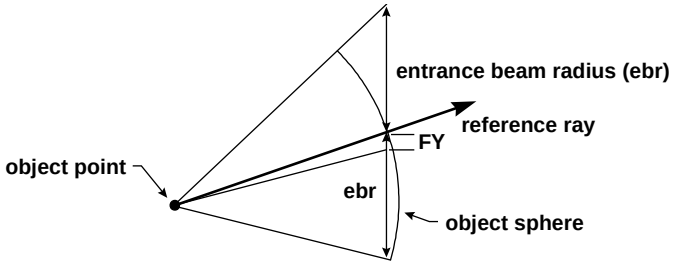
When the object point moves off axis, or when the pupil is decentered, several effects arise that complicate the definition of fractional aperture coordinates. The figure below shows the situation. Since surface 1 is generally not coincident with the entrance pupil, the projection of the pupil is displaced laterally from the axis.



The reference ray is the real ray from the off-axis object point that goes through the center of the aperture stop. The points FY = ±1 correspond to points located at ±**ebr** on the plane tangent to surface 1, measured from the intersection point of the reference ray, as shown.

Note that the above definition of fractional coordinates produces a reduction in pupil area proportional to the cosine of the field angle. In some wide-angle systems, pupil distortion is sufficiently large to overcome this reduction. To obtain a correct evaluation of such systems, it may be necessary to increase the numerical aperture on axis, and use a checked aperture on the aperture stop to limit the size of the axial beam. Then the off-axis beam will be sufficiently large to completely fill the aperture stop. OSLO has a special wide-angle ray trace mode that can deal with this effect automatically.

For paraxial ray aiming, the above definition of fractional aperture coordinates implies that the reference ray always passes through the point FY = 0. However, when aplanatic ray aiming is used, the reference ray is displaced from the point FY = 0, as shown in the figure below. This displacement results from the tilt of the object sphere with respect to the tangent plane, and is not the result of pupil aberration.



It is possible to specify that the reference ray should go through some point other than the center of the aperture stop. When a reference ray is traced, its fractional heights FYRF and FXRF relative to the aperture stop radius can be provided. The reference ray will be iterated to pass through the given point, and FY and FX will be defined with respect to its intersection with surface 1, as described above.

In addition to specifying non-zero coordinates for the reference ray in the aperture stop, it is possible to set up a reference surface that is different from the aperture stop. OSLO actually traces the reference ray to the point FYRF, FXRF on the reference surface (**rfs**). The default value of the reference surface number is coincident with the aperture stop surface number, but the two can be set independently.

As discussed previously, OSLO uses two different types of rays, called ordinary (Lagrangian) and reference (Hamiltonian) rays. The former are rays whose coordinates can be completely described in object space, while the latter involve coordinates that are defined inside the lens. Hamiltonian rays must be iterated to pass through the specified points, so they take longer to trace than ordinary rays.

The ray that is traced by the **sop** command is a Hamiltonian ray. In addition to passing through a precisely specified point on an interior reference surface, a Hamiltonian ray provides differential information needed to compute field sags.

Because of speed considerations, Hamiltonian rays should only be traced when necessary. Most evaluation routine (lens drawings, ray fans, spot diagrams, etc.) are set up to use Lagrangian rays. In optimization, rays can be specified to be either Lagrangian or Hamiltonian, but excessive use of Hamiltonian rays can substantially decrease program performance, sometimes without compensating increases in accuracy.

Rim reference ray-aiming

In this mode, in addition to the reference ray that is traced to define the center of the ray bundle from the current object point, three (or four, depending on the symmetry of the lens) additional "rim" reference rays are traced for each new object point. These rays are traced for the "top" (FYRF = 1, FXRF = 0), "bottom" (FYRF = –1, FXRF = 0), and "right" (FYRF = 0, FXRF = 1)

extremes of the reference surface and all surfaces that have "checked" apertures. If the lens is not symmetric, or FBX is not zero, the "left" (FYRF = 0, FXRF = –1) rim ray is also traced. Data from these rays is used to compute effective "vignetting" factors for the current object point, based on the object space fractional coordinates of the rays that define the extent of the transmitted ray bundle for that object point.

Use of this mode, in effect, allows the entrance beam radius or object numerical aperture to "float" so that the real, possibly vignetted, pupil is filled by the –1 <= FX, FY <= +1 ray bundle. Of course, it also means that the ray bundle size will be equivalent to that specified by the entrance beam radius only if the marginal ray height on the reference surface set by the entrance beam radius and the aperture radius of the reference surface "match." In the absence of vignetting and in rim reference ray mode, then, FX and FY are the (approximate) fractional coordinates (based on the aperture radius value) on the reference surface.

Extended aperture ray-aiming

Some imaging systems, and many illumination systems, collect light over greater than a hemispherical solid angle. OSLO has another ray aiming mode called XARM (eXtended Aperture Ray-aiming Mode) to handle such systems. Ordinarily, rays to be traced are specified by fractional object coordinates and fractional coordinates at the reference surface. Extended-aperture ray-aiming mode replaces the reference surface coordinate specification with the specification of two object space angles.

The direction of a ray in object space is specified by two Euler angles, *A* and *B*, which are defined similarly to the surface tilt angles **tla** and **tlb**. In terms of these angles, the direction cosines of a ray in object space are:

$$\begin{aligned} K &= -\sin B \\ L &= \sin A \cos B \\ M &= \cos A \cos B \end{aligned} \tag{6.14}$$

When XARM is on, the fractional coordinates FY and FX used in tracing single rays and ray fans are mapped to angles *A* and *B* through the following relations:

$$\begin{aligned} A &= A_{obj} + \mathrm{FY} * XABA \\ B &= B_{obj} + \mathrm{FX} * XABA \end{aligned} \tag{6.15}$$

Here, A_{obj} and B_{obj} are the *A* and *B* Euler angles of the normal to the object surface at the current object point. For an untilted plane object surface, $A_{obj} = B_{obj} = 0$. XABA is the extended-aperture beam angle specified by the **xarm_beam_angle** (**xaba**) command. The extended-aperture beam angle (specified in degrees) may be as large as 180 degrees. If the extended-aperture beam angle is 180 degrees, a *y*-fan defined by $\mathrm{FY}_{min} = -1.0$ and $\mathrm{FY}_{max} = +1.0$ will sweep through 360 degrees.

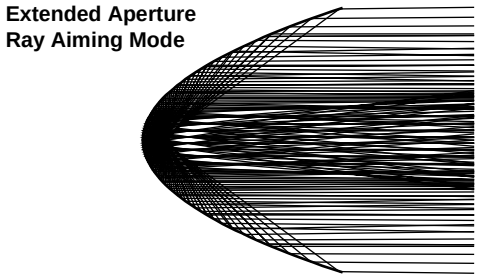
If spot diagrams are traced when XARM is on, the object point should be thought of as radiating uniformly into a cone whose vertex is at the object point and whose axis coincides with the normal to the object surface. **xaba** is the half-angle of the cone. Thus, if **xaba** is 90 degrees, the object point radiates uniformly into a hemisphere; if **xaba** is 180 degrees, the object point radiates uniformly in all directions.

For spot diagrams, the angular distribution of rays in object space is best described in spherical coordinates. Consider a sphere of radius *R* centered on the object point. Let the point at which the normal to the object surface pierces the sphere be the north pole of the sphere. For reference, construct a Cartesian coordinate system with its origin at the center of the sphere and its positive *z*-axis extending from the center through the north pole of the sphere. Any point on the sphere may be specified by two angles, θ and ϕ. Spot diagrams are calculated by tracing rays which pierce the sphere at points defined by a distribution of values for θ and ϕ. The Cartesian coordinates of such ray points are:

$$z = R\cos\theta$$
$$y = R\sin\theta\cos\phi \qquad (6.16)$$
$$x = R\sin\theta\sin\phi$$

When a spot diagram is calculated, the aperture divisions argument is used to calculate an increment for θ. This increment, which will be called $\Delta\theta$, is just the extended aperture beam angle divided by the aperture divisions value. The set of θ values to be used in calculating the spot diagram are 0, $\Delta\theta$, 2*$\Delta\theta$, … , *XABA*. For each such θ value, a set of *N* equally spaced ϕ values are chosen. The increment for ϕ, which will be called $\Delta\phi$, is chosen to be as close to $\Delta\theta$ as possible subject to the condition that *N*$*\Delta\phi$ = 360 degrees. The set of ϕ values used for the spot diagram are 0.5*$\Delta\phi$, 1.5*$\Delta\phi$, … , (*N* – 0.5)*$\Delta\phi$.

The figure below shows a paraboloidal mirror with a small but finite source located at its focus. XARM is turned on, with a beam angle XABA of 135 degrees.



Telecentric ray-aiming

Ordinary lenses are divided into two groups for ray tracing: lenses where the object distance is infinite, and lenses where the object distance is finite. OSLO considers object distances less than 1×10^8 to be finite, and object distances equal or greater than 1×10^8 to be infinite (defined as 1×10^{20}). These must be handled separately because it is not possible to trace a ray over a distance of 1×10^{20} without unacceptable loss of accuracy caused by numerical round-off.

For finite conjugate systems, rays are normally traced from the object surface, and aimed at the entrance pupil. However, if the entrance pupil is at infinity, the lens said to be telecentric, and all chief rays enter the system parallel to the axis. Then the chief rays cannot be traced between the object and the entrance pupil, because of the above-mentioned loss of accuracy.

OSLO has a general operating condition (**tele**) that defines the entrance pupil to be at infinity. In this mode, all chief rays (reference rays for new field points) will have an initial direction in object space that is parallel to the optical axis. The aperture stop and reference surface designations will be ignored, so in general, if no constraints are applied, the chief ray will not pass through the center of the aperture stop surface.

If you want chief rays to pass through the center of the stop, you can either use a chief ray solve or include chief ray heights on the stop surface as constraints in the optimization error function.

Telecentric mode is only valid for finite conjugate systems and is not used if wide-angle ray tracing mode is also enabled. The setting of **tele** of course has no effect on whether a system is telecentric in image space.

Afocal mode

An afocal system is one in which the principal points and focal points are at infinity. This does not necessarily imply that the object and image are at infinity. However, in the case where the image is at infinity, special procedures must be used in ray tracing because image-space rays can not actually be traced to infinity. The term *afocal on the image side* is sometimes used to describe the situation where the image is at infinity, whether or not the actual system is afocal.

OSLO has an evaluation mode called **afo** that causes image-space ray displacements to be reported in angular measure, optical path differences to be referred to a plane wave in the exit pupil of a system, and field sags to be reported in diopters. Afocal mode has no effect on single ordinary rays.

Astigmatic sources

Some types of laser diode emit coherent beams that are astigmatic, that is, the radius of curvature of the wavefront is different in different azimuths. This asymmetry is independent of the intensity distribution in the beam, and must be accounted for in ray tracing. OSLO can be set up to handle such astigmatic sources, assuming that they are located at the vertex of the object surface. The operating condition **sasd** gives the longitudinal distance between the effective source position for rays in the *xz* and *yz* planes.

Interpretation of ray data

In this section, the types of output data available from OSLO are summarized and presented for a standard lens. Most of this ray output data is readily understood. An exception is optical path difference. Before turning to specific examples, it is worth considering some general concepts used in OSLO ray tracing, particularly the convention of a current object point.

Current object point

The paradigm for ray tracing in OSLO consists of setting up an object point, which automatically traces a reference ray from the selected object point through a given point (usually the origin) of the aperture stop, then tracing one or more ordinary rays from this object point. Often ray trace output data compares two rays, typically the ray under study, and the reference ray.

In OSLO, all ray trace commands (excepting the **trace_ray_generic** command) assume that an object point has been defined using the **set_object_point** (**sop**) command. This command is identical to **trace_reference_ray** (**trr**), and provides values for the current object point fractional coordinates (FBY, FBX, FBZ), and the aperture-stop coordinates (FYRF, FXRF) of a reference ray traced from the object point. The reference ray is used to compute values for items needed to compute ray displacements, optical path differences, field sags, and other results that compare data for two rays. Because reference rays are always Hamiltonian rays, while ordinary rays are usually Lagrangian rays, in systems with large aberrations the interpretation of two-ray data can be complicated.

Changing any lens data invalidates any previous reference ray data. OSLO is set up to automatically retrace the current reference ray if necessary, whenever its data are needed. In such a case, the previous object point data are used. However, if a new lens is opened, the object point data is reset to default values (on-axis, center of aperture stop). To check the values of the current field point data, you can use the **sop** command with no arguments. The object point data is also available in the global variables trr_fby, trr_fbx, trr_fbz, trr_fyrf, and trr_fxrf.

Single ray trace

The most fundamental ray-trace analysis in OSLO is the single ray trace, which shows the trajectory of a single Lagrangian ray through an optical system (a Hamiltonian ray is traced if WARM is on). The trajectory of the ray is specified by a point on a surface and a direction of propagation following the surface. The point and direction can be specified in either local or global coordinates. The global coordinate command for ray data (**gcs**) applies to the whole system, so either all or no ray data are reported in global coordinates, unlike the case of global coordinates used to specify surface data.

The point specifying where the ray intersects the surface is given in the current coordinate system, whether it be local or global. Ray data are never given in the base coordinate system of a surface. The listing below shows typical output from the **tra ful** command in OSLO.

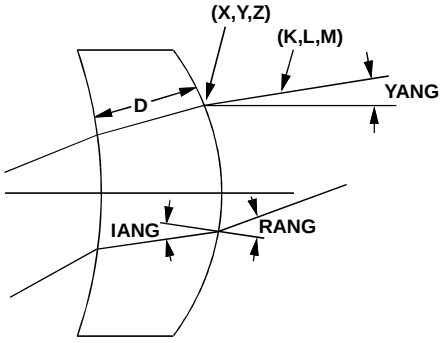
```
*TRACE RAY - LOCAL COORDINATES
 SRF        Y/L         X/K         Z/M      YANG/IANG   XANG/RANG      D/OPL
```

```
  3       1.836269   -0.136580      --         9.974625   -6.880999    5.864380
          0.172001   -0.118022    0.978001   12.040286   12.040286   10.683781
PUPIL       FY          FX                                             OPD
          0.300000    0.500000                                       14.202026
```

The direction of a ray is reported in three ways. The first gives the direction of a ray in terms of its direction cosines *K*, *L*, and *M*; that is, the projection of the normalized ray vector on the *x*, *y*, and *z* axes of the current coordinate system. Some internal calculations use *optical* direction cosines, which are normalized to the refractive index, rather than unity, but these are *not reported as ray output data*. The direction cosines of a ray are constrained by the requirement that their sum of squares be unity. Note in the listing that *Y* and *L* appear before *X* and *K*. The reason for this is that most rays are meridional rays, so specifying the *y* coordinates first produces a more compact notation.

Second, the direction of a ray is given by its angle in degrees, relative to the axes of the current coordinate system. Only two of the three angles (YANG and XANG) are reported.

Finally, the direction is given by the angles of incidence and refraction (IANG and RANG), also measured in degrees, relative to the surface normal at the point of incidence. The figure below shows the various items used to specify ray trajectories.



The last column of the text output from the single ray trace command displays the values of D and OPL. These are the distance measured along the ray from the immediately preceding surface, and the total optical distance from the object sphere, a sphere having its vertex at surface 1, and its center on the object point.

The last row of output from the single ray trace gives the pupil coordinates of the ray, in fractional units. At the far end of the last row is the optical path difference between the ray and the reference ray. This is only computed if the ray is traced completely through the lens.

Ray fans

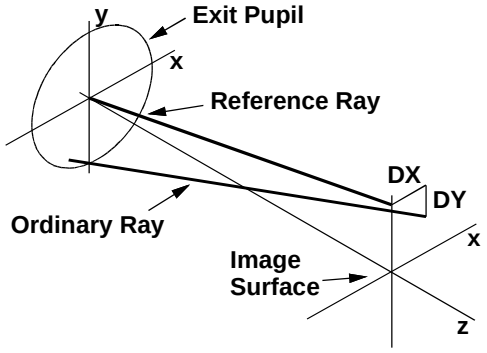
Ray fans provide more information per unit computational effort than any other ray tracing evaluation, so there are a number of options for ray intercept curves in OSLO. The following output shows the basic data computed by the **trace_fan** (**trf**) command. Note that there are ten columns of output; only the first seven will normally be seen (without scrolling) unless the text output window is extended.

```
*TRACE FAN - FBY  1.00, FBX  0.00, FBZ  0.00
RAY      FY         FRAC RFS      DYA          DXA          DY          DX          DZ          OPD          DMD
  1    1.000000    0.893566   -0.112783       --        0.136204       --          --        -0.875304    4.030302
  2    0.500000    0.431721   -0.059022       --       -0.023215       --          --         1.590462    1.205471
  3      --          --          --           --          --          --          --           --          --
  4   -0.500000   -0.410141    0.061566       --        0.064471       --          --         2.663131    0.937402
  5   -1.000000   -0.804004    0.123652       --        0.100441       --          --         9.810314    5.338150
```

The rays that constitute a fan are ordinary rays from the current field point (if WARM is off). The data show the fractional coordinate FY in object space, the fractional height of the ray on the reference surface (normalized to the aperture radius of the surface), the difference in ray slope between the ray and the reference ray (DYA and DXA), and the intersection coordinates (DY, DX, and DZ) of the ray with the image surface, relative to the intersection of the reference ray with the image surface (i.e., the ray *displacement*).

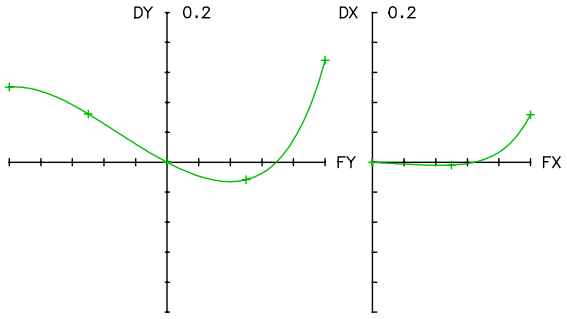
The last two columns contain the optical path difference between the ray and the reference ray, and the Conrady *D–d* aberration, with respect to the current reference ray. The DMD term is computed by multiplying the geometrical path between each surface by the dispersion between wavelengths 2 and 3 of the intervening medium. This provides an indication of the chromatic aberration of the system for the ray in question, without requiring the actual tracing of rays in different colors.

In the case of a ray failure, the value 1×10^{20} is printed in the data fields.



Ray intercept curves

Ray intercept curves are graphical plots of ray fans. To interpret the curves correctly, it is important to understand exactly what is being plotted. In the above case, the data presents image-space displacements as a function of object-space fractional coordinates. Obviously it is important that fractional aperture coordinates completely fill the aperture stop. The figure below shows the same data obtained above, in graphical format (a sagittal fan is also shown).

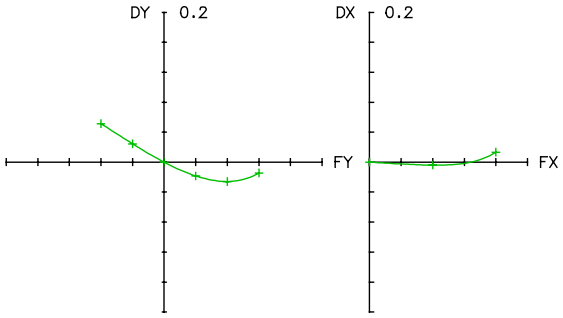


OSLO contains several options for convenient display of ray-intercept curves. For example, you can select whether you want to plot *y* or *x* displacements vs. *y* or *x* fractional coordinates. If your system has meridional symmetry, only half of the sagittal curves will be shown (as above).

The interpretation of ray intercept curves is described in Chapter 3, as well as throughout the standard optical design literature. However, there are some specific points that you should consider when working with ray-intercept curves in OSLO.

For large aperture systems used at finite conjugates, the ray intercept curves will be different for aplanatic and paraxial ray aiming, but since the beams extend over the same limiting aperture, the differences will generally be small. On the other hand, ray intercept curves may be clearly different for wide angle systems, depending on the setting of WARM, since the aperture will be different for the two cases.

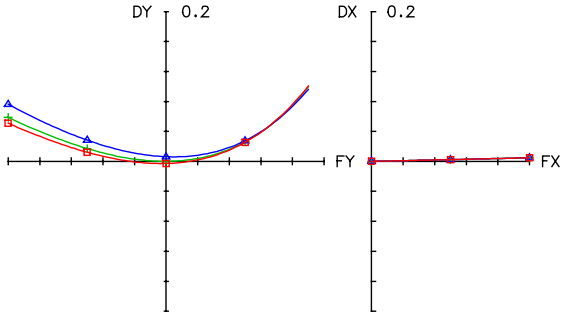
Ray-intercept curves will clearly differ depending on whether aperture checking is enabled. OSLO provides an operating condition (**apck**) that enables or disables checked apertures, to allow the designer to see whether apertures have been properly assigned to surfaces. The figure above shows ray-intercept curves for the demotrip lens with **apck** off, while the figure below shows the curves with **apck** on.

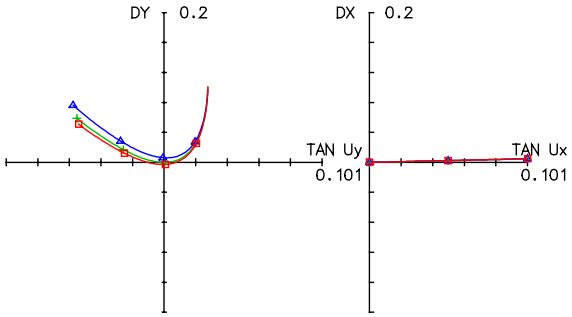


Systems that contain cylindrical lenses or other anamorphic elements need special consideration when setting up ray-intercept curves, because of the mapping of pupil coordinates. As described above, ordinary ray-intercept curves constitute a plot of image-space ray displacements vs. fractional pupil coordinates. In OSLO, object space coordinates are isomorphic, so that distances are measured in the same units in the *x* and *y* directions.

When a circular beam passes through an anamorphic element, one meridian becomes compressed relative to the other, so the output beam becomes elliptical. In such a case, the real distance corresponding to FY = ±1 is different from the real distance corresponding to FX = ±1. This can be confusing when interpreting ray-intercept curves.

OSLO provides an alternative to using fractional coordinates on the abscissa of ray-intercept curves, namely what are called *H*-tan*U* curves. *H*-tan*U* curves are produced using the **htnu** operating condition in the general operating conditions spreadsheet. If the operating condition is set to **on**, ray-intercept curves produced using the commands in the Calculate menu will be *H*-tan*U* curves. For *H*-tan*U* curves, the ray displacement on the image surface is plotted as a function of the tangent of the image-space ray slope. The figures below show a comparison of ordinary ray intercept curves (top) and *H*-tan*U* curves (bottom) for the anamorph.len system included in the OSLO public library. The curves show that the system is at the *xz* focus, and that the beam is compressed in the *yz* plane by a factor of two, relative to the *xz* plane.





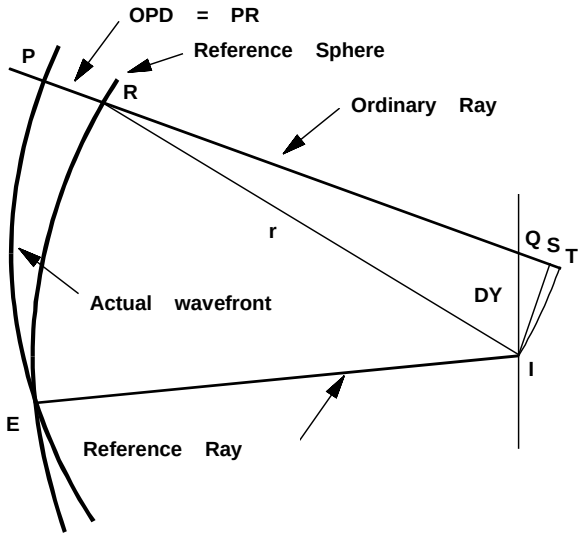
With *H*-tan*U* curves, a perfect image point always plots as a straight line, with a slope equal to the distance from the reference plane to the point (See, for example, Kingslake(5) or Smith(6)). In addition to plotting *H*-tan *U* curves, OSLO will plot the OPD as a function of the sine of the angle between the pupil ray and the reference ray. The ray's exit pupil position is related to sin *U*, so this plot is indicative of the relative size of the exit pupil in *x* and *y*.

The default scale when **htnu** is on is the largest value of tan *U* or sin *U* for the traced fans of rays. This value may not correspond to an edge of the paraxial pupil boundary. Note that the limits for tan *U* (and sin *U*) are not, in general, symmetric for an off-axis object point, even for an unvignetted system. For these reasons, interpretation of vignetting information from an *H*-tan *U* curve requires some care. You may enter a scale for the abscissa of the plots, if desired.

Optical path difference

Optical path difference (OPD) is used to measure wavefront aberration. A wavefront is the locus of points that are equal optical lengths from a point source, so a wavefront can be constructed from ray trace data. Optical path difference is the distance between an actual and an ideal wavefront propagated through the system under study. An ideal wavefront, in the present context, has a spherical shape, so the distance is between an actual (aberrated) wavefront and a reference spherical wavefront. An aberrated wavefront changes shape as it propagates, so the optical path difference depends on the radius of the reference sphere. For small aberrations, this is a small effect, but for large aberrations, it is easily calculated and observed experimentally.

The calculation of OPD can be described with reference to the figure below. The optical path difference compares the time of flight of light along a ray to that along the reference ray. In the figure, let a reference point on the image surface be given by *I*, which may be the point where the reference ray intersects the image, or more generally the point that minimizes the rms OPD. Consider a reference sphere centered on the point *I*, with a radius equal to *EI* = *RI* = *r*, the distance along the reference ray between its intersection with the exit pupil *E*, and the image point *I*, as shown. If an aberrated ordinary ray travels from the object to point *P* in the same time that the reference ray travels from the object to *E* (so that *P* lies on the actual wavefront from the object point), the optical path difference is defined to be the distance *PR* (times the refractive index) between the actual wavefront and the reference sphere, measured along the ray, as shown. A general operating condition (**opdw**) controls whether OPD is displayed in wavelengths (on) or current units (off).



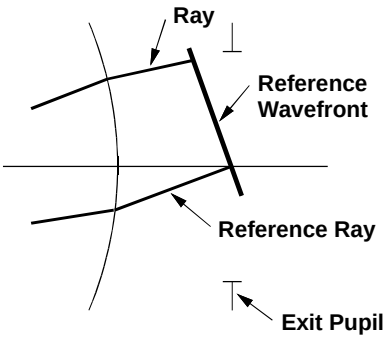
The line segment *QST* shows the effects of the radius of the reference sphere on the optical path difference. As the radius of the reference sphere is increased, the distance *ST* decreases and disappears when the radius of the reference sphere is infinite. The question naturally arises as to

the proper value for the reference sphere radius. In OSLO, the default value is the distance of the exit pupil from the image surface; for diffraction calculations, this amounts to the assumption that diffraction occurs at the exit pupil.

There is an operating condition (**wrsp**) that can be used to set the reference sphere radius to infinity or to the distance between the next to last surface and the last surface, in addition to the default value described above. It is sometimes necessary to set the reference sphere radius to infinity, if the aberration of the lens is extremely large, or if the exit pupil is close to the image plane. Then, a condition can arise where a ray fails to intersect the reference sphere, so the OPD cannot be calculated. OSLO will then issue the message “Reference sphere is inside caustic”.

The above figure shows that the point *I* where the reference ray intersects the image surface is not necessarily the optimum reference point for minimizing the optical path difference. A quick inspection shows that the OPD would be less if the reference sphere were tilted from its present position. In the spot diagram routines, where several rays are traced, the OPD is normally referenced to the point on the image surface that minimizes the variance of the OPD. When only a single ray is traced, there is insufficient information to make such an adjustment. However, the reference ray can be displaced in the pupil, which will usually change the OPD.

The optical path difference for an afocal system is a special case, owing to the fact that rays cannot be traced to the image surface. OSLO calculates the OPD in an afocal system as shown in the figure below. The reference wavefront is a plane that passes through the center of the exit pupil, and the OPD is the optical path length along the ray, minus the optical path length along the reference ray, as shown.



Non-sequential ray tracing

For a lens entered in the conventional manner, the surfaces are entered in the order in which light traverses the system. Each surface is traversed once by each ray and every ray traverses the surfaces in the same order. However, there are optical systems for which this type of ray tracing scheme is not adequate.

A simple example is the cube corner retro-reflector, which consists of three reflecting surfaces arranged as the corner of a cube. All rays incident within the aperture of the cube corner reflect once off each of the three reflecting surfaces and exit the cube corner parallel to their incident direction. The order in which any particular ray visits each surface depends upon its incident direction and relative aperture position. A system such as this must be treated as a non-sequential surface group, i.e., a group of surfaces whose positions in space are fixed but for which a prescribed ray trace sequence can not be defined.

Groups

In OSLO, a non-sequential surface group is treated as a sub-group of surfaces that is associated with a selected surface in the normal (sequential) portion of the lens. This surface is called the *entry port* for the non-sequential group. Surfaces in the non-sequential group itself are numbered in ascending order, although this order does not necessarily relate to the order in which rays strike the surfaces. The last surface in the group is called is called the *exit port* for the group.

When any ray reaches the entry port, a special ray tracing procedure is invoked that attempts to find a path through the non-sequential group to the exit port surface. Surfaces within the group are traversed in whatever sequence is necessary to produce a real path to the exit port, i.e., the path length from each surface to the next surface through the group is required to be positive. Total internal reflection is allowed and the same surface may be visited multiple times.

To define a non-sequential surface group, you should insert as many surfaces as you need (including the entry and exit ports), then select the surfaces and group them using the Non-sequential group command on the Edit menu. If you prefer to work in command mode, use the **lmo nss** and **lme** commands to define a non-sequential group.

The position and orientation of each surface within a non-sequential group is specified with respect to the local coordinate system of the entry port surface. The thickness of a group surface is not used, with the exception of the thickness of the exit port, which is the distance to the next sequential surface. The position of a group surface is specified by the three decentration coordinates (**dcx**, **dcy**, and **dcz**) and the orientation is specified by the three tilt angles (**tla**, **tlb**, and **tlc**). These items can be specified in the coordinates spreadsheet, which is accessed by activating the Special options button for the desired surfaces.

For non-sequential surface groups, a global coordinate (**gc**) command may only be used on the entry surface, since the remaining surfaces in the group are always positioned and oriented with respect to the entry port surface.

The position and orientation of each surface following the entry surface in a non-sequential group is required to be specified globally with respect to the entry surface. When a non-sequential group is created, either by executing the group command or by selecting the group item from the Edit menu in the Update Surface Data spreadsheet, tilt angles and decentered coordinates with respect to the entry surface are also calculated automatically for each surface following the entry surface.

Similarly, surface positions and orientations are converted back to a local representation when a non-sequential group is ungrouped. Note, however, that when a non-sequential group is ungrouped, the resulting lens will not necessarily make optical sense.

The surfaces in a non-sequential group are specified similarly to those in a sequential system. All of the surface types defined in OSLO are allowed, although the nature of non-sequential ray tracing may impede or prevent convergence of iterative ray trace algorithms under certain circumstances. Each surface is defined in the coordinate system described above, with the same sign conventions used for sequential surfaces.

Apertures

Apertures have a somewhat different meaning within non-sequential groups. If an aperture radius is specified for a group surface, it is assumed to apply to that part of the surface for which $(1 + \mathbf{cc})\ \mathbf{cv}\ z < 1$, where **cc** is the conic constant and **cv** is the curvature of the surface. Only that part of the surface for which the preceding condition is true is assumed to be present. Therefore, the surface is "open." A spherical surface, for example, will be treated as a hemisphere. If no aperture radius is specified for a spherical or elliptical group surface (or an *x*-toric surface with a spherical or elliptical *xz* cross section), the entire area of the closed surface is assumed to be present. Since paraxial rays are not traced through a non-sequential group, there are no solved apertures, so non-zero aperture radii must be entered explicitly for each surface.

If an aperture is specified for a group surface, a ray that falls outside the aperture is free to "pass by" the surface if aperture checking is off for that surface. If aperture checking is on, the region exterior to the surface is regarded as an infinite baffle, and a ray that reaches the surface via a real path will be stopped. Special apertures can be used on non-sequential surfaces, and have their normal effect.

The glass specification for a surface specifies the medium into which a ray is refracted or reflected, i.e., the medium into which the ray goes after it traverses the surface. In sequential ray tracing, this information is sufficient to trace the rays. For surfaces in a non-sequential group, however, the sequence is not necessarily known in advance. Moreover, rays may hit a surface

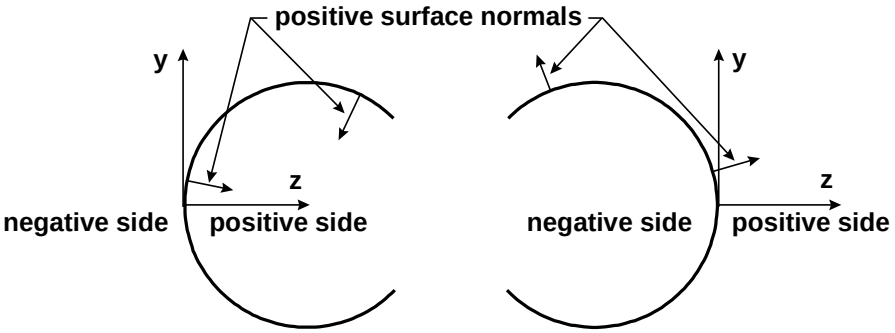
from either side. The surfaces are sitting in space and may be traversed in whatever order yields the shortest real path from surface to surface.

Actions

In a sequential system, rays intersect surfaces in a prescribed order, and reflect or refract according to the model set up by the designer. In a non-sequential system, there is usually a set of conditions that occur with greater probability than others. In a light pipe, we normally expect rays to strike the walls from the inside. This would be called the normal action of the wall surface. The refractive index to associate with a surface should be the one on the side following a normal action, the same as for a sequential surface. The normal action is defined by the user, and should describe the usual case. A special action can be defined to handle the alternate case.

For example, suppose that in a non-sequential group, surfaces 3 and 4 bound a BK7 glass lens in air. Suppose also that the normal action is that light enters surface 3, passes through the glass, and emerges from surface 4. The BK7 glass should be associated with surface 3, and AIR should be associated with surface 4. If rays were to be reflected by some other surface and enter the lens from the surface 4 side, it would be necessary to set up a special action for this case, as described below. However, it is not necessary to set up a special action for a case that never occurs.

In describing special actions, the two sides of a surface are called the *positive* side and the *negative* side. This nomenclature comes from considering the surface *near the vertex*. The positive side contains the +z axis, as shown below. A special action of *to positive* affects rays traveling towards the positive side. A special action of *to negative* affects rays traveling towards the negative side. Note that the normal action of a surface might be either *to positive*, or *to negative*, according to the particular setup.



The convention of the positive and negative side of a surface is also used to describe surface normals, as shown in the above figure. A positive surface normal points towards the positive side of a surface. Thus the surface normal points inwards for a positive surface, and outwards for a negative surface.

In the special actions spreadsheet, you can specify both a special action and a condition under which the action is to be taken. One action may be specified for ordinary rays and one action may be specified for the reference ray traced from a new object point. When no action is specified, or the condition specified for the action does not apply, the normal glass specification for the surface determines the action. If a special action is specified for ordinary rays but not for reference rays, the action specified for ordinary rays will apply to reference rays as well. The action and condition are chosen by clicking the appropriate cell in the spreadsheet and selecting the desired item from the pop-up list.

The possible special actions are:

- Pickup from group surface *surface_number*
- Reflect
- Obstruct
- Pass undeviated

- No action

The possible conditions are:

- To negative
- To positive
- First *n* hits
- Subsequent hits

The *pickup from group surface* action means that the ray is to be refracted into the medium defined by the glass entry on the specified group surface. For example, in the example above, if the ray is reflected and arrives again at group surface 4 traveling from positive to negative, it should refract from air into the glass BK7. Since AIR is specified for group surface 4, it is necessary to define a special action for group surface 4 for this case. The action would be pickup from group surface 3 and the condition would be *to negative*.

The special action *obstruct* might be used on the small secondary mirror of a 2 mirror system in which rays passing outside the aperture of the secondary mirror hit the primary mirror and reflect back to the secondary mirror. Rays that hit the secondary mirror from the rear should be obstructed. Thus, a special action *obstruct* would be specified for the secondary mirror under the condition *to positive*, and the normal glass specification REFLECT would then apply for rays arriving at the secondary from positive to negative. It would not be desirable for the reference ray to be obstructed, however, so the action *pass undeviated* might be specified for reference rays under the condition *to positive*.

The conditions *first n hits* and *subsequent hits* might be useful in modeling special situations. *First n hits* means that the specified action is to be taken the first *n* times a ray visits the surface. *Subsequent hits* means that the action is to be taken on all subsequent times the surface is visited.

It is possible to assign an ID number to a surface in a non-sequential group. The ID number is only used for lens drawings and is available for the user to designate which surfaces are to be drawn as elements, i.e., with connected edges. Since the entry sequence of the surfaces within a group is not important, there is no way to know, *a priori*, that, surfaces 3 and 4, say, are the front and back surfaces of a singlet. If possible, the lens drawing routines will draw all surfaces with the same (non-zero) element number as surfaces in a single element.

Array ray tracing

Lens arrays are aggregations of identical optical assemblies that differ only in their position. The assemblies may be regarded as occupying channels defined by array parameters which specify the location and orientation of each channel. Lens arrays are similar to non-sequential groups, in that the actual surfaces traversed by a ray are not known at the outset.

In OSLO, an array is defined with respect to a specified surface within the lens system called the channel surface. This surface encompasses all of the channels in the array. Channels are centered at specified (x, y) positions on this surface and may be shifted by varying amounts in the z direction.

The overall extent of a lens array is determined by the aperture of the channel surface. There are no special restrictions on the surfaces that comprise the optical assembly replicated at each channel location. The assembly may contain non-sequential groups, tilted and decentered surfaces, light pipes, grin rods, etc.

OSLO supports two types of lens arrays: regular arrays and tabular arrays. For regular arrays, channel centers are distributed in a uniform grid over the channel surface. For tabular arrays, channel centers are individually specified.

An **ary** command entered at a surface within the main lens system defines that surface to be the channel surface for an array, and indicates the type of array to be constructed. The optical assembly to be placed at each channel center is defined by the sequence of surfaces following the channel surface up to and including the end-of-array surface, which is identified by entering the **ear** command. If an array is created within the Surface Data spreadsheet (by activating the Special button and selecting either Regular Lens Array or Tabular Lens Array from the Surface Control pull-right menu) you will be prompted for the array type and the *ear* (end-of-array) surface.

The thickness entered at the *ear* surface specifies the distance from the array's channel surface to the surface following the end of the array – i.e., an automatic **rco** (return coordinates) to the channel surface is created at the *ear* surface. This avoids the need for a special end-of array surface encompassing all channels. If the surface following the *ear* surface has a **gc** (global coordinate) specification, the thickness at the *ear* surface is, of course, unnecessary.

Regular arrays

The command **ary** (**reg**, *x_spacing*, *y_spacing*, *y_offset*) defines a regular array. For this type of array, the channel centers are located at (*x*, *y*) coordinates

$x = i*x\_spacing$, where $i = 0, \pm1, \pm2, ...$
$y = j*y\_spacing + offset$, where $j = 0, \pm1, \pm2, ...,$

and

offset = *y_offset* if *i* is odd, or 0 if *i* is even.

The *z* coordinate of a channel center is the sag of the channel surface at the (*x*, *y*) coordinates of the channel center.

If a zero value is specified for *x_spacing*, all channel centers will lie along the *y* axis and *y_offset* will be ignored. Similarly, if *y_spacing* is zero all channel centers will lie along the *x* axis and *y_offset* will be ignored.

Channels are aligned so that the *z*-axis of the channel coincides with the normal to the channel surface at the channel center. The *y*-axis of the channel is oriented so that it is parallel to the *yz* plane of the local coordinate system of the channel surface.

Tabular arrays

The command **ary** (**tbl**, *number_of_channels*) initializes a tabular array and reserves memory for the specified *number_of_channels*. The center of each channel may then be specified by entering **ach** (array channel) commands:

ach (*channel_number*, *x_center*, *y_center*, *z_center*, *channel_tla*, *channel_tlb*, *channel_tlc*)

channel_number is an identifying number in the range 1 to *number_of_channels*. *x_center*, *y_center*, and *z_center* specify the coordinates of the channel center. The *z_center* value is added to the sag of the channel surface at the point (*x_center*, *y_center*). Thus if *z_center* = 0, the channel center will lie on the channel surface. The remaining arguments specify the *a*, *b*, and *c* tilt angles that are used to orient the channel relative to the local coordinate system of the channel surface. For tabular arrays, the normal to the channel surface is not used in orienting channels.

Tabular channels may be effectively deleted by entering the command **ach** (off, *channel_number*). The memory allocated for the channel is still retained, however, and channels are not renumbered. Thus the channel may be redefined by a subsequent ach command.

In the Tabular Lens Array data spreadsheet, the *channel_number* field may be activated to toggle channels on and off. Channel center coordinates and channel tilt angles are preserved when this is done so that this data need not be re-entered when an off channel is turned back on.

A lens array may be deleted by executing the **ard** command at the channel surface of the array. All array surfaces are retained as normal lens surfaces.

Ray tracing through lens arrays

Channel selection

When a ray has been traced to the channel surface of an array (and refracted or otherwise bent as determined by the properties of the surface) a channel is selected prior to continuing the trace. Channel selection is based solely on (*x*, *y*) coordinates. The channel whose center is closest to the point of intersection of the ray with the channel surface is the one selected. The *z* coordinate of the selected channel is obtained by finding the sag of the channel surface at the channel center and, if the channel is tabular, adding the specified *z_center* of the channel.

Once the coordinates of the channel center have been determined, a new coordinate system is established. The new coordinate system has its origin at the channel center and has its axes oriented as described above. The ray point is then transformed into this new coordinate system and ray tracing continues. When the ray has been traced to the *ear* (end of array) surface and refracted (or otherwise bent), a return is made to the original coordinates of the channel surface before the thickness on the *ear* surface is applied. Ray tracing then continues normally.

Channel clipping

The aperture (or any special apertures or obstructions) at the channel surface is used to clip the array channels. If the selected channel center is not obstructed and falls within the aperture (or any special apertures) ray tracing continues through the array. Otherwise, a ray failure occurs. Thus the extent of the array is determined by the channel surface aperture specification. This channel clipping occurs whether or not aperture checking has been requested at the channel surface. Note that it is possible for a ray to fall outside the channel surface aperture but still continue through the array, since the clipping criterion is only that the selected channel *center* falls within the aperture. However, if aperture checking has been requested at the channel surface, normal aperture checking is performed prior to channel selection and rays may be rejected prior to channel selection.

Random ray tracing

The normal method for tracing rays in OSLO involves setting an object point, and then tracing all rays from that object point until it is changed to a new one. This is a convenient way to work with most image-forming systems, but for systems that involve many field points, it becomes inefficient.

An example is the simulation of an extended source, which must be divided into a grid of pixels, each of which is characterized by an intensity and wavelength distribution. Unlike points, which have no width, pixels have a small but non-negligible size, determined by the accuracy requirements for the evaluation model.

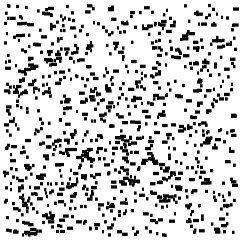
To handle ray tracing from extended sources, it is often convenient to employ random ray tracing, in which the field and aperture coordinates of a ray are random variables. OSLO provides a special command called **trace_ray_generic** (**trg**) in which all of the field, aperture, and wavelength data needed to trace a ray are given as arguments. Generic rays are ordinary single rays that bypass the data for the current wavelength, field point and reference ray. The **trace_ray_generic** command is not restricted to stochastic ray tracing, of course, but is useful whenever it is desirable to trace rays from other than the current field point.

OSLO provides three random number generators, **rand( )**, **grand( )**, and **lrand( )**, which provide random numbers with uniform, Gaussian, and Lambertian probability distributions. The Gaussian generator produces random numbers with a zero mean and unit variance. The Lambertian generator uses a probability density function equal to

$$\frac{\pi}{2}\cos\left(\frac{\pi}{2}x\right) \text{ for } 0 \le x \le 1 \quad \text{or} \quad 0 \text{ for } x < 0, x > 1 \tag{6.17}$$

which is useful in simulating the properties of an incoherent source.

The figure shown below shows a simple example of the use of generic rays to simulate the imaging of a single pixel in OSLO.



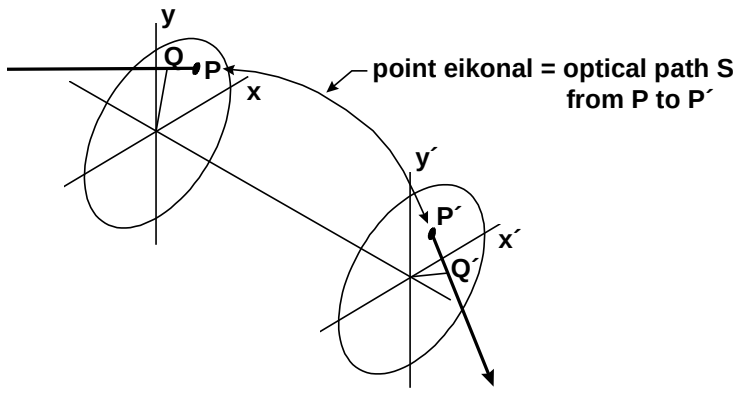
An SCP command used to produce the above figure is shown below. After initializing some input data, the command loops over wavelengths and wavelength weights, tracing rays from random field points through random aperture points to simulate light from an incoherent pixel. A more extensive command, which can be tailored to particular user requirements, is the ***xsource** SCP command that is supplied with OSLO.

```
*randray
// Displays image of a pixel centered at a given object point
// in an extended source, using stochastic ray tracing.
r = 500; // minimum number of rays/pixel/wavelength
s = .5; // fractional y height of center of pixel
t = .5; // fractional x height of center of pixel
u = .1; // pixel half-width, fractional coordinates
stp outp off;
//************************ Get paraxial image height
sbr(0,1);
paraxial_trace();
h = d1; // paraxial image radius
//************************ Setup window
graphwin_reset;
viewport(0, 1, 0, 1);
window(iso, -1.0, 1.0, -1.0, 1.0);
//************************ Loop over wavelengths


for (n = 0; n < numw; n++)
{
	pen(n + 2);
	k = 0;
	//********************* Loop over wvl wgts (assumed integers)
	for (i = 0; i < r*ww[n+1]; i++)
	{
		sbr(0, 1, 1e20);
		//***************** Get random ray inside beam radius
		do
		{
			y = 1 - 2*rand(0);
			x = 1 - 2*rand(0);
		} while (y*y + x*x > 1.0);
		//***************** Trace a random ray
		trace_ray_generic(std, loc, s + u*(1 - 2*rand(0)),
			t + u*(1 - 2*rand(0)), 0.0, y, x, n + 1, ims, 0);
		if (b1 != 1e20)
		{
			ya[k] = (a1/h - s)/(2*u);
			xa[k++] = (b1/h - t)/(2*u);
		}
		//***************** End of random ray block
	}
	if (k)
		polysymbol(xa, ya, k, 0, 0);
}
stp outp on;
```

Eikonal ray tracing

OSLO has two ways to simulate theoretical lenses that have specified properties: user defined surfaces and eikonal surfaces(7). User defined surfaces invoke a subroutine that connects the ray parameters before the surface (the point of intersection with a reference plane and two direction cosines) with the ray parameters after the surface. This is not as simple as it sounds. It is necessary to make sure that the transformation used is physically realizable. It is, for example, quite possible to create a user defined surface that simulates a lens that forms a perfect image for all magnifications. Unfortunately such a lens would be incompatible with Fermat's principle, and therefore not realizable. The subroutine that connects the ray in the object space with the ray in the image space must satisfy a number of constraints that are not always easy to deal with.

These difficulties fade away when a lens is described by an eikonal function. To do this, choose reference planes (x, y) and (x', y') perpendicular to the axis in object space and image space. Then the point eikonal $S(x, y, x', y')$ is defined as the optical path along a ray from the point $P(x, y)$ in the object side reference plane to the point $P'(x', y')$ in the image side reference plane.



Differentiation of the point eikonal by its four variables yields the ray directions:

$$\begin{aligned}
\frac{\partial S(x, y, x', y')}{\partial x} &= -nK \\
\frac{\partial S(x, y, x', y')}{\partial y} &= -nL \\
\frac{\partial S(x, y, x', y')}{\partial x'} &= -n'K' \\
\frac{\partial S(x, y, x', y')}{\partial y'} &= -n'L'
\end{aligned} \tag{6.18}$$

These formulas can be derived from Fermat's principle, which is therefore automatically satisfied. On the other hand we now have a practical problem if we want to follow a ray through the system in that the first two of the above equations must be solved for x' and y'. Usually the structure of the eikonal is rather complicated, so that an analytic solution of these equations is out of the question. Fortunately, OSLO has numerical techniques for solving these equations built in, so that from the user's point of view this difficulty does not exist.

Eikonal functions are always defined between two reference planes perpendicular to the axis, one in object space and one in image space. In OSLO eikonals are represented by a single plane surface, which plays the dual role of object side reference plane and image side reference plane. OSLO specifies an incident ray by the coordinates x and y of its intersection point with this surface and its direction cosines K and L. The eikonal routines use these data to generate x', y', K', and L' for the ray in image space. To be able to continue with the regular OSLO ray trace the program refers these four image side variables to the same plane lens surface used for the input data. Note

that the eikonal routines are restricted to real ray tracing only; paraxial calculations may therefore give erroneous results.

Let P and P' be the points of intersection of a ray with the reference planes in object space and image space, and let Q and Q' be the intersection points of perpendiculars dropped on the ray from the origins in the object space and the image space reference planes. Then the four eikonals in common use are the values of the following optical paths:

- point eikonal: $S(x, y, x', y') = PP'$
- point-angle eikonal: $V(x, y, K', L') = PQ'$
- angle-point eikonal: $V'(K, L, x', y') = QP'$
- angle eikonal: $W(K, L, K', L') = QQ'$

The properties of these eikonals upon differentiation can be summarized as follows:

$$\begin{aligned} dS &= n'\left(K'dx' + L'dy'\right) - n\left(Kdx + Ldy\right) \\ dV &= -n'\left(x'dK' + y'dL'\right) - n\left(Kdx + Ldy\right) \\ dV' &= n'\left(K'dx' + L'dy'\right) + n\left(xdK + ydL\right) \\ dW &= -n'\left(x'dK' + y'dL'\right) + n\left(xdK + ydL\right) \end{aligned} \tag{6.19}$$

So if, for example, V is differentiated by L' the result is $-n'y'$.

Eikonal for a spherical surface

To see how the eikonal ray trace works in practice, take any lens you are familiar with and replace one of its spherical surfaces by an equivalent eikonal function. The angle eikonal from vertex to vertex for a single spherical surface with radius r, separating media with refractive indices n and n', is given by

$$W\left(K, L, K', L'\right) = r\Delta M\left(\sqrt{1 + \frac{\Delta K^2 + \Delta L^2}{\Delta M^2}} - 1\right) \tag{6.20}$$

where

$$\begin{aligned} \Delta K &= n'K' - nK \\ \Delta L &= n'L' - nL \\ \Delta M &= n'M' - nM \end{aligned} \tag{6.21}$$

A CCL routine to calculate this eikonal is included in OSLO. To use it, first of all change any paraxial solves into direct specifications. Then choose a spherical surface in your lens, set its curvature equal to zero, and make it an eikonal surface. The program will ask you how many parameters the eikonal needs; your answer should be 1. Specify in the eikonal spreadsheet that the type of eikonal is ANGLE eikonal and that the name of the CCL procedure is SPHERIC. Then enter the radius of curvature of the surface as EI0. The lens can now be traced again, and the results should be the same as before.

The calculation is a bit slower on account of the great generality of the eikonal ray trace, but this only begins to present a problem when spot diagrams with many points need to be calculated. Calculating aberration curves and optimization data is more than fast enough for most applications.

Perfect lens

The point-angle eikonal of a perfect lens in air with magnification *m* and focal length *f*, using the perfect object plane and image plane as reference planes, is given by

$$V(x,y,L',M') = -m(xK'+yL') - m\frac{x^2+y^2}{2f} + F(x^2+y^2) \tag{6.22}$$

The function $F(x^2 + y^2)$ is related to the spherical aberration of the pupil. For a lens with perfectly imaged principal points we have

$$V(x,y,K',L') = -m(xK'+yL') - \frac{(1-m)^2}{m} f\sqrt{1+\frac{m^2(x^2+y^2)}{(1-m)^2 f^2}} \tag{6.23}$$

This eikonal can be used conveniently whenever a lens with a finite focal length is needed that is perfect at a finite magnification. When the object is at infinity this eikonal fails, because in that case the front reference plane cannot be placed in the object plane. The angle eikonal between the front principal plane and the back focal plane can be used instead:

$$W(K,L,K',L') = \frac{f}{M} - f\frac{KK'+LL'}{M} \tag{6.24}$$

This eikonal represents a lens perfectly corrected for infinity, again with the additional property that the principal points are free from spherical aberration. OSLO implements a perfect lens as a predefined surface type, rather than explicitly as an eikonal, although the same theory is used.

Variable conjugates

As an example of optical design using eikonals, consider a lens with the following specifications:

- focal length: 100 mm.
- object height: 3 mm.
- nominal magnification: –1.
- nominal numerical aperture: 0.5. The stop is located in the front principal plane.

The lens will be used for a range of object distances differing by ±3 mm from the nominal object distance. The lens has to image a 6 × 6 mm object field. A finite volume cannot be imaged perfectly, so the question arises how small the aberrations can be made. Fortunately the magnification in the center point is –1, so that Abbe's sine rule and Herschel's rule can both be satisfied. But the magnification varies by ±3 per cent, so the value of this advantage is questionable.

To determine the inevitable aberrations we use a mock ray tracing process. We describe the lens by a point-angle eikonal taken between the nominal object plane and image plane. We use a series development that accounts for the paraxial, the third order, and the fifth order terms, but, because the object height is small, we delete all terms that are of power higher than two in *x* and *y*.

We use the rotationally symmetric variables

$$\begin{aligned} a &= x^2 + y^2 \\ b &= xK' + yL' \\ c &= K'^2 + L'^2 \end{aligned} \tag{6.25}$$

and write

$$\begin{aligned} V(x,y,K',L') = b + a/200 + (E_0 + E_1 a + E_2 b + E_3 c)c^2 + \\ + (E_4 a + E_5 b + E_6 b^2)c + E_7 b^7 \end{aligned} \tag{6.26}$$

The first two terms are the paraxial terms, accounting for the –1 magnification and the 100 mm focal length. The other terms are related to the third and fifth order aberrations. The CCL code for this eikonal is shown below.

```
cmd eik0916a()
{
double a, b, c, aux;

a = Eik_x**2 + Eik_y**2;
b = Eik_x*Eik_kp + Eik_y*Eik_lp;
c = Eik_kp**2 + Eik_lp**2;
aux = b + a/200.0 + (Eik_cof[0] + Eik_cof[1]*a + Eik_cof[2]*b + Eik_cof[3]*c)*c*c;
Eik_path = aux + (Eik_cof[4]*a + Eik_cof[5]*b + Eik_cof[6]*(b*b))*c +
Eik_cof[7]*b*b;
}
```

The lens data are entered as follows. We use three configurations: the nominal system shown below, and systems in which the object distance is 203 mm and 197 mm, with paraxial image distances of 197.1 mm and 203.1 mm. The aberration coefficients E_0, … , E_7 correspond to the eikonal constants EI0, … , EI7.

```
*LENS DATA
Eikonal lens design - conjugates
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE   NOTE
  0        --        200.000000      3.000000            AIR

  1        --       -200.000000    115.470054 AS         AIR

  2        --       -200.000000      3.000000 S          AIR   *

  3        --        200.000000 V  121.470054 S          AIR

  4        --            --          3.022251 S

*CONFIGURATION DATA
TYPE  SN   CFG  QUALF       VALUE
TH     0    2     0    197.000000
TH     3    2     0    203.100000
TH     0    3     0    203.000000
TH     3    3     0    197.100000

*EIKONAL SURFACE DATA
  2   Point-Angle   eik0916a   8
       EI0      --       EI1       --      EI2       --      EI3       --
       EI4      --       EI5       --      EI6       --      EI7       --
```

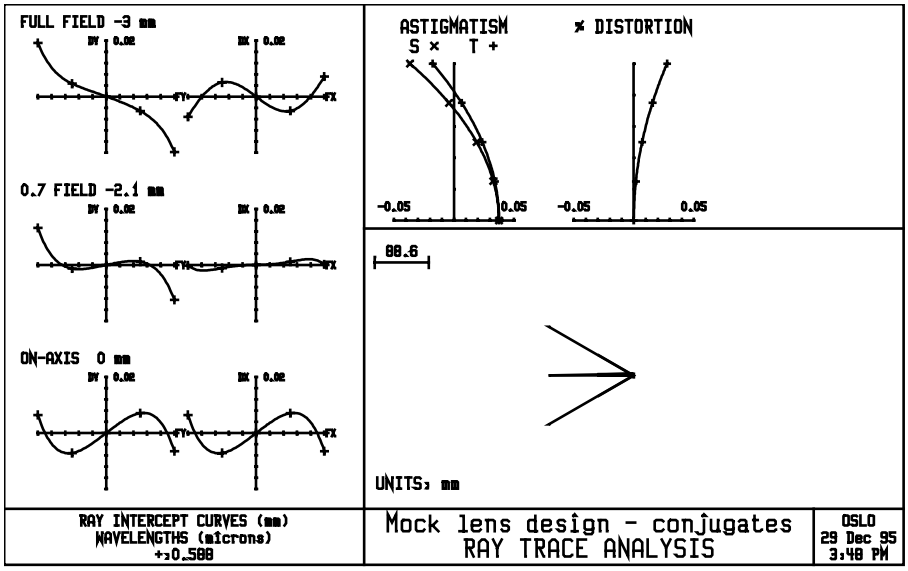
We now optimize the lens by using the OSLO default error function with OPD operands, minimizing it by varying the eikonal constants EI0, … , EI7 as well as the image distances for the three configurations. The results are as follows:

```
*VARIABLES
 VB   SN  CF  TYP       MIN         MAX        DAMPING      INCR        VALUE
V 1    2   -  EI0       --          --       437.365114  1.0000e-07    -0.036381
V 2    2   -  EI1       --          --       5.0111e+03  1.0000e-07     0.011276
V 3    2   -  EI2       --          --       684.913387  1.0000e-07    -0.000295
V 4    2   -  EI3       --          --       116.658680  1.0000e-07    -0.053174
V 5    2   -  EI4       --          --       1.9061e+04  1.0000e-07    -0.004126
V 6    2   -  EI5       --          --       2.8243e+03  1.0000e-07    -0.000756
V 7    2   -  EI6       --          --       4.6592e+03  1.0000e-07    -0.008006
V 8    2   -  EI7       --          --       1.8695e+04  1.0000e-07     0.001044
V 9    3   1  TH      0.100000  1.0000e+04  467.574477    0.011547   199.962437
V 10   3   2  TH      0.100000  1.0000e+04  448.299975    0.011547   203.063356
V 11   3   3  TH      0.100000  1.0000e+04  486.253613    0.011547   197.058727
*OPERANDS
 OP    DEFINITION                  MODE     WGT     NAME          VALUE   %CNTRB
O 8    "RMS"                         M    0.125000 Orms1         1.064088  12.77
O 23   "RMS"                         M    0.500000 Orms2         0.278954   3.51
O 38   "RMS"                         M    0.125000 Orms3         1.100865  13.67
O 46   "RMS"                         M    0.125000 Orms4         1.005903  11.41
O 61   "RMS"                         M    0.500000 Orms5         0.395249   7.05
O 76   "RMS"                         M    0.125000 Orms6         1.179024  15.68
O 84   "RMS"                         M    0.125000 Orms7         0.992325  11.10
O 99   "RMS"                         M    0.500000 Orms8         0.393443   6.98
O 114  "RMS"                         M    0.125000 Orms9         1.257561  17.83
MIN ERROR:     0.701888
```

The ray analysis below shows the aberrations that cannot be corrected, no matter how complicated we make the lens. Extreme high index glasses, grin elements, diffractive optical elements, aspherics, none of these will allow the lens to perform better than shown. Of course the lens must still be designed; but it is of great help to know what aberrations to aim for and when to stop working because the theoretical limit has been reached.

For comparison, the ray analysis can be repeated by setting EI0, … , EI7 equal to zero. This corresponds to the case in which the center plane is imaged perfectly. The result is that the error function is about a factor of three higher, so not insisting on perfection in the center plane makes the lens significantly better.

For more information on eikonal ray tracing, see Walther.1



Diffractive optics

In general, the term *diffractive optics* refers to those optical elements that utilize some fundamental structure periodicity along with the wave nature of light in order to change the direction of propagation of the light in a controlled way. There are different classes of diffractive optics, each with its own set of applications: classical ruled diffraction gratings, holographic optical elements, computer-generated holograms, binary optics, surface relief kinoforms. Although the details of the operation of these elements differ from type to type, the fundamentals of their operation contain many similarities. In this chapter, we will explore some of the basic principles of diffractive optics, with an emphasis on those areas related to the recent revival of interest in diffractive optics, namely the advanced manufacturing techniques (precision diamond machining, binary optics, laser writing systems) that allow for the production of very general diffractive optical surfaces.

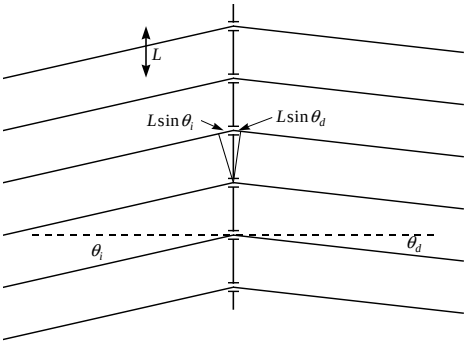
Most of the principles of operation for diffractive surfaces can be explained and understood by the consideration of a simple linear grating. Even though most diffractive surfaces do not have the constant periodicity of a grating, they do have an underlying periodicity of some sort and thus their behavior can be understood by analogy with the grating. Consider the grating illustrated in the figure below. It exhibits a structure with a fundamental period of *L*. At this point, it does not matter whether this structure is a pattern in amplitude transmission, phase transmission, or both; the important feature is that the structure repeats itself with a period of *L*. Confining our attention,

1 A. Walther, *The Ray and Wave Theory of Lenses*, Cambridge University Press, 1995.

for simplicity, to the case where the light is incident in a plane that is orthogonal to the plane of the grating, the propagation of light through the grating is described by the familiar *grating equation*:

$$n'\sin\theta_d - n\sin\theta_i = \frac{m\lambda_v}{L} \tag{6.27}$$

where n and n' are the refractive indices on the incident and diffracted sides of the grating, θ_i is the angle of incidence, θ_d is the angle of diffraction, λ_v is the vacuum wavelength of the light, and m is an integer known as the diffraction order. (The form of the grating equation used for ray tracing in OSLO is given on page 6-135.)



In contrast to the cases of refraction and reflection, the influence of the diffraction order m is seen to allow for multiple diffracted beams from a single incident beam. This can be considered a consequence of constructive interference; we would expect the diffracted light to constructively interfere at an observation point when the individual rays arrive with path length differences of 0, λ_v, $2\lambda_v$, $3\lambda_v$, etc.

Scalar diffraction analysis

A complete analysis of diffraction gratings requires the use of vector diffraction theory (Maxwell's equations). This is usually an extremely numerically intensive operation and it is sometimes difficult to draw general conclusions from the results of the analysis. Fortunately, many of the diffractive surfaces used in optical systems have periodicities that are many times the wavelength. In this case, we can approximate the optical field as a scalar quantity, and make use of the tools of Fourier optics. Although this results in a great simplification of the analysis, we should always keep in mind that it is an approximation, and as such, the conclusions are only as valid as the initial assumptions made in using the theory.

Diffraction grating

For simplicity, we will consider a one-dimensional grating and restrict the analysis to the *yz* plane, with the grating positioned along the *y*-axis at $z = 0$. In diffraction theory, an object is described by its *transmission function*, which is the ratio of the optical field exiting the object to the incident field. The grating has a fundamental period of L; this means that the transmission function $t(y)$ for the grating has the property $t(y) = t(y + L)$. Because of this periodicity, it is instructive to use the Fourier series representation for $t(y)$:

$$t(y) = \sum_{m=-\infty}^{\infty} c_m \exp\left(i2\pi m f_0 y\right) \tag{6.28}$$

where $f_0 = 1/L$ is the spatial frequency of the grating and the Fourier coefficient c_m is given by

$$c_m = \frac{1}{L}\int_0^L t(y)\exp\left(-i2\pi m f_0 y\right)dy \tag{6.29}$$

If $t(y)$ is a purely real and positive function, then the grating is referred to as an amplitude grating, while if $t(y)$ is a unit-modulus complex function, the grating is a phase grating. In general, $t(y)$ can

have both amplitude and phase components, but since the grating is a passive element $|t(y)| \leq 1.0$. For most applications, it is important to maximize the transmission of the system, so phase elements are preferred.

In the language of Fourier optics, in the $z = 0$ plane, a plane wave traveling at an angle θ_i with respect to the z-axis has the form

$$U_{plane-wave}(y, z=0) = \exp\left(i \frac{2\pi}{\lambda_v / n} y \sin\theta_i \right)$$

The field transmitted by the grating $U_t(y, z = 0)$ is given by the product of the incident field with the transmission function

$$\begin{aligned} U_t(y, z-0) &- U_{plane\ wave}(y, z-0) t(y) \\ &= \exp\left(i \frac{2\pi}{\lambda_v / n} y \sin\theta_i \right) \sum_{m=-\infty}^{\infty} c_m \exp(i 2\pi m f_0 y) \\ &= \sum_{m=-\infty}^{\infty} c_m \exp\left[i 2\pi y \left(\frac{\sin\theta_i}{\lambda_v / n} + m f_0 \right) \right] \\ &= \sum_{m=-\infty}^{\infty} c_m \exp\left(i 2\pi y \frac{\sin\theta_d}{\lambda_v / n'} \right) \end{aligned} \tag{6.31}$$

The last line of Eq. (6.31) can be interpreted as a series of plane waves, where the propagation angles of the plane waves are given by

$$\frac{\sin\theta_d}{\lambda_v / n'} = \frac{\sin\theta_i}{\lambda_v / n} + m f_0 \tag{6.32}$$

It is easy to see that Eq. (6.32) is just the grating equation. Thus the diffracted field consists of a set of plane waves, traveling at diffraction angles given by the grating equation, with amplitudes equal to the value of the Fourier coefficient c_m. The diffraction efficiency η_m is the fraction of the incident energy that is diffracted into order m. This efficiency is given by the squared modulus of the Fourier coefficient:

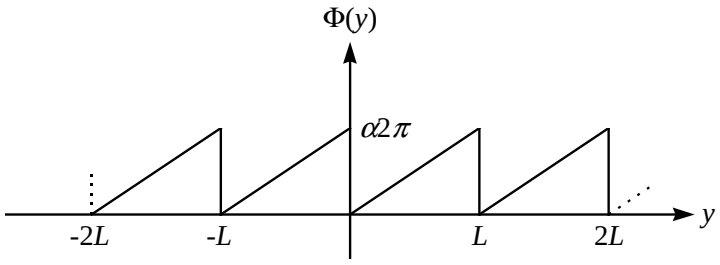
$$\eta_m = c_m c_m^* \tag{6.33}$$

where the asterisk denotes the complex conjugate. Note that even though $t(y)$ may have a discontinuity at the boundaries between periods, Eq. (6.28) is an expansion in continuous basis functions, so that Eq. (6.31) is an expansion in continuous diffracted waves.

Equation (6.31) illustrates one very important fundamental property: the grating frequency determines the direction of propagation of the diffracted orders while the transmission properties of an individual period [used to calculate c_m via Eq. (6.29)] determine the distribution of energy among the various diffracted orders. All gratings of frequency f_0 will diffract light at angles given by the grating equation, but the energy distribution among the orders will be a function of the single period transmission function. It is important to realize that this conclusion is valid regardless of whether scalar or vector diffraction theory is applied to the grating: the diffracted waves propagate at angles that are given by the grating equation. The use of a rigorous electromagnetic grating theory will result in a more accurate prediction of the diffraction efficiency of a grating but will not change the diffraction angles of the diffracted waves.

As mentioned above, based on throughput considerations we are generally interested in phase gratings. The desired phase modulation may be introduced by a variation in refractive index across a constant thickness of material or, more commonly, by a surface-relief profile between two media of differing refractive indices. These surface relief diffractive structures, sometimes called

kinoforms, have been subject of most of the recent research and development in diffractive optics. Consider the sawtooth phase function $\Phi(y)$ illustrated below.



Within each period, the phase function is linear, reaching a maximum value of $\alpha 2\pi$. One can show that for this transmission function $t(y) = \exp[i\Phi(y)]$, the diffraction efficiency is

$$\eta_m = \frac{\sin^2\left[\pi(\alpha - m)\right]}{\left[\pi(\alpha - m)\right]^2} = \mathrm{sinc}^2(\alpha - m) \qquad (6.34)$$

The function $\sin(\pi x)/(\pi x)$ is often called the *sinc*(x) function. The sinc function is equal to unity if its argument is zero and is equal to zero for any other integer argument. Recall that m, the diffraction order, is an integer value. We see from Eq. (6.34) that if $\alpha = 1$, i.e., if there is exactly 2π of phase delay at the edge of each grating period, then $\eta_1 = 1.0$ and all other $\eta_m = 0$. Thus, we would have a grating with a diffraction efficiency of 100%: all of the incident light is diffracted into the first diffraction order. It is this potential for very high diffraction efficiencies, coupled with modern manufacturing techniques, that has stimulated much of the recent interest in diffractive optics.

As mentioned earlier, the phase function is usually implemented as a surface-relief pattern. For a thin structure, the optical path difference (OPD) introduced by the structure is $(n' - n)d(y)$, where $d(y)$ is the height profile of the surface. Thus, the phase function Φ for a wavelength λ is just $\Phi(y) = (2\pi/\lambda)[n'(\lambda) - n(\lambda)]d(y)$. For this sawtooth pattern, the only parameter to be chosen is the maximum thickness d_{max}. This value is chosen such that exactly 2π of phase delay is introduced for a specified wavelength, usually called the design wavelength and denoted by λ_0. If this phase is to have a maximum value of 2π, this means that the maximum thickness d_{max} is

$$d_{\max} = \frac{\lambda_0}{\left|n'(\lambda_0) - n(\lambda_0)\right|} \qquad (6.35)$$

It is worthwhile to point out that for typical refractive materials used for visible light and in air, $|n' - n|$ is about 0.5, so d_{max} is about two wavelengths.

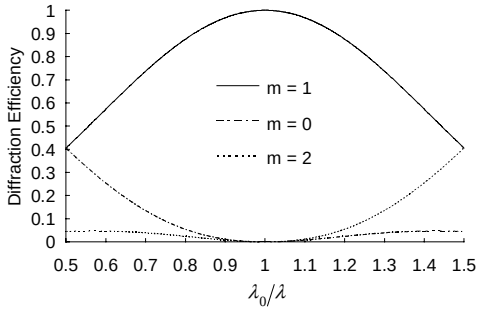
For wavelengths other than the design wavelength, the maximum phase delay is $\Phi_{\max} = (2\pi/\lambda)[n'(\lambda) - n(\lambda)]d_{max}$. Using Eq. (6.35), we find that the wavelength detuning parameter α is given by

$$\alpha(\lambda) = \frac{\lambda_0\left[n'(\lambda) - n(\lambda)\right]}{\lambda\left[n'(\lambda_0) - n(\lambda_0)\right]} \qquad (6.36)$$

Usually, the effect of the material dispersion components of Eq. (6.36) is dwarfed by the wavelength terms, and $\alpha(\lambda)$ can be approximated by λ_0/λ. With this approximation, the diffraction efficiency as a function of wavelength and diffraction order takes the form

$$\eta_m(\lambda) = \mathrm{sinc}^2\left(\frac{\lambda_0}{\lambda} - m\right) \qquad (6.37)$$

The diffraction efficiency as calculated from Eq. (6.37) is illustrated in the figure below for the orders $m = 0$, $m = 1$, and $m = 2$.



It is often useful to know the average diffraction efficiency $\overline{\eta_m}$ over the spectral band of interest (from λ_{min} to λ_{max}):

$$\overline{\eta}_m = \frac{1}{\lambda_{\max} - \lambda_{\min}} \int_{\lambda_{\min}}^{\lambda_{\max}} \eta_m(\lambda)\, d\lambda \tag{6.38}$$

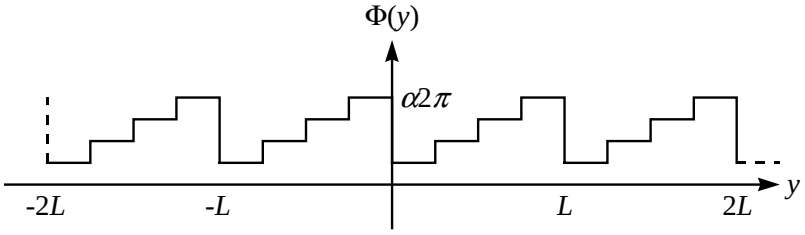
We can find an approximate expression for $\overline{\eta_m}$ by using a power series expansion of Eq. (6.37) and integrating term by term. For the usual case of $m = 1$, the result is

$$\overline{\eta}_1 \cong 1 + \frac{\pi^2}{3\lambda_0}\left(\lambda_{\min} + \lambda_{\max} - \lambda_0\right) - \frac{\pi^2}{9\lambda_0^2}\left(\lambda_{\min}^2 + \lambda_{\min}\lambda_{\max} + \lambda_{\max}^2\right) \tag{6.39}$$

If the spectral band is symmetric about λ_0, with a width $\Delta\lambda$, so that $\lambda_{min} = \lambda_0 - \Delta\lambda/2$ and $\lambda_{max} = \lambda_0 + \Delta\lambda/2$, then Eq. (6.39) simplifies to

$$\overline{\eta}_1 \cong 1 - \left(\frac{\pi\Delta\lambda}{6\lambda_0}\right)^2 \tag{6.40}$$

Photolithographic techniques are often used to produce a stair-step approximation to the ideal sawtooth profile, as illustrated in the figure below for the case of four discrete phase levels. This process is known as *binary optics*, since each photolithographic mask and etch step increases the number of phase levels by a factor of two.



If the linear phase is approximated by P equally incremented constant phase levels ($P = 4$ in the above figure), the diffraction efficiency is given by

$$\eta_{m,P}(\lambda) = \frac{\operatorname{sinc}^2\left(m/P\right)}{P^2} \frac{\sin^2\left[\pi(\alpha - m)\right]}{\sin^2\left[\frac{\pi}{P}(\alpha - m)\right]} \tag{6.41}$$

In the limit as $P \to \infty$, the binary optics profile approaches the continuous linear profile, and Eq. (6.41) reduces to Eq. (6.24). For

the important case of $m = 1$ and $\alpha = 1$ (i.e., $\lambda = \lambda_0$), Eq. (6.41) reduces to

$$\eta_{1,P}(\lambda_0) = \mathrm{sinc}^2\left(1/P\right) \tag{6.42}$$

If N photolithography steps are performed, the resulting diffractive surface has $P = 2^N$ phase levels. So, for example, if $N = 4$, then $P = 2^4 = 16$ and Eq. (6.42) predicts a peak efficiency of 98.7%. In practical situations, P is often large enough so that $P^2 \sin^2[(\pi/P)(\alpha - m)]$ can be approximated by $P^2[(\pi/P)(\alpha - m)]^2 = [\pi(\alpha - m)]^2$. For large enough P, then, Eq. (6.41) is approximated by

$$\eta_{m,P}(\lambda) \cong \mathrm{sinc}^2\left(m/P\right)\mathrm{sinc}^2(\alpha - m) \tag{6.43}$$

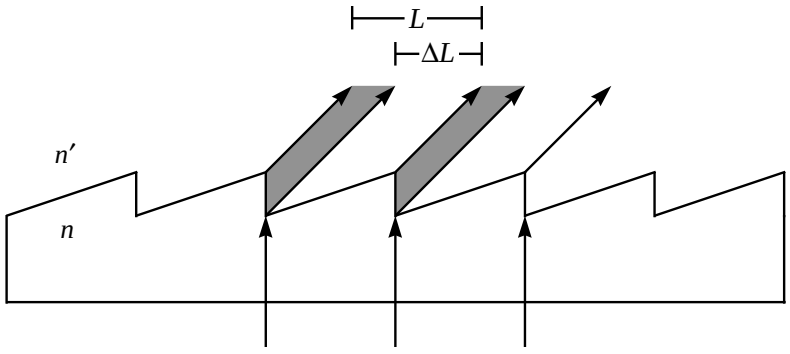
Equation (6.43) can be considered as the product of the "binary optics term" $\mathrm{sinc}^2(m/P)$ with the ideal sawtooth phase efficiency $\mathrm{sinc}^2(\alpha - m)$.

Extended scalar theory

The analysis in the previous section made the assumption that the optical field could be approximated as a scalar quantity and the diffracting structure could be modeled as a thin screen that could be analyzed by a simple transmission function. These approximations are only valid when the ratio of wavelength-to-period (λ/L) is small. When this ratio is not small, an accurate analysis requires the use of vector diffraction theory. This is a well-developed area of research and there are several theories that may be used for accurate computations of diffraction efficiency for gratings of arbitrary λ/L ratios. Generally, these computations are extremely numerically intensive and time consuming.

For the diffractive optics used in imaging systems, we usually find λ/L ratios that are neither very small (in which case we could use the scalar theory developed in the previous section) nor very large (requiring the use of vector diffraction theory). This suggests that it would be useful to find a "middle ground" theory, i.e., something more accurate than the simple scalar theory but not as time consuming as vector theory. One approach to this problem is the "extended scalar theory" developed by Swanson(8). The extended scalar theory attempts to avoid the assumption that the diffracting structure is infinitely thin.

The extended scalar theory uses ray tracing to approximate the field exiting the structure and then applying scalar diffraction theory. Consider tracing rays through the surface profile illustrated below. For this situation, ray tracing considerations would indicate that the finite thickness of the structure leads to "shadowing" effects. The light distribution exiting the grating does not fill the aperture. We would expect this underfilled aperture to have a lower diffraction efficiency than a filled aperture.



Given a grating period L and an angle of incidence θ_i, we can find the rays that are transmitted by each period of the sawtooth surface profile. (We assume that the incident wave can be approximated as locally planar.) By keeping track of the optical path of the two extreme rays, we can find the effective maximum phase difference α across a grating period. Also, we can find the position ΔL where the transition from light to no light occurs. Given these values, the transmission function for the fundamental period of the grating becomes

$$\begin{aligned} t(y) &= \exp\left(i\frac{\alpha 2\pi}{\Delta L}y\right) && 0 \le y \le \Delta L \\ &= 0 && \Delta L \le y \le L \end{aligned} \tag{6.44}$$

Using this transmission function in Eq. (6.29), we find that the diffraction efficiency is given by

$$\eta_{m,ext} = \left(\frac{\Delta L}{L}\right)^2 \operatorname{sinc}^2\left(\alpha - \frac{\Delta L}{L}m\right) \tag{6.45}$$

In the limit of zero thickness, $\Delta L/L = 1$, and Eq. (6.45) reduces to the scalar result of Eq. (6.34). The primary effect of the finite thickness is the reduction of the diffraction efficiency by the $(\Delta L/L)^2$ factor. The quantity $\Delta L/L$ is the ratio of the area filled with light to the total area and is called the *duty cycle*.

A key feature of the extended scalar theory is that the efficiency is still given by an analytic expression, not the result of a numerical calculation. Almost all of the quantities needed to evaluate Eq. (6.45) are available from the ray trace; the only exception is the depth of the sawtooth surface profile. In OSLO, you may enter the depth directly, or use what Swanson terms the "optimum depth." The optimum depth is found by requiring the propagation direction for normally incident light to be the same regardless of whether Snell's law or the grating equation is used. The optimum depth d_{opt} for the design wavelength λ_0 and diffraction order m (usually 1) is

$$d_{opt} = \frac{|m\lambda_0|}{\left|n - \sqrt{n'^2 - \left(\frac{m\lambda_0}{L}\right)^2}\right|} \tag{6.46}$$

Note that the optimum depth is a function of the period L; the smaller the period, the larger the influence of the finite thickness of the structure on the path lengths of rays traced through the surface relief profile. In the limit of very small wavelength-to-period ratios ($\lambda_0/L \to 0$), the optimum depth simplifies to the scalar value, Eq. (6.35). In OSLO, if you enter a value of zero for the depth of the surface relief profile, the optimum depth as computed by Eq. (6.46) will be used.

It should be stressed that this extended scalar theory is still only an approximation to the real physics. It has the advantage of correctly modeling the observed physics that the simple scalar theory does not, i.e., a decrease in diffraction efficiency as the wavelength-to-period ratio increases. Swanson shows that there is good agreement between the extended scalar theory and the exact vector theory for wavelength-to-period ratios as large as 0.5. This includes the majority of diffractive optics that are used as lenses and/or aspheric correctors. For larger values of λ/L, one must use a vector theory, particularly for the accurate modeling of polarization-dependent properties. The purpose of the extended scalar theory is to provide a reasonable estimate of diffraction theory so that the effects of non-unity efficiency may be modeled and potential trouble spots identified. It is not meant to replace rigorous electromagnetic grating theory.

The diffraction theory presented above suggests that we can separate the design of a diffractive optical element into two parts: *i*) the specification of the grating spacing and *ii*) the design of the structure of each grating period. The grating spacing determines the direction of propagation of the diffracted light for each order of interest. The structure of the grating period determines the diffraction efficiency, i.e., how much energy is diffracted into each order. The geometrical optics parameters are completely governed by the grating spacing; thus, this is usually the primary

function of interest during the design phase of a project. Given a desired distribution of grating periods, it is then necessary to compute the diffraction efficiencies, which are always a function of the type of diffractive element (amplitude transmission, phase transmission, etc.) and manufacturing process (diamond turning, binary optics, holography, etc.). Any diffracting surface relief structure can be considered to be a collection of locally constant-period gratings, so the diffraction efficiency calculations presented above may be applied to an arbitrary surface-relief diffracting structure, as long as the phase function is locally "blazed" in the manner of the sawtooth grating.

Aberrations of diffractive elements

We now consider the aberrations introduced by diffractive elements in an optical system. Throughout this section we will be concerned with rotationally symmetric phase surfaces and their use in both monochromatic and spectrally broadband systems. We will assume that the design diffraction order is the first ($m = 1$), as is usually the case in practice. Thus, in an effort to avoid notational clutter, the order m will not be explicitly shown in the analysis. The phase function we will be using is the **DFR** type in OSLO:

$$\Phi(r) = \frac{2\pi}{\lambda_0}\left(DF0 + DF1\, r^2 + DF2\, r^4 + DF3\, r^6 + \quad\right) \tag{6.47}$$

Note that this phase function is a continuous function of r, indicating that the diffracted waves are also continuous. This is justified by the discussion following Eq. (6.33). When implemented as a kinoform, the surface of the diffractive lens defined by Eq. (6.47) is discontinuous; the discontinuities (diffracting zone boundaries) occurring when $\Phi(r) = j2\pi$, where j is an integer. The *DF0* term is just a constant phase offset and has no effect on the imaging performance; we will assume that $DF0 = 0$. The paraxial properties of the lens are completely described by the *DF1* term. Paraxially, the transmission function of a diffractive lens is

$$t_{diff,parax}(r) = \exp\left(i\frac{2\pi}{\lambda_0} DF1 r^2\right) \tag{6.48}$$

Standard Fourier optics theory says that the transmission function for a lens of focal length f (and power $\phi = 1/f$) is

$$t_{lens}(r) = \exp\left(-i\frac{\pi}{\lambda f} r^2\right) = \exp\left(-i\frac{\pi\phi}{\lambda} r^2\right) \tag{6.49}$$

Comparison of Eqs. (6.48) and (6.49) reveals two important facts. First, at the design wavelength λ_0, the paraxial power ϕ_0 is equal to $-2\,DF1$. In other words, given a design focal length f, the r^2 coefficient should be $-1/(2f)$. Also, the paraxial power as a function of wavelength is

$$\phi(\lambda) = \frac{\lambda}{\lambda_0}\phi_0 \tag{6.50}$$

Using Eq. (6.50), we can compute an Abbe value ν_{diff}, defined over the wavelength range from λ_{short} to λ_{long}:

$$\nu_{diff} = \frac{\lambda_0}{\lambda_{short} - \lambda_{long}} \tag{6.51}$$

Equation (6.51) reveals several interesting features of diffractive lenses. First note that the value of ν_{diff} depends only on the wavelengths and is independent of any material-type parameters. All diffractive lenses have the same dispersion (over the same wavelength range); there are no "crown" or "flint" diffractive optics. Also, since λ_{long} is greater than λ_{short}, ν_{diff} is always negative. This means that the dispersion is in the opposite sense from conventional refractive materials. For

glasses, the refractive index is higher for blue light than for red light, so the lens has more power in the blue. On the other hand, from Eq. (6.50) we see that a diffractive lens has more power in the red (longer wavelength). This can also be seen from the grating equation, which implies that red light is "bent" more than blue. Finally, the absolute value of ν_{diff} is much smaller than the Abbe value for conventional refractive materials. Since the Abbe value is a measure of inverse dispersion, a small value of ν means more chromatic dispersion. This should not be surprising, given the wide use of gratings as dispersive elements in devices like spectrometers. To illustrate these points, consider the usual visible spectrum defined by the d, F, and C lines. Using these wavelengths, the value of ν_{diff} is –3.45.

Despite their dispersive properties, much of the recent work in system design with diffractive optics has, in fact, been in spectrally broadband systems, where diffractive lenses are used in conjunction with refractive and/or reflective elements. As a side note, an all-diffractive achromatic system can be made, but it has some peculiar properties. The most unattractive feature of such a system is that the achromatic image formed by an all-diffractive system must be virtual, although a conventional achromat may be used to form a final real image. Since diffractive lenses are very thin structures, by putting a diffractive surface on one side, a refractive singlet can be made into an aspherized achromat with only a tiny change in weight or bulk. Diffractive lenses have been proposed for use in both achromatic systems and systems in which a large amount of chromatic aberration is desired.

We have seen that the *DF1* coefficient determines the paraxial properties of the lens. The fourth-order coefficient *DF2* affects the Seidel and higher order aberrations; the sixth-order coefficient *DF3* affects fifth (ray) and higher order aberrations, etc., in a manner akin to aspheric surfaces on a refractive lens. A simple way to derive the aberration coefficients for a diffractive lens is to make use of the Sweatt model. One can start with the familiar thin lens aberration coefficients and take the limit as the refractive index n approaches infinity and the thin lens surface curvatures c_1 and c_2 approach the diffractive lens substrate curvature c_s. Assuming that the aperture stop is in contact with the diffractive lens, we find that the spherical aberration is a quadratic function of both the bending of the lens and the conjugates and that coma is a linear function of these two parameters. This functional dependence is the same as for a thin, refractive lens, although the forms of the aberration coefficients themselves are, of course, different. We also find, for this stop-in-contact case, that the astigmatism is a function only of the power of the lens and that the distortion is zero; these dependencies are also the same as the refractive lens. The major difference between the diffractive lens and the refractive lens is in the Petzval curvature coefficient; for a diffractive lens, the Petzval term is zero. Thus, one can add diffractive power to a system without changing the Petzval sum. (Recall that the thin lens Petzval coefficient is proportional to $1/n$; as $n \rightarrow \infty$, the Petzval coefficient approaches zero.) If the aperture stop is not in contact with the diffractive lens, the aberration coefficients may be found by using the standard stop-shift equations.

---

1 G.H. Spencer and M.V.R.K. Murty, "General Ray Tracing Procedure", J.Opt.Soc.Am. **52**, 672-678 (1962).
2 W. T. Welford, "A vector raytracing equation for hologram lenses of arbitrary shape," Opt. Commun. **14**, 322-323 (1975).
3 W. C. Sweatt, "Describing holographic optical elements as lenses," J. Opt. Soc. Am. **67**, 803-808 (1977); "Mathematical equivalence between a holographic optical element and an ultra-high index lens," J. Opt. Soc. Am. **69**, 486-487 (1979).
4 W. A. Kleinhans, "Aberrations of curved zone plates and Fresnel lenses," Appl. Opt. **16**, 1701-1704 (1977).
5 R. Kingslake, *Lens Design Fundamentals*, Academic Press, 1978, pp. 143-144.
6 W. J. Smith, *Modern Optical Engineering*, Second Edition, McGraw-Hill, 1990, pp. 79-84.
7 The eikonal ray tracing algorithms in OSLO and the material in this section were contributed by A. Walther, Worcester Polytechnic Institute.
8 G. J. Swanson, "Binary Optics Technology: Theoretical Limits on the Diffraction Efficiency of Multilevel Diffractive Optical Elements," Massachusetts Institute of Technology, Lincoln Laboratory, Technical Report 914, 1 March 1991.

Chapter 7 Image evaluation

The goal of most optical design projects is to assemble an optical system that performs a certain task to a specified degree of accuracy. Usually, the goal is to form an image, i.e., a distribution of light that "resembles" some object. A necessary part of this process is determining how closely this goal has been achieved. Alternatively, you may have assembled two (or more) different systems and be forced to answer the question: which is better, system A or system B (or C, etc.). Obviously, we need some way to quantitatively assess the optical performance of a lens.

It is unrealistic to be forced to formulate a different figure of merit for every conceivable object of which one might wish to form an image. Fortunately, we can think of an arbitrary object as a collection of simple objects: points, lines, edges, etc. The image evaluation techniques discussed in this chapter are all based on looking at the image of one of these "primitive" objects. These objects are simple enough to allow us to analyze their imagery in great detail and also provide insight into the imaging of more complex objects.

Geometrical vs. diffraction evaluation

In general, we consider light as a *vector* wave phenomenon, which implies looking for solutions to Maxwell's equations. This is not necessary for most optical design problems. If we ignore the vector nature of the field, we reduce the problem to a *scalar* wave problem, which implies looking for solutions to the Helmholtz equation, i.e., the scalar field $u(x, y, z)$ that satisfies

$$\nabla^2 u(x,y,z) + k^2 u(x,y,z) = 0 \qquad (7.1)$$

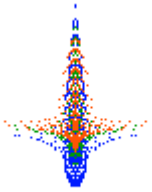
where $k = 2\pi/\lambda$ and λ is the wavelength. If we further ignore the wave-like nature of light, we reduce the problem to a *ray* phenomenon, whose behavior is described by Fermat's principle, which states that the optical path length along a physically possible ray between two points is a stationary value. In this chapter, we will generally consider the propagation of light to be accurately described by rays, but we will utilize many principles of scalar wave theory.

When we perform some image evaluation based solely on the basis of light rays, we call this a *geometrical* evaluation. Another geometrical quantity is the *geometric wavefront*, i.e., a locus of constant optical path from a single point. Keep in mind that even though we are considering a wave, this wavefront is constructed according the laws of geometrical (ray) optics. We will often use results from scalar wave theory with the assumption that the incident field is well approximated by this geometrical wavefront. In this case, we are performing a *diffraction* evaluation. Diffraction effects impose the limits on the performance of an optical system. Geometrically, it is possible that all of the rays from an object point may converge to a single image point, but wave theory shows that the image will be of a finite size, called the *diffraction limit*. For systems with small amounts of aberration, diffraction cannot be ignored.

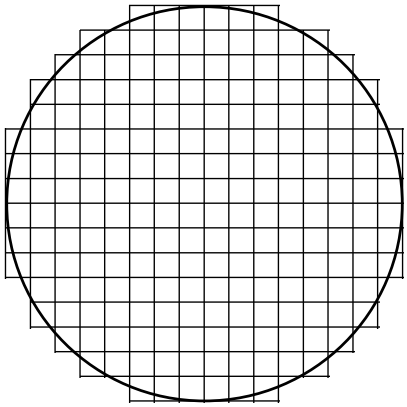
Spot diagrams and wavefronts

A collection of ray data resulting from tracing a large number of rays from a single object point through the aperture of the lens is called a *spot diagram*. The name is a consequence of the appearance of the resulting plot when the ray distributions in the image plane are displayed as "spots" in a graph, as in the figure below. In OSLO, this ray distribution is called a spot diagram even though it is used for much more than just plotting the "spots". For example, since the wavefront is the surface that is orthogonal to the rays at a constant optical path from the source point, we can keep track of the optical path lengths of the rays and construct the wavefront exiting the lens. It is important to note that the spot diagram does not necessarily indicate the distribution of irradiance in the image, as the plot does not show any weighting of the rays. Only in the case

when the pupil is uniformly illuminated, and the rays are uniformly distributed in the pupil, is the ray intersection density in the spot diagram proportional to the geometric irradiance.



In OSLO, the rays that are traced for a spot diagram and wavefront are distributed in a square grid in the entrance pupil. The parameters controlling this grid are set in the Setup spreadsheet accessed using the Setup button in the lens spreadsheet. The number of rays that will be traced is determined by the number of aperture divisions (abbreviated APDIV in OSLO) across the entrance pupil. The grid pattern for APDIV = 16.0 is shown below.



In OSLO, spot diagram rays are aimed such that the above grid corresponds to equal division of the pupil in direction cosine space. This results in an evenly spaced grid in the paraxial entrance pupil (which is a plane) only in the limiting case of an infinitely distant object. Aiming the rays in this way results in a more uniform ray distribution in the exit pupil, particularly for systems that have a large numerical aperture in object space. This so-called aplanatic ray aiming is motivated by the Abbe sine condition, which states that for an infinitely distant object, it is the sine, rather than the tangent, of the exiting slope angle that is proportional to the ray height in the pupil.

Since the exit pupil is an image of the entrance pupil, we would expect that if the ray distribution in the entrance pupil is in the above grid pattern, there should be a similar grid pattern in the exit pupil. But just as the real image is, in general, an aberrated version of the object, the real exit pupil is an aberrated version of the entrance pupil. Thus, the ray distribution in the exit pupil may be somewhat distorted from the regular grid illustrated above. For most systems and applications, the effects of these *pupil aberrations* are negligible. OSLO Premium provides the option of tracing *image space spot diagrams*, for which the rays are traced, iteratively if necessary, such that they form an equally spaced grid in image space direction cosines (i.e., equally incremented on the reference sphere). Image-space spot diagrams can be used for increased accuracy, particularly with calculations that assume equally spaced input data (e.g., Zernike wavefront analysis). However, since this is an iterative process, image space spot diagrams require more computation time.

You can think of the grid cells as sampling the continuous wavefront that actually propagates through the lens. As with any calculation in which a fundamentally continuous function is approximated by a finite number of samples, the accuracy of any resultant quantity depends on whether the sampling is sufficient. Unfortunately, there is no universal way to determine the necessary number of aperture divisions for an arbitrary system. To ensure accuracy it is always a good idea to repeat a calculation with an increased number of divisions until the resulting

calculation does not change substantially (i.e. the numerical process has converged). A simple, but illustrative, example of the effect of APDIV on the accuracy of the resulting calculations is given by the above figure. The real pupil is a circle, but the pupil boundary used for computation is the outer boundary of the grid. As the number of aperture divisions increases, the difference between the circle and the grid boundary becomes smaller.

When a spot diagram is traced, OSLO saves more information than just the ray coordinates on the image surface, which are necessary to plot the spot diagram itself. For example, by saving the differences in the *x* and *y* direction tangents of the rays (measured relative to the reference ray), it is possible to compute spot sizes on focal shifted image surfaces without retracing the rays. Also, by saving the coordinates of each ray's intersection with the reference sphere, the optical path difference (and, hence, the change in the wavefront) introduced by a change of focus can be accurately computed, again without retracing the rays. Saving this data with the spot diagram leads to an increased efficiency when computing quantities such as through-focus *MTF* and focal shifts for minimum spot sizes or wavefront error.

Usually, it is assumed that the light entering the system is of uniform intensity. This is not the case if the system is to be used with a laser. In this case, it is appropriate to use a beam with a Gaussian intensity distribution. In OSLO, this can be done in the Setup spreadsheet by clicking the Gaussian apodization button, and entering the *x* and *y* beam sizes. These are the $1/e^2$ radii, measured at surface 1. It is important to remember that the overall size of the ray bundle is determined by the entrance beam radius; the Gaussian spot sizes just provide an amplitude weighting of the rays. You should make the entrance beam radius at least twice the value of the larger of the *x* and *y* Gaussian spot sizes if you want to simulate an untruncated Gaussian beam.

Spot size analysis

The general concept of spot diagram analysis involves tracing enough rays so that the data for any particular ray can be treated statistically. Each ray is considered to carry a weight proportional to the area of its cell in the aperture of the system and also proportional to the value of the Gaussian apodization function, if any, at the center of the cell. In the study of random variables, a much used concept is that of *moments*, or average values. For example, if a variable *x* has a probability density function of $p(x)$, the first moment, or *centroid*, of *x* is

$$\langle x \rangle = \int_{-\infty}^{\infty} x p(x) dx \tag{7.2}$$

and the second moment is

$$\langle x^2 \rangle = \int_{-\infty}^{\infty} x^2 p(x) dx \tag{7.3}$$

The *variance*, or second central moment, is defined by

$$\sigma^2 = \int_{-\infty}^{\infty} \left(x - \langle x \rangle \right)^2 p(x) dx = \langle x^2 \rangle - \langle x \rangle^2 \tag{7.4}$$

The square root of the variance, σ, is the *standard deviation* and is used to measure the spread of values taken on by *x*. We can use these concepts from statistics to form measures of the ray distributions in the spot diagram. If we have *n* rays in the spot diagram, each with a weight w_i and transverse aberration (DX_i, DY_i), measured relative to the reference ray, then the position of the centroid, relative to the point of intersection of the reference ray with the image surface is

$$\langle x \rangle = \frac{1}{W} \sum_{i=1}^{n} w_i DX_i \qquad \langle y \rangle = \frac{1}{W} \sum_{i=1}^{n} w_i DY_i \tag{7.5}$$

where

$$W = \sum_{i=1}^{n} w_i \tag{7.6}$$

Similarly, the variances are determined by

$$\sigma_x^2 = \frac{1}{W}\sum_{i=1}^{n} w_i \left(DX_i - \langle x \rangle\right)^2 \qquad \sigma_y^2 = \frac{1}{W}\sum_{i=1}^{n} w_i \left(DY_i - \langle y \rangle\right)^2 \tag{7.7}$$

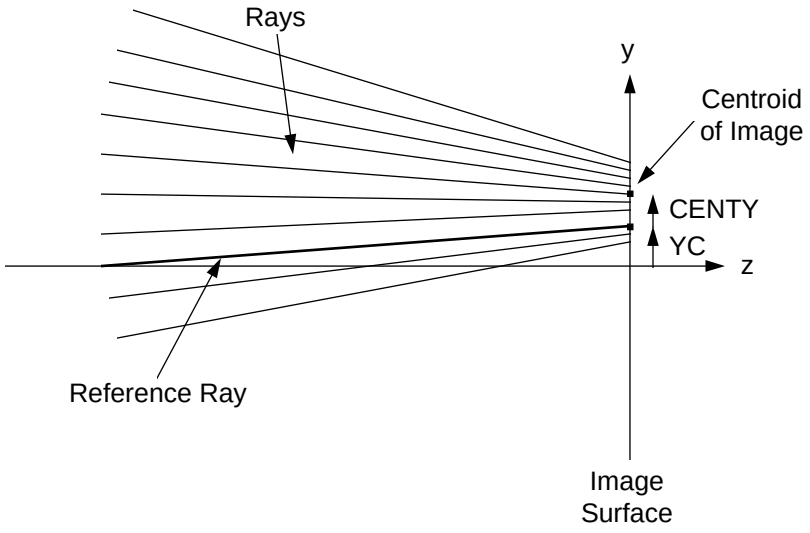
The root-mean-square (RMS) spot sizes in *x* and *y* are the square roots of the above quantities. Finally, the radial RMS spot size σ_r is

$$\sigma_r = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{7.8}$$

The above computations are carried out by OSLO when you request a spot diagram. OSLO also keeps track of the directions on the rays in the spot diagram so that the spot sizes may be computed on a focal shifted image surface, and also so that optimum (in the sense of minimum RMS) focus positions may be found. Also note that by measuring spot sizes relative to the centroid, rather than the reference ray location, any distortion in the image is effectively "subtracted out" from the calculation. For example, the above analysis performed on the spot diagram displayed above, yields the output

```
*SPOT SIZES
   GEO RMS Y   GEO RMS X   GEO RMS R  DIFFR LIMIT     CENTY       CENTX
    0.002907    0.007909    0.008426    0.001296  6.5773e-05       --
```

The standard deviations σ_x, σ_y, and σ_r in the above equations are labeled as GEO RMS X, GEO RMS Y, and GEO RMS R in the OSLO output. The positions of the centroid relative to the reference ray ❶*x*❷ and ❶*y*❷ are labeled CENTX and CENTY. Since this is a rotationally symmetric lens and we have chosen an object point on the *y*-axis, CENTX is identically zero by symmetry, as in the figure below. The DIFFR LIMIT is the radius of the equivalent Airy disk for the system. This value can be compared to the geometric spot sizes as an indication of how close (or far) the performance is to the diffraction limit. If the geometric spot size is much larger than the diffraction limit, we would expect that the performance of the lens will be limited by the geometric aberrations.

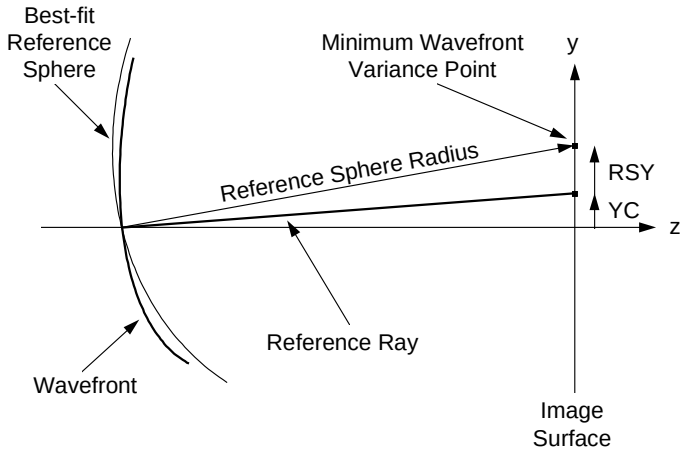


Wavefront analysis

We can also apply the statistical ideas developed in the previous section to the geometric wavefront that is constructed from the rays in the spot diagram. For a geometrically perfect point image, the corresponding wavefront is a sphere, centered on that point. The *optical path difference* (OPD) or *wavefront aberration* is the departure of the actual wavefront from this *reference sphere*.

One subtlety in considering the wavefront rather than the geometric spot distribution is that there is some amount of freedom in choosing the center and radius of the reference sphere. Since an aspheric wavefront changes shape as it propagates, the amount of OPD depends on the location and, hence, the radius, of the reference sphere. For most systems, the OPD is relatively insensitive to where the wavefront aberration is measured, but for some systems (particularly those with a large amount of pupil aberration) the location of the reference sphere is a critical parameter. By default, OSLO locates the reference spherical wavefront in the real exit pupil of the system, although you are free to specify another location, as a General Operating Condition. The exit pupil is located at the image space intersection of the reference ray with a ray that is differentially displaced in the field from the reference ray.

Also, we are usually free to choose the position of the center of the reference sphere, constrained to the image surface. By default, OSLO chooses the reference sphere center location such that the RMS wavefront aberration (i.e., the standard deviation of the OPD) is minimized, as shown in the figure below. For reasonably well corrected systems, the point that minimizes the RMS wavefront error coincides with the peak of the diffraction point image. This point is sometimes called the *diffraction focus*. Most diffraction based calculations in OSLO use this best-fit reference sphere by default. This choice of reference sphere center gives a true indication of the wavefront's departure from sphericity but not necessarily of the fidelity of the image point to its desired location. Thus, distortion must usually be considered separately from the RMS wavefront aberration.



From the same spot diagram rays traced to generate the spot diagram displayed earlier in this Chapter, a wavefront analysis yields

```
*WAVEFRONT RS
 WAVELENGTH 1
   PKVAL OPD     RMS OPD  STREHL RATIO    RSY         RSX         RSZ
    4.267764    1.047296    0.018790   -0.000430      --          --
```

Note that we are usually not concerned with the average value of the OPD, since a constant OPD error (sometimes termed piston error) has no direct effect on image formation in incoherent light. RSY, RSX, and RSZ are the positions of the center of the reference sphere, relative to the reference ray intersection with the image surface. For this example, the object point is on the *y*-axis, so RSX is zero, as expected by symmetry considerations. RSZ is also zero, since we are evaluating the wavefront based on the nominal location of the image surface.

For some evaluations, particularly interferometric testing, it is useful to represent the wavefront in the form of polynomials. The most common polynomials used for this purpose are the *Zernike polynomials*. These polynomials form an orthogonal basis set in the polar coordinates ρ and θ over the interior of the unit circle. They have the properties that the average value of each polynomial (other than the constant term) is zero over the unit circle and each term minimizes the RMS wavefront error to the order of that term. Thus, the RMS wavefront error can not be decreased by the addition of lower order Zernike polynomial terms. It should be noted that the orthogonality condition for the Zernike polynomials is only maintained if the domain considered is the entire interior of the unit circle. Zernike fits performed over non-circular data sets will usually not give

an accurate representation of the actual wavefront. On the positive side, knowing the Zernike coefficients allows for a very compact description (only *n* numbers, the coefficients) of a potentially complex shape wavefront. If we denote the i^{th} Zernike polynomial by $Z_i(\rho, \theta)$, then the total wavefront $W(\rho, \theta)$ is given by

$$W(\rho,\theta) = \sum_{i=0}^{n} c_i Z_i(\rho,\theta) \tag{7.9}$$

where c_i are the coefficients of the expansion. As mentioned earlier in this chapter, it is recommended that image space spot diagrams be used for Zernike analysis (OSLO SIX only). The orthonormalization procedure that is performed as part of the Zernike analysis yield more accurate results if the input data points are equally spaced. More information on Zernike polynomials may be found in Chapter 3.

Point spread functions

The diffraction image of a point object is called the *point spread function* or, in the language of linear systems, the *impulse response*. Fortunately we can calculate the point spread function from a knowledge of the geometric wavefront, which is available from the spot diagram rays.

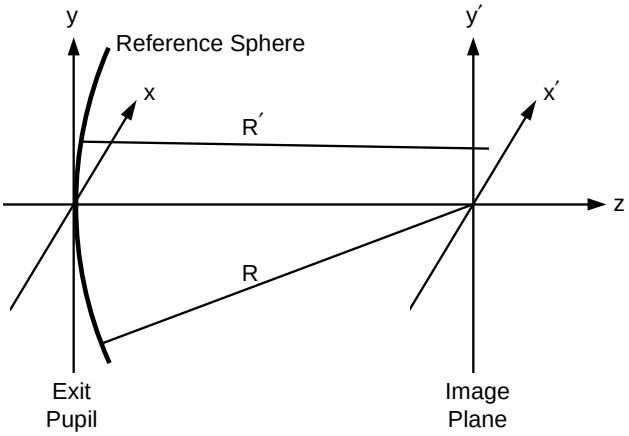
The amplitude distribution in the exit pupil $A(x, y)$ and wavefront aberration $W(x, y)$ can be combined to form the complex pupil function $P(x, y)$:

$$P(x,y) = A(x,y)\exp\left[ikW(x,y)\right] \tag{7.10}$$

where $k = 2\pi/\lambda$, and λ is the wavelength. The pupil function is defined such that $P(x, y) \equiv 0$ outside the pupil. Within the Kirchhoff approximation, the diffracted amplitude $U(x', y')$ is given by

$$U(x',y') = \frac{i}{\lambda}\iint_A P(x,y)\frac{\exp(-ikR')}{R'}dA \tag{7.11}$$

where A is the area of the pupil and R' is the distance from the pupil sphere point (x, y) to the observation point (x', y'). The coordinate system is illustrated below.



For most of cases of interest, Eq. (7.11) is well approximated by

$$U(x',y') = \frac{i\exp\left\{-ik\left[R+\varepsilon(x',y')\right]\right\}}{\lambda M_R R}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x,y)\exp\left[i\frac{2\pi}{\lambda R}(xx'+yy')\right]dxdy \tag{7.12}$$

where R is the radius of the reference sphere, $\varepsilon(x', y')$ is a quadratic phase factor, and M_R is the z-direction cosine of the reference ray. A detailed derivation of Eq. (7.12) may be found in Hopkins and Yzuel(1). Since $P(x, y)$ is zero outside A, the limits on the integrals in the above equation have been written as $-\infty$ to ∞ in order to show more explicitly the form of the integral as a Fourier

transform. (The double integral in Eq. (7.12) is easily recognized as a 2-dimensional Fourier transform with frequency variables $\nu_x = x'/\lambda R$ and $\nu_y = y'/\lambda R$.) In most cases, imaging systems are used with incoherent illumination and we are only interested in the irradiance [squared modulus of $U(x', y')$] in the point image. Thus, the usual definition of the point spread function is in terms of irradiance, i.e.,

$$PSF(x', y') = \left|U(x', y')\right|^2 \tag{7.13}$$

The Fourier transform relationship between the pupil function and the complex amplitude of the *PSF* allows us to make use of the powerful tools of Fourier and linear systems theory. The prime example of this is the widespread use of transfer functions, which are described later in this chapter.

There are two primary techniques for the numerical computation of the integral in Eq. (7.12): direct integration and the Fast Fourier Transform (FFT). There are advantages and disadvantages associated with both of these methods. Direct integration allows for the computation of the *PSF* anywhere in image space, but only one point at a time. The FFT is generally a much faster way to compute Fourier kernel integrals than point-by-point methods, but the increased speed comes at the expense of flexibility in choosing the sampling points in the image plane. The nature of the FFT algorithm fixes the sampling interval in the image plane once the pupil sampling (i.e., the number of rays traced across the pupil diameter) and the number of points in the FFT (N) are chosen. Also, since the FFT computes the entire $N \times N$ transform in a single operation, this is an inefficient way to compute the *PSF* value at a single point, as is necessary, for example, when computing the Strehl ratio. On the other hand, for applications such as displaying a perspective plot of the *PSF* or computing an energy distribution, it is necessary that the *PSF* be known over a grid of points in the image plane and the loss of sampling freedom is often offset by the decrease in computation time. OSLO uses both of these methods, depending on the application.

When using an FFT, it is important to understand how the sampling in the pupil and the image are related. (For simplicity and clarity, we will restrict the discussion here to one dimension. Similar arguments apply for the second dimension.) As stated above, the integral in Eq. (7.12) is a Fourier transform, where the spatial frequency variable is $\nu_y = y'/\lambda R$. Let N be the number of points (in one-dimension) in the array used for the FFT calculation. (The FFT algorithm requires that N be a power of 2.) Then, the relationship between the sampling intervals in the spatial and spatial frequency domains is given by

$$\Delta\nu_y = \frac{1}{N\,\Delta y} \tag{7.14}$$

Thus, given a sampling interval in the pupil of Δy, the sampling interval in the image is

$$\Delta y' = \frac{\lambda R}{N\,\Delta y} \tag{7.15}$$

If we have traced M rays (where $M < N$) across a pupil of diameter D, then $\Delta y = D/M$, so

$$\Delta y' = \frac{\lambda RM}{ND} = \frac{\lambda_0}{2\,NA}\frac{M}{N} \tag{7.16}$$

where λ_0 is the vacuum wavelength and NA is the numerical aperture ($NA = n\,(D/2)/R$). We can consider M/N as the "fill factor" of the input array. Equation (7.16) indicates the tradeoff between pupil and image sampling and also how the sampling interval in the image is fixed, once M and N are chosen. Also, note that the length of one side of the square image patch represented by the FFT grid is

$$\text{Patch size} = N\,\Delta y' = \frac{\lambda_0}{2\,NA}M \tag{7.17}$$

which is only a function of M. It may appear that values of M close to N are preferable, so that most of pupil array is filled with data, rather than padded with zeros. Since, however, we are dealing with the discrete representation of a continuous function, we need to be aware of the

possibilities of aliasing. The Whittaker-Shannon sampling theorem states that aliasing is eliminated if the sampling rate is at least twice the highest frequency component in the function, ν_{max}. Thus, to avoid aliasing,

$$\Delta y' \le \frac{1}{2\nu_{max}} \tag{7.18}$$

In the transfer function section later in this chapter, it is shown that the cutoff frequency for the *OTF* (which is the Fourier transform of the *PSF*) is $\nu_0 = 2\ NA/\lambda_0$. Thus, the maximum value of $\Delta y'$ that satisfies Eq. (7.18) is $\lambda_0/(4\ NA)$ and the value of M_0 that corresponds to this $\Delta y'$ is such that $(\lambda_0\ M_0)/(2\ NA\ N) = \lambda_0/(4\ NA)$ or $M_0 = N/2$. Values of M much larger than M_0 lead to the increased possibility of aliasing (and decreased resolution in the image plane grid) while values of M much smaller than M_0 may result in an inaccurate representation of the wavefront, due to undersampling. Generally a value of M close to $N/2$ results in acceptable sampling intervals and efficiency.

For a diffraction-limited system, the wavefront aberration is zero, so the pupil function is just the amplitude of the spherical wavefront in the exit pupil. If the pupil is uniformly illuminated, then the *PSF* is just the squared modulus of the Fourier transform of the shape of the exit pupil. For example, for the common case of a circular pupil of radius a, the point spread function is rotationally symmetric (i.e., it is a function only of $r' = (x'^2 + y'^2)^{1/2}$) and can be written as

$$PSF(r') = \left[\frac{2J_1(b)}{b}\right]^2 \tag{7.19}$$

where J_1 is the Bessel function of the first kind of order 1 and

$$b = \frac{2\pi a r'}{\lambda R} = \frac{2\pi}{\lambda_0} NA\ r' \tag{7.20}$$

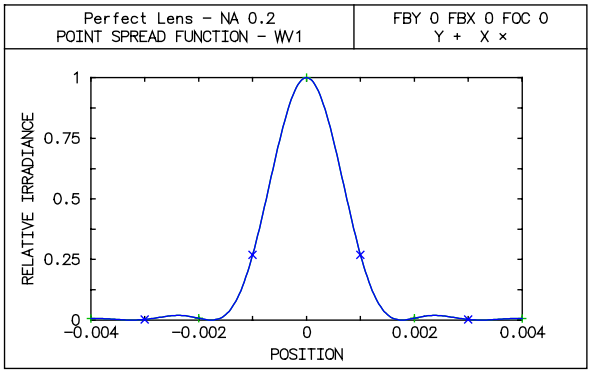
In the above equation, λ is the wavelength in image space, λ_0 is the vacuum wavelength, and NA is the numerical aperture ($NA = n \sin U = n\ a/R$). Equation (7.19) is referred to as the Airy pattern. $J_1(\pi x)$ has its first zero at $x = 1.22$, so the radius of the Airy disk is the value of r' such that $b = 1.22\ \pi$, or

$$r'_{Airy} = \frac{0.61\lambda_0}{NA} \tag{7.21}$$

The above radius is reported as the DIFFR LIMIT as part of the spot size analysis.

We can use the perfect lens surface in OSLO to illustrate the form of the perfect point spread function described by Eq. (7.19). If we set up a perfect lens with a numerical aperture of 0.2, the computed diffraction limit should be close to r'_{Airy} = 0.61*0.00058756 mm/0.2 = 0.001792 mm. The geometric spot sizes are zero, since the lens is geometrically perfect.

```
*SPOT SIZES
   GEO RMS Y    GEO RMS X    GEO RMS R  DIFFR LIMIT     CENTY        CENTX
      --           --           --        0.001792      --           --
```



A widely used figure of merit for well-corrected systems is the *Strehl ratio* (or Strehl definition). This is the ratio of the peak value of the *PSF* to the peak of the *PSF* for an equivalent perfect (unaberrated) system. (For small wavefront aberrations, the Strehl ratio can be shown to be directly related to the variance of the wavefront.) This normalization is used by OSLO; thus the *PSF* values reported by OSLO are relative to the diffraction-limited peak value. Note that other normalizations are possible and sometimes used. For example, it is convenient to normalize the total energy in the *PSF* to 1 when discussing energy distributions (see below).

Line spread functions and knife edge distributions

It is sometimes more instructive to consider the image of an infinitely long line, rather than a point. Since we can consider a line to be a collection of points, the *line spread function* (*LSF*), in x say, is the summation of an infinite number of points oriented in the y-direction:

$$LSF(x') = \int_{-\infty}^{\infty} PSF(x', y')\, dy' \tag{7.22}$$

If we consider scanning an infinitely long, straight "knife edge" across the line spread function, and plot the fractional energy uncovered by the knife edge as we scan from minus infinity to plus infinity, we obtain the image of an infinite edge or the *knife edge distribution* (*KED*):

$$KED(x') = \int_{-\infty}^{x'} LSF(\tilde{x})\, d\tilde{x} \tag{7.23}$$

We can calculate line spread functions and knife edge distributions geometrically by using the spot diagram in place of the irradiance point spread function.

Fiber coupling

One application of the point spread function is the computation of fiber coupling efficiency. This calculation involves an optical system that is designed to collect light from some source (a laser, another fiber, etc.) and couple it into a receiving fiber. The structures of optical fibers support *guided modes* of propagation, whose energy is mainly confined to a region near the axis of the fiber.

Given an amplitude diffraction pattern (amplitude of the point spread function) $U(x', y')$ and a fiber mode pattern $\psi(x', y')$, the coupling efficiency η is defined as the normalized overlap integral

$$\eta = \frac{\iint U(x', y')\psi^*(x', y')\, dx'dy'}{\sqrt{\iint U(x', y')U^*(x', y')\, dx'dy' \iint \psi(x', y')\psi^*(x', y')\, dx'dy'}} \tag{7.24}$$

where the asterisk denotes the complex conjugate. The power coupling efficiency T is given by

$$T = \eta\eta^* = |\eta|^2 \tag{7.25}$$

T is the fraction of the power in the incident field that is coupled into the mode of the fiber defined by $\psi(x', y')$. Since the numerical evaluation of Eq. (7.24) requires that the diffracted amplitude be known over a grid of points (x', y') in the image plane, OSLO uses FFT diffraction calculations to compute $U(x', y')$. Note that it follows from Schwarz's inequality that $0 \le T \le 1$ and that $T = 1$ if and only if $\psi(x', y') = KU(x', y')$, where K is a complex constant. This makes intuitive sense, as we would expect to couple all of the incident light into the fiber only if the field and the mode overlapped completely.

The form of the mode pattern $\psi(x', y')$ depends upon the structure of the fiber, of which the most common types are gradient index and step index. Most single-mode, gradient-index fibers have a fundamental mode that is well described by a Gaussian function of the form

$$\psi_{Gaussian}(x', y') = \exp\left[-\left(\frac{x'^2 + y'^2}{r_0^2}\right)\right] = \exp\left[-\left(\frac{r'}{r_0}\right)^2\right] \tag{7.26}$$

Thus, the Gaussian mode is completely specified by the radius r_0 at which the mode amplitude drops to $1/e$ of its axial value.

A step-index fiber consists of a central core of homogeneous material surrounded by a cladding material of slightly lower refractive index. This type of fiber requires three parameters for its specification: the refractive index of the core material n_{core}, the refractive index of the cladding material $n_{cladding}$, and the radius of the cylindrical core a. Generally, this type of fiber supports many propagating modes, but we are usually interested in using the fundamental mode. For the usual case of a weakly guiding fiber, i.e., $(n_{core} - n_{cladding})/n_{cladding} << 1$, the fundamental mode is given by

$$\psi_{step-index}(r') = \begin{cases} \dfrac{J_0\left(\dfrac{ur'}{a}\right)}{J_0(u)}, & r' \le a \\ \dfrac{K_0\left(\dfrac{wr'}{a}\right)}{K_0(w)}, & r' > a \end{cases} \tag{7.27}$$

where $r' = (x'^2 + y'^2)^{1/2}$ and u and w are constants determined by the fiber construction parameters. J_0 is the Bessel function of order 0, and K_0 is the modified Hankel function of order 0. The parameters u and w are a function of the "normalized frequency" v:

$$v = \frac{2\pi}{\lambda_0} a \sqrt{n_{core}^2 - n_{cladding}^2} \tag{7.28}$$

The explicit forms for u and w for the fundamental mode are

$$u = \frac{\left(1 + \sqrt{2}\right)v}{1 + \left(4 + v^4\right)^{1/4}} \tag{7.29}$$

and

$$w = \sqrt{v^2 - u^2} \tag{7.30}$$

The derivation of this mode structure can be found in, for example, Gloge(2). If the desired fiber mode is neither Gaussian nor this fundamental step-index mode, OSLO Premium allows for the specification of an arbitrary mode structure via the use of a CCL command.

For a general discussion of the calculation of fiber coupling efficiency, the interested reader is referred to Wagner and Tomlinson(3).

Energy distribution

The spread functions defined in the previous section give a complete description of the distribution of irradiance in the image of a point, line, or edge. In some cases, it is more useful to know how much of the total energy in the point image is contained within a circle or square of a given size. For example, we may define the radius of the circle that contains, say, 80% of the total energy as the "spot size." Or, we may be using a detector array with square pixels and we need to ensure that a certain fraction of the energy is contained within the size of one pixel so that we can accurately determine the position of the spot on the array. These types of calculations require the computation of *energy distributions*. As usual, we can base the calculation on either a geometric (using the spot diagram) or diffraction (using the point spread function) basis.

We can consider the point spread function to be a function of two image plane coordinates: Cartesian coordinates x' and y' or polar coordinates r' and θ'. For simplicity, we will assume that a normalization has been chosen such that the total energy in the *PSF* is 1.0, i.e.,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} PSF(x',y')\,dx'dy' = \int_{0}^{2\pi}\int_{0}^{\infty} PSF(r',\theta')\,r'dr'd\theta' = 1 \tag{7.31}$$

When computing integrals numerically, which is, of course, what a computer does, there are some subtleties raised when calculating the normalization factor of the above equation. Since the pupil for any real system must be a finite size, the Fourier transform relationship [Eq. (7.12)] implies that the *PSF* extends over the entire (infinitely large) image surface. For example, the perfect point spread function of Eq. (7.19) tends to zero as r' tends to infinity, but there is no finite value of r' outside of which the *PSF* is zero. Practically speaking, of course, the *PSF* can be considered to be negligibly small for large enough values of r'. When computing Eq. (7.31) numerically, OSLO assumes that the finite integration patch on the image surface contains all of the energy in the *PSF*, i.e., the *PSF* is zero outside this patch. Conservation of energy implies that the total energy in the *PSF* must equal the total energy in the pupil function. These considerations make the FFT technique more attractive than direct integration when computing *PSF*s for energy distributions. Parseval's theorem for the discrete Fourier transform ensures that energy is conserved when using the FFT. The value of Eq. (7.31) computed by direct integration is much more sensitive to image surface sampling interval and image patch size. Thus, OSLO uses the FFT algorithm when computing diffraction energy distributions.

Then, the *radial energy distribution* (*RED*) or *encircled energy* for a circle of radius *a* is given by

$$RED(a) = \int_{0}^{a}\int_{0}^{2\pi} PSF(r',\theta')\,r'dr'd\theta' \tag{7.32}$$

and the *ensquared energy* (*SQE*) for a square of side length *s* is

$$SQE(s) = \int_{-s/2}^{s/2}\int_{-s/2}^{s/2} PSF(x',y')\,dx'dy' \tag{7.33}$$

Obviously the form of either energy distribution depends on the choice of origin. In OSLO, the distributions are centered at the centroid of the *PSF* (diffraction) or spot diagram (geometric).

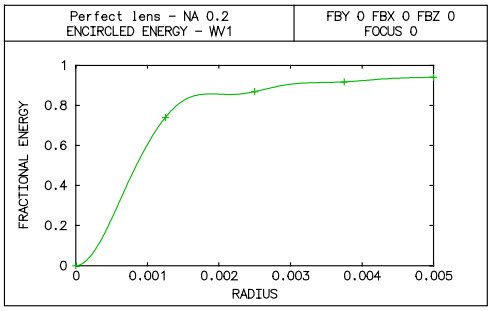
We will compare the radial energy distributions for perfect point spread functions for uniform and Gaussian pupils, using the same perfect lens as in the point spread function example above.

For the perfect point spread function given by Eq. (7.19), Lord Rayleigh derived an analytic form for the encircled energy:

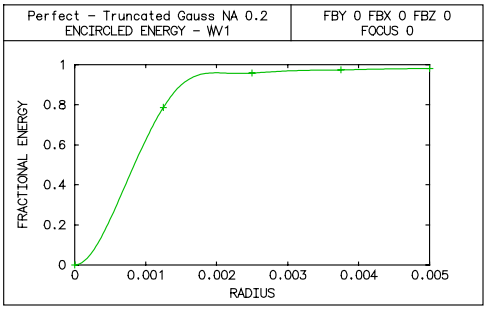
$$RED(a) = 1 - J_0^2\left(\frac{2\pi}{\lambda_0} NA\, a\right) - J_1^2\left(\frac{2\pi}{\lambda_0} NA\, a\right) \tag{7.34}$$

From this formula, we can compute that approximately 84% of the total energy is contained in the Airy disk and 91% of the total energy is within a radius equal to the second dark ring of the Airy pattern.

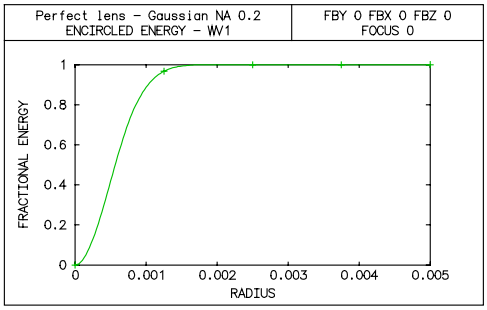
We want to plot the radial energy distribution and compare it to Eq. (7.34). The third zero of $J_1(\pi x)$ occurs at $x = 3.238$ or $r' = 1.619\ \lambda_0/NA = 1.619*0.00058756$ mm/0.2 = 0.0048 mm. This suggests that a maximum radius of 5 μm would include the most interesting parts of the curve. In the radial energy distribution plot, the places where the curve is flat correspond to the dark rings in the pattern, since there is no energy at that radius. From the plot we see that these flat portions of the curve occur at the expected radii of 0.0018 mm, 0.0033 mm, and 0.0048 mm.



Now we will compute the energy distribution for a Gaussian apodized pupil. We will set up the beam so that the *NA* of the $1/e^2$ irradiance point is 0.2. In the Setup spreadsheet, we turn on the use of the Gaussian apodized pupil, and enter 20 for the entrance Gaussian spot sizes in *x* and *y*. We can now plot the radial energy distribution and compare it to the same plot for the uniformly illuminated pupil.



In order to avoid truncating the incoming beam, so we need to increase the entrance beam radius. If we make the EBR twice as large as the $1/e^2$ radius, this should result in a negligible input truncation. Changing the entrance beam radius to 40 mm produces the following result.



Since the Gaussian *PSF* does not have the lobes of the Airy pattern, the Gaussian encircled energy is always increasing (i.e., no "flat" spots). In fact, it is easy to show that for an irradiance distribution of the form $PSF(r') = 2\exp[-2(r'/w)^2]/(\pi w^2)$, where *w* is a constant, the encircled energy is given by $RED(a) = 1 - \exp[-2(a/w)^2]$.

Transfer functions

Closely related to the point spread function is its Fourier transform, the *optical transfer function* (*OTF*). The *OTF* is a measure of the accuracy with which different frequency components in the object are reproduced in the image. There is a vast literature on transfer function theory; we are only able to provide a brief outline here. Keep in mind that. strictly speaking, transfer function concepts are only applicable to incoherent illumination. Different techniques must be used for systems operating in fully coherent or partially coherent illumination.

As stated above the *OTF* is the (normalized) Fourier transform of the *PSF* and, as such, is a function of the spatial frequencies ν_x and ν_y.

$$OTF(\nu_x,\nu_y)=\frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}PSF(x',y')\exp\left[i2\pi(\nu_x x'+\nu_y y')\right]dx'dy'}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}PSF(x',y')dx'dy'} \tag{7.35}$$

The normalization is such that $OTF(0, 0) = 1$. Because of the additional Fourier transform relationship between the complex pupil function and the *PSF* [See Eq. (7.12) above], the *OTF* can also be expressed as the autocorrelation of the pupil function. This is, in fact, the way that the *OTF* is computed in OSLO. The *OTF* includes the effects both of geometric aberrations and diffraction. If we consider the diffraction-limited *PSF* given by Eq. (7.19), we can compute the corresponding *OTF* by taking the Fourier transform. We find that the *OTF* is identically zero for spatial frequencies larger than a certain value, the cutoff frequency ν_0 given by

$$\nu_0=\frac{2\,NA}{\lambda_0} \tag{7.36}$$

Just as diffraction limits us from forming image points smaller than a certain size, it also sets an upper limit on the maximum spatial frequency that can be present in an image.

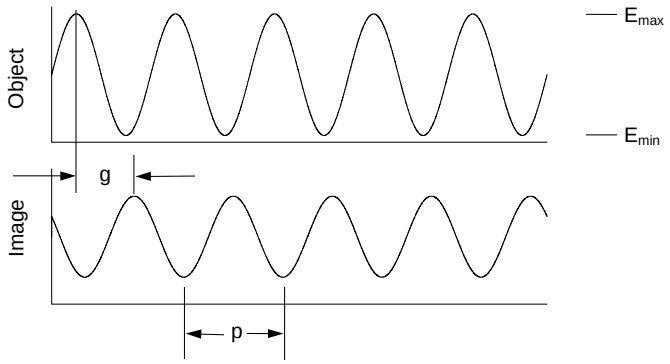
It is obvious from the above definition that the *OTF* is a complex function. The modulus of the *OTF* is called the *modulation transfer function* (*MTF*) and the phase of the *OTF* is the *phase transfer function* (*PTF*). The *MTF* is the ratio of the modulation in the image to the modulation in the object. The *PTF* is a measure of the shift of that spatial frequency component from its ideal position. Referring to the figure below, which is a schematic illustration of a sinusoidal irradiance object and its image, the modulation *M* is defined by

$$M=\frac{E_{max}-E_{min}}{E_{max}+E_{min}} \tag{7.37}$$

and the *MTF* and *PTF* are

$$MTF=\frac{M_{image}}{M_{object}} \tag{7.38}$$

$$PTF=2\pi\frac{g}{p}=2\pi g\nu \tag{7.39}$$

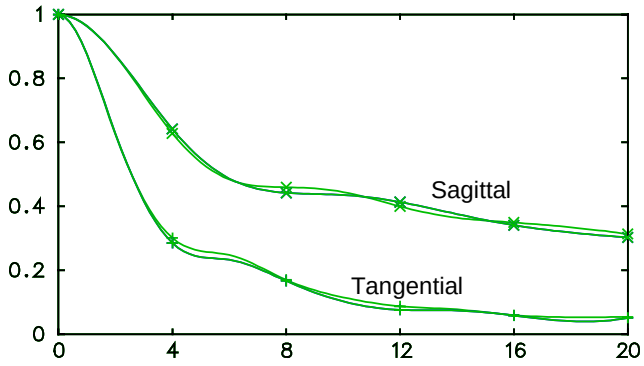


The *OTF* is independent of the precise nature of the object. This is what makes transfer functions useful: they can be computed from the wavefront of the lens and used to calculate image quality, independent of the specific object. If we consider a general object irradiance distribution $o(x, y)$ with Fourier spectrum $O(\nu_x, \nu_y)$ then (assuming proper normalization for magnification) the Fourier spectrum of the image $I(\nu_x, \nu_y)$ is given by the product

$$I(\nu_x,\nu_y)=OTF(\nu_x,\nu_y)\,O(\nu_x,\nu_y) \tag{7.40}$$

Even though most objects are not sine wave targets, the *OTF* is still useful because Fourier analysis allows us to represent an arbitrary object as a superposition of various spatial frequency components.

It is possible to calculate an *OTF* using only geometrical optics considerations. In this case, the *PSF* in Eq. (7.35) is replaced by the spot diagram. Obviously, this approximation should only be expected to give an accurate representation of the *OTF* when the aberrations are large (greater than a few wavelengths). It turns out that the convolution method for computing diffraction MTF eliminates the need for geometrical MTF calculations; the convolution method produces accurate answers, comparable to those obtained from geometrical calculations, even when the system has many waves of aberration. The figure below shows a superposition of two plots, one geometrical and one diffraction (based on the convolution MTF), for a system having approximately 20 waves of OPD. The results for both tangential and sagittal plots are within the expected accuracy range.



Since the MTF is the Fourier transform of the pupil function, it is also possible to compute the MTF using FFT techniques. However, in order to obtain sufficient accuracy with systems that are not diffraction limited, it is necessary to use very fine sampling, and the Fast Fourier Transform turns rapidly into a Slow Fourier Transform as the aberration increases. OSLO does provide a FFT calculation of MTF, because the fact that the entire transform is computed at once makes it easy to plot in two dimensions. However, the sampling limitations of FFT make this method unsatisfactory for general use.

Partial coherence

As part of the process of optical design, we consider the propagation of light from the object to the image. Usually, we are less concerned with the source of the incident light. It is clear, however, that a complete analysis of the imaging properties of a lens must take the nature of the incident illumination into consideration. Take, as an example, the difference between images formed in sunlight and those formed with laser illumination. The laser-illuminated images have a grainy texture (speckle) that is not found in the natural light images. In this chapter, we will examine the imaging process for a class of systems sometimes called *projectors*, i.e., optical systems that both illuminate an object and form an image of it.

Coherence functions

When including the effects of illumination in optical imagery, we are led into the field of *optical coherence*, which is the analysis of light propagation from a statistical point of view. The fact that a careful analysis requires the use of statistics should not be a surprise to anyone familiar with quantum mechanics. For example, most thermal light sources operate by exciting a group of atoms or molecules, which then emit light by spontaneous emission. This spontaneous emission, which occurs during a transition from the excited state to a lower energy state, occurs randomly and independently from each atom or molecule. The resulting light from this source consists of the sum of all of these independent contributions. Fortunately, we shall only need a few concepts from classical coherence theory to study partially coherent imagery. Coherence theory is a very deep

and well-developed branch of optics; we shall only be able to give a brief introduction here. The interested reader is referred to the literature for more detailed information.

It is common to consider interference a manifestation of coherence; the visibility of the interference fringes in Young's double-slit experiment is directly related to the coherence of the incident light. A naive explanation may be that coherent light interferes and incoherent light does not. This view is, however, inadequate. As first demonstrated by Verdet in the 1860's, one can observe interference fringes using a source we commonly think of as incoherent, namely the sun.

Since we are dealing with fundamentally random processes, it is convenient to work with correlation functions or averages. In particular, we will use second-order correlations, since these are directly related to the observable quantity of irradiance. Let $U(\mathbf{x};\ \nu)$ denote the complex amplitude of a particular monochromatic (of frequency ν) component of the optical field at the point $\mathbf{x}$. (The vector $\mathbf{x}$ denotes the point (x, y, z).) The amplitude U is a random variable. The *cross-spectral density function* W is the ensemble-averaged correlation function of U at the point $\mathbf{x}_1$ with the complex conjugate of U at another point $\mathbf{x}_2$. Denoting the ensemble average by angle brackets and the complex conjugate by an asterisk, we can write the cross-spectral density as

$$W(\mathbf{x}_1, \mathbf{x}_2; \nu) = \left\langle U(\mathbf{x}_1; \nu) U^*(\mathbf{x}_2; \nu) \right\rangle \tag{7.41}$$

(It is assumed that the random process associated with U is ergodic, so that time averages and ensemble averages are equal.)

It is often convenient to work with a normalized correlation coefficient, called, in this case, the *complex degree of spectral coherence* $\mu_{12}(\nu)$.

$$\mu_{12}(\nu) = \mu(\mathbf{x}_1, \mathbf{x}_2; \nu) = \frac{W(\mathbf{x}_1, \mathbf{x}_2; \nu)}{\sqrt{W(\mathbf{x}_1, \mathbf{x}_1; \nu) W(\mathbf{x}_2, \mathbf{x}_2; \nu)}} \tag{7.42}$$

It can be shown that

$$0 \le \left| \mu_{12}(\nu) \right| \le 1 \tag{7.43}$$

If the magnitude of the complex degree of spectral coherence is unity (i.e., perfect correlation), this indicates that the field is perfectly *coherent* between $\mathbf{x}_1$ and $\mathbf{x}_2$. On the other hand, a value of $\mu_{12}(\nu) = 0$ (i.e., no correlation at all) indicates complete *incoherence*. Values greater than 0 and less than 1 are indicative of *partial coherence*. Already we see that the familiar labels of coherent and incoherent are just the limiting cases of a continuum of possible values of the coherence of the light.

In coherence theory, the observable quantity is usually called the *intensity*, although it more properly corresponds to the irradiance of radiometry theory. In the context of partial coherence theory, the intensity is the time-averaged square magnitude of the field. Due to the ergodicity of the process, the intensity I (or *spectral intensity*, since we are really dealing with the spectrum of the light) is defined as the trace of the cross-spectral density function

$$I(\mathbf{x}_1; \nu) = W(\mathbf{x}_1, \mathbf{x}_1; \nu) = \left\langle \left| U(\mathbf{x}_1; \nu) \right|^2 \right\rangle \tag{7.44}$$

Different monochromatic components of the field can not interfere since they are mutually incoherent. Thus the intensity of the total field I_{total} is the sum of the intensities of the monochromatic components.

$$I_{total}(\mathbf{x}_1) = \int_0^\infty I(\mathbf{x}_1; \nu)\, d\nu \tag{7.45}$$

Van Cittert-Zernike theorem

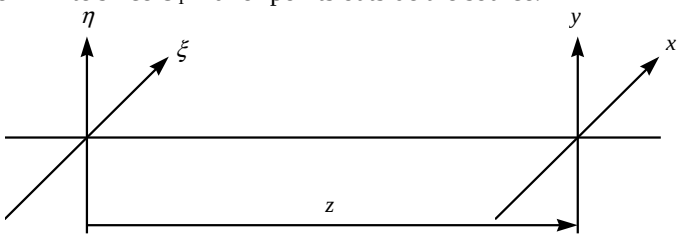
Except for laser-based systems, most optical systems operate with sources that can be considered to be a collection of independently radiating source points. As long as the optical system cannot resolve these independent source points, we can model the source as incoherent. (The sources are

assumed to be independent, so there is no correlation from one source point to another.) Since this situation is so common, it is useful to examine the coherence properties of such a primary source.

For simplicity, consider the case where both the source and observation regions are planes. Let the coordinates in the source plane (plane I) be (ξ, η) and the coordinates in the observation plane (plane II) be (x, y). The two planes are separated by a distance z, as illustrated in the figure below. Using the normal conditions of Fresnel diffraction for an incident coherent field $U_I(\xi, \eta; \nu)$, the diffracted coherent field is given by

$$U_{II}(x, y; \nu) = \frac{-ik \exp(ikz)}{2\pi z} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_I(\xi, \eta; \nu) \exp\left\{ i \frac{k}{2z}\left[(x-\xi)^2 + (y-\eta)^2 \right] \right\} d\xi d\eta \tag{7.46}$$

where $k = 2\pi/\lambda = 2\pi\nu/c$. λ is the wavelength of the radiation and c is the speed of light. The limits on the integrals are infinite since $U_I \equiv 0$ for points outside the source.



We can use Eq. (7.46) and the definition of the cross-spectral density [Eq. (7.41)] to write the cross-spectral density in the observation plane as

$$\begin{aligned} W_{II}(\mathbf{x}_1, \mathbf{x}_2; \nu) &= \left\langle U_{II}(\mathbf{x}_1; \nu) U_{II}^*(\mathbf{x}_2; \nu) \right\rangle \\ &= \frac{1}{(\lambda z)^2} \iint\iint d\xi_1 d\eta_1 d\xi_2 d\eta_2 \left\langle U_I(\xi_1, \eta_1; \nu) U_{II}^*(\xi_2, \eta_2; \nu) \right\rangle \\ &\quad \exp\left\{ i \frac{k}{2z}\left[(x_1-\xi_1)^2 + (y_1-\eta_1)^2 - (x_2-\xi_2)^2 - (y_2-\eta_2)^2 \right] \right\} \\ &= \frac{1}{(\lambda z)^2} \iint\iint d\xi_1 d\eta_1 d\xi_2 d\eta_2 \, W_I(\xi_1, \eta_1; \xi_2, \eta_2; \nu) \\ &\quad \exp\left\{ i \frac{k}{2z}\left[(x_1-\xi_1)^2 + (y_1-\eta_1)^2 - (x_2-\xi_2)^2 - (y_2-\eta_2)^2 \right] \right\} \end{aligned} \tag{7.47}$$

For the case we are interested in, i.e., an incoherent source, the cross-spectral density in plane I can be expressed as

$$W_I(\xi_1, \eta_1; \xi_2, \eta_2; \nu) = \kappa I_I(\xi_1, \eta_1; \nu) \delta(\xi_1 - \xi_2) \delta(\eta_1 - \eta_2) \tag{7.48}$$

where κ is a constant, I_I is the source intensity and $\delta(x)$ is the Dirac delta function. Substituting Eq. (7.48) into Eq. (7.47) yields

$$W_{II}(\mathbf{x}_1, \mathbf{x}_2; \nu) = \frac{\kappa \exp(i\psi)}{(\lambda z)^2} \iint I_I(\xi, \eta; \nu) \exp\left[-i \frac{2\pi}{\lambda z} (\Delta x \xi + \Delta y \eta) \right] d\xi d\eta \tag{7.49}$$

where

$$\psi = \frac{\pi}{\lambda z}\left(\rho_1^2 - \rho_2^2\right) \qquad (7.50)$$

$$\rho_1^2 = x_1^2 + y_1^2 \qquad \rho_2^2 = x_2^2 + y_2^2$$

and

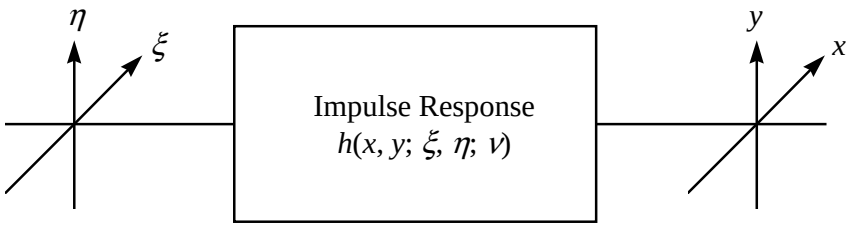
$$\Delta x = x_1 - x_2 \qquad \Delta y = y_1 - y_2 \qquad (7.51)$$

Equation (7.49) is one form of the *Van Cittert-Zernike theorem*. It states that the cross-spectral density is related to the Fourier transform of the intensity distribution of an incoherent source. Note the functional similarity between Eq. (7.49) and the familiar Fraunhofer diffraction formula; the quantities involved in the two cases are, however, quite different. The Van Cittert-Zernike theorem tells us that there is some non-zero correlation length on the (x, y) plane, even if the primary source is incoherent. Thus, we should expect to see interference fringes from Young's experiment, even with an extended incoherent source (e.g., the sun), if the cross-spectral density, as given by Eq. (7.49) is large enough.

The Fourier transform relationship between the cross-spectral density in plane II (which we call a *secondary source*) and the primary source intensity means that the "sizes" of these two functions are, roughly speaking, inversely related. For an infinitesimal incoherent source (a point source), the cross-spectral density is a constant. This makes sense, since a point source is perfectly coherent with itself, and we would expect the resulting interference fringes to have perfect visibility. As the incoherent source becomes larger, the "size" of the cross-spectral density decreases, approaching a delta function as I_I becomes infinitely large. Each point on the extended source produces interference fringes of perfect visibility, but each fringe pattern has a different phase, so the contrast of the resulting overall pattern is reduced.

Partial coherence in a linear system

The familiar treatment of modulation transfer functions and point spread functions derives from the analysis of optical systems within the framework of *linear systems*. The key descriptor of a linear system is called the *impulse response*, which describes the output of the system to an impulse (delta function) input. Let the input plane (plane I) of the optical system have coordinates (ξ, η) and the output plane (plane II) have coordinates (x, y). The impulse response is a function of the output coordinates, the input coordinates, and the frequency ν, and will be denoted by $h(x, y; \xi, \eta; \nu)$. This relationship is illustrated schematically in the figure below.



In optics, the impulse response is the complex amplitude of the point spread function. For the general linear system, the output spectral amplitude in plane II is related to the input spectral amplitude in plane I via

$$U_{II}(x, y; \nu) = \iint U_I(\xi, \eta; \nu)\, h(x, y; \xi, \eta; \nu)\, d\xi d\eta \qquad (7.52)$$

Using Eq. (7.52) and the definition of the cross-spectral density, we find that the cross spectral density in the output plane is

$$W_{\mathrm{II}}(x_1, y_1; x_2, y_2; \nu) = \iint\iint d\xi_1 d\eta_1 d\xi_2 d\eta_2 W_{\mathrm{I}}(\xi_1, \eta_1; \xi_2, \eta_2; \nu) \\ h(x_1, y_1; \xi_1, \eta_1; \nu) h^*(x_2, y_2; \xi_2, \eta_2; \nu) \tag{7.53}$$

If we denote the incident cross-spectral density by W_{inc} and the amplitude transmittance of the object in plane I by t, W_{I} is given as

$$W_{\mathrm{I}}(\xi_1, \eta_1; \xi_2, \eta_2; \nu) = t(\xi_1, \eta_1; \nu) t^*(\xi_2, \eta_2; \nu) W_{\mathrm{inc}}(\xi_1, \eta_1; \xi_2, \eta_2; \nu) \tag{7.54}$$

The spectral intensity in plane II, I_{II}, is computed from the resulting cross-spectral density in the usual way

$$I_{\mathrm{II}}(x, y; \nu) = W_{\mathrm{II}}(x, y; x, y; \nu) \tag{7.55}$$

Starting from an incoherent source, W_{inc} can be found by using the Van-Cittert-Zernike theorem. This results in Eq. (7.53) taking the form of a rather formidable looking six-dimensional integral. Fortunately, we can make some simplifications. We assume that the impulse response is *space invariant* (otherwise known as *stationary* or *isoplanatic*). This means that h is only a function of the coordinate differences $x - \xi$ and $y - \eta$. Practically speaking, this means that the optical aberrations are effectively constant over the image region of interest (called the *isoplanatic patch*). Also, the illumination is chosen to eliminate space-variant phase factors, such as ψ in Eq. (7.49). Such illumination, the primary example of which is Köhler illumination, is sometimes called *matched illumination*. If all of these conditions are true, the image spectral intensity is given by

$$I_{\mathrm{II}}(x, y; \nu) = \iint\iint d\xi_1 d\eta_1 d\xi_2 d\eta_2 W_{\mathrm{inc}}(\xi_1 - \xi_2, \eta_1 - \eta_2; \nu) \\ h(x - \xi_1, y - \eta_1; \nu) h^*(x - \xi_2, y - \eta_2; \nu) \\ t(\xi_1, \eta_1; \nu) t^*(\xi_2, \eta_2; \nu) \tag{7.56}$$

Note that the right-hand-side of Eq. (7.56) is grouped such that the first two lines contain terms that are related to the projector (illumination and imaging optics), while the third line is the object-dependent term. The integrals in Eq. (7.56) are convolution-like, and the imaging equations can also be expressed in the Fourier (spatial frequency) domain. The resulting expressions are still rather complicated, but suffice it to say that the transition to Fourier space results in a great increase in computational efficiency. For more of the mathematical details, the interested reader is referred to the pioneering work of Hopkins[4] and the textbook by Goodman.[5]

The general procedure then, for computing the spectral intensity is to start with an incoherent source, use the Van Cittert-Zernike theorem to compute the cross-spectral density incident upon the object, use Eq. (7.54) to form the input cross-spectral density to the imaging system, use Eq. (7.53) (or, usually, its Fourier domain equivalent) to find the output cross-spectral density, and, finally, use Eq. (7.55) to find the spectral intensity, i.e., the output irradiance distribution.

It is instructive to look at two limiting cases of the general result of Eq. (7.53). For the case of completely coherent incident illumination, W_{inc} has the form

$$W_{\mathrm{inc}}(\xi_1, \eta_1; \xi_2, \eta_2; \nu) = \kappa_{12}(\nu) U_{\mathrm{inc}}(\xi_1, \eta_1; \nu) U_{\mathrm{inc}}^*(\xi_2, \eta_2; \nu) \tag{7.57}$$

where κ_{12} is a complex quantity. The cross-spectral density integral then separates into integrals over (ξ_1, η_1) and (ξ_2, η_2). The resulting spectral intensity is

$$I_{\mathrm{II}}(x, y; \nu) = \left| \kappa_{12}(\nu) \iint t(\xi, \eta; \nu) U_{\mathrm{inc}}(\xi, \eta; \nu) h(x - \xi, y - \eta; \nu) d\xi d\eta \right|^2 \tag{7.58}$$

The convolution in Eq. (7.58) can be evaluated, using Fourier theory, as the Fourier inverse of the product of the transforms of the individual functions. The Fourier transform of h is called the *coherent transfer function*. Note that for coherent illumination, the system is linear in the optical field, not irradiance.

At the other extreme, for the case of completely incoherent incident illumination, W_{inc} has the form

$$W_{\text{inc}}(\xi_1,\eta_1;\xi_2,\eta_2;\nu)=\kappa I_{\text{inc}}(\xi_1,\eta_1;\nu)\delta(\xi_1-\xi_2)\delta(\eta_1-\eta_2) \tag{7.59}$$

where κ is a real constant and I_{inc} is the incident spectral intensity. In this case, the output spectral intensity is

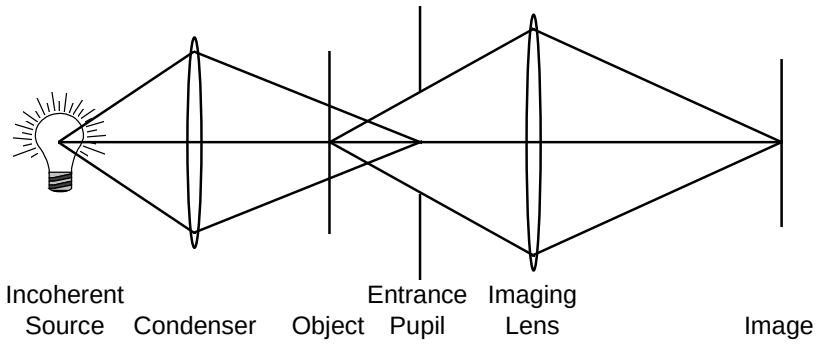
$$I_{\text{II}}(x,y;\nu)=\kappa\iint\left|t(\xi,\eta;\nu)\right|^2 I_{\text{inc}}(\xi,\eta;\nu)\left|h(x-\xi,y-\eta;\nu)\right|^2 d\xi d\eta \tag{7.60}$$

Once again, the resulting integral has the form of a convolution. Now the incident intensity is convolved with $|h|^2$, the incoherent impulse response, i.e., the point spread function. The Fourier transform of $|h|^2$ is the optical transfer function. Note that for incoherent illumination, the system is linear in spectral intensity. Keep in mind that the familiar concept of *MTF* is only applicable in this incoherent limit.

For both coherent and incoherent limits, the required integral has the form of a convolution of an object term with an impulse response term. The appropriate impulse response, either h or $|h|^2$, is independent of the object. Thus, one can properly speak of a transfer function (the Fourier transform of the impulse response) that multiplies the object spectrum to result in the image spectrum. For the case of partially coherent illumination, however, the image spectrum calculation does not take the form of a convolution with the object, so there is no equivalent transfer function. The image irradiance must be explicitly calculated for each object of interest.

Partially coherent imagery

As mentioned in the previous section, we need to consider the propagation of light from an incoherent source to the final image. We will assume that the object is illuminated by matched illumination, so that the imagery is stationary. Illustrated schematically, the complete optical system is shown below. Common examples of systems of this type include microscopes and photolithography systems.



In OSLO, only the imaging optical system is entered; the source and condenser are assumed to exist and their relevant properties are described by the partial coherence operating conditions. The coherence properties of the incident illumination are determined by the geometric image of the primary source in the entrance pupil of the imaging lens. This incoherent image is known as the *effective source* and can be related to the coherence of the illumination incident upon the object by the Van Cittert-Zernike theorem.

The most important property of the effective source is its size, relative to the size of the entrance pupil of the imaging lens. Most systems for which coherence effects are important are systems that produce high quality images. Since the imagery is to be of uniform quality regardless of azimuth, an axis of rotational symmetry is usually desired. Thus, in OSLO, the effective source and imaging lens entrance pupil are assumed to be nominally circular. The key parameter for specifying the effective source is the ratio of the radius of the effective source to the radius of the entrance pupil of the imaging lens. This ratio is commonly denoted by σ. A value of $\sigma = 0$ means that the effective source is a point, and the object is illuminated coherently. As σ increases from 0, the illumination coherence decreases, becoming fully incoherent in the limit as $\sigma \to \infty$. For most objects, a value of $\sigma \geq 2$ is indistinguishable from fully incoherent illumination. By default, OSLO assumes that the effective source is a uniform disk, but options are provided for annular effective sources and effective sources with Gaussian irradiance profiles.

Because of the lack, in the general case, of a transfer function, OSLO needs to know the form of the ideal image of interest. This ideal image consists of a bar target, with a specified bar width, period, and number of bars. The default ideal image consists of alternately opaque and clear bars (i.e., unit modulation), but the modulation and background may be adjusted, if desired. Also, there may be a phase transmittance difference between the bars.

Since it is assumed that the imaging is stationary, as much of the calculation as possible is performed in the Fourier domain. For efficiency, we make use of the Fast Fourier Transform (FFT) algorithm. Thus, the same sampling considerations that were discussed in Chapter 6, with regard to the calculation of the point spread function, are applicable to the partial coherence calculations. To avoid aliasing, the number of rays traced across the pupil diameter is always taken to be half of the number of points in the image (i.e., the size of the FFT array). Thus, for the partial coherence calculations, the sampling interval in the image, Δy, is given by

$$\Delta y = \frac{\lambda_0}{4\,NA} \tag{7.61}$$

where λ_0 is the wavelength and *NA* is the numerical aperture. Also, the size of the one-dimensional image is

$$\text{Image patch size} = N\,\Delta y = \frac{N\lambda_0}{4\,NA} \tag{7.62}$$

where *N* is the number of points in the image. For the partial coherence calculations, we see from Eq. (7.62) that the only way to increase the size of the image is to increase the number of points in the FFT array, which must, of course, be a power of two.

---

1 H. H. Hopkins and M. J. Yzuel, "The computation of diffraction patterns in the presence of aberrations," Optica Acta **17**, 157-182 (1970).
2 D. Gloge, "Weakly guiding fibers," Appl. Opt. **10**, 2252-2259 (1971).
3 R. E. Wagner and W. J. Tomlinson, "Coupling efficiency of optics in single-mode fiber components," Appl. Opt. **21**, 2671-2688 (1982).
4. H. H. Hopkins, "On the diffraction theory of optical images," Proc. Roy. Soc., **A217**, 408-432 (1953).
5. J. W. Goodman, *Statistical Optics*, Wiley, 1985.

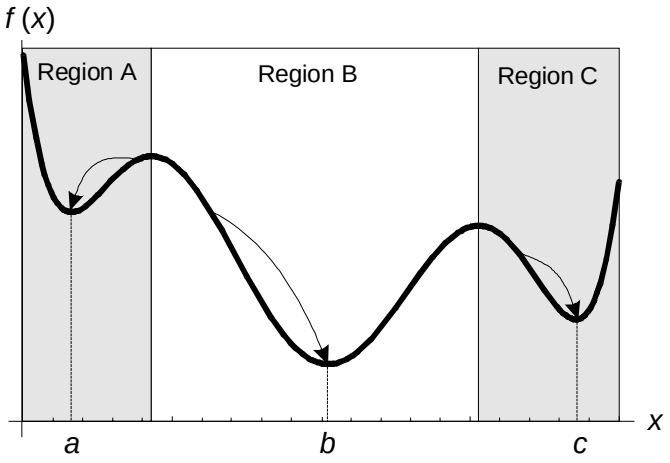
Chapter 8 Optimization

The term *optimization* in optical design refers to the improvement of the performance of an optical system by changing the values of a subset of the system's constructional parameters (*variables*). Typically, the variables are quantities such as surface curvatures, element and air-space thicknesses, tilt angles, etc. The system's performance is measured by a user-defined *error function*, which is the weighted sum of squares of *operands*, that represents an estimate of the difference in performance between a given optical system and a system that meets all the design requirements.

The conventional approach to optimization is *iterative*, in which the user chooses initial values for the variables (a *starting point*) and an optimization algorithm is applied that repeatedly attempts to find new values for the variables that yield ever lower error function values. This approach to optimization, illustrated in the figure below, depends heavily upon the choice of the starting point: If a starting point is chosen from Region A or Region C, the program will proceed the corresponding *local* minimum at $x = a$ or $x = c$, rather than the *global* minimum at $x = b$.

If the starting point is chosen from Region B, however, the program will proceed to the *global* minimum at $x = b$. In optical design, the dimensionality (number of independent variables) is high and the number of local minima is typically large, making the choice of starting point crucial to the success of optimization. Starting points from which the optimizer can proceed to the global minimum (or a suitable local minimum) are typically determined by experience or by finding an existing design with properties similar to those desired.

The prime source of difficulty in optimizing a given error function is that the minimum usually depends on balancing, rather than removing, the aberrations of the system. This process creates artificial minima caused by numerical effects in the optimization routines, and these are often indistinguishable from the artificial minima caused by aberration balancing. In a practical problem, this can be a source of considerable frustration to the optical designer.



There are many local optimization methods available; most of those employed in optical design are *least-squares methods*, meaning that they produce a solution that is as close as possible to the desired solution when a complete solution is not possible (ordinarily there are more *operands* than *variables*). All versions of OSLO include the standard optimization algorithm called *damped least squares*, and OSLO Premium includes several other methods that work around the stagnation sometimes experienced with this method.

Damped least squares

The damped least squares (DLS) optimization used by OSLO is the most popular optimization algorithm in optical design software – a variant of DLS is included in nearly every optical design program. In DLS the error function $\phi(\mathbf{x})$ is expressed in the form of a *weighted sum of squares*:

$$\phi(\mathbf{x}) = \sum_{i=1}^{m} w_i f_i^2(\mathbf{x}) \tag{8.1}$$

where the vector **x** represents the set of optimization variables,

$$\mathbf{x} = \langle x_1, x_2, x_3, \ldots, x_n \rangle \tag{8.2}$$

and the *f*'s are called *operands*. The individual operands are defined by

$$f_i = (c_{1i} \oplus c_{2i}) \tag{8.3}$$

The *c*'s are called *components*. Each operand contains two components that are linked by a math operator, which may be addition, subtraction, multiplication, division, exponentiation, greater than, or less than. Usually the first component is a ray displacement, the second component is a target value, and the operator is subtraction, but the definition allows for more complex operands. The *w*'s are weights, which can be used to account for the relative importance of different operands. To simplify the discussion here, we will assume that the weights are all unity.

OSLO by default reports error function values as root-mean-square (RMS) error, which is the square root of the error function. If all the operands are of the same type, the rms error function gives a convenient indication of the average size of an operand.

The error function has its minimum value when all the operands are zero. It is useful to write $\phi(\mathbf{x})$ in vector notation as

$$\phi(\mathbf{x}) = \mathbf{f}^T\mathbf{f} \tag{8.4}$$

where

$$\mathbf{f} = \langle f_1, f_2, f_3, \ldots, f_m \rangle \tag{8.5}$$

Minimization of the error function is based on a piece-wise linear model of the operand dependencies on the variables. That is, the change in the i^{th} operand due to a change in the j^{th} variable is assumed to be given by

$$f_i(x_j + \Delta x_j) = f_i(x_j) + \frac{\partial f_i}{\partial x_j}\Delta x_j \tag{8.6}$$

Of course, this is an idealized model. In a real situation, there may be nonlinearities that require second or higher-order derivatives of the operands in the above equation. In addition, there may be no solution to the real optimization problem that reduces the value of f_{1i} to zero, because of these nonlinearities (i.e. because of physics). There is always, however, a least-squares solution for Δx_j, at which the operand has its minimum allowed value.

To describe the optimization of systems in which there are several operands and several variables, it is best to use matrix notation. If we understand the *f*'s to mean the changes in the operands relative to their minimum values, we obtain a set of equations for the change in **x** that minimizes the error function:

$$\mathbf{A}\Delta\mathbf{x} = -\mathbf{f} \tag{8.7}$$

where **A** is the derivative matrix of each of the operands with respect to each of the variables:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{8.8}$$

Ordinarily there are more equations than variables, so there is no direct solution. However, there is a *least squares* solution for which the error function has a minimum value.

The least squares algorithm operates as follows: given an initial estimate $\mathbf{x}_0$ of the minimum (a *starting point*), iteratively determine new estimates of the minimum $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$ by solving the following linear system, called the *least-squares normal equations*, for the *change vector* $\Delta\mathbf{x}_k$:

$$\mathbf{A}_k^T \mathbf{A}_k \Delta\mathbf{x}_k = -\mathbf{A}_k^T \mathbf{f}_k \tag{8.9}$$

where k is the iteration number. Of course, if the dependence of the operands on the variables is linear, iteration is not needed. The iteration is provided to account for the nonlinearities in real systems.

The above scheme usually doesn't work. For a nonlinear system, the change vector $\Delta\mathbf{x}$ typically diverges. The solution to this problem is to add a damping term μ to the equations that prevents large values for $\Delta\mathbf{x}$, which leads to the *damped least squares normal equations*

$$\left(\mathbf{A}_k^T \mathbf{A}_k + \mu_k \mathbf{I}\right) \Delta\mathbf{x}_k = -\mathbf{A}_k^T \mathbf{f}_k \tag{8.10}$$

where $\mathbf{I}$ is the identity matrix. It is possible to derive the damped least squares equations mathematically, by including second-order derivatives in Eqs. (8.7), but for practical optical design, it isn't worth the effort, because it is computationally expensive, and the nonlinearities are often higher than second-order. The important points to understand about the damped least squares equations are

1. The damped least squares equations are formed by adding a term proportional to the change in the system variables to the iteration equations to reduce the magnitude of the change vector.
2. The dimensionality of the equations is $n \times n$, i.e. equal to the number of variables. The process of multiplying $\mathbf{A}$ by its transpose effectively sums over the number of operands (which is usually larger than the number of variables).
3. The solution point for the damped least squares equations is the same as the solution point for the normal least squares equations. This is because at the solution point, the value of $\Delta\mathbf{x}$ is zero.

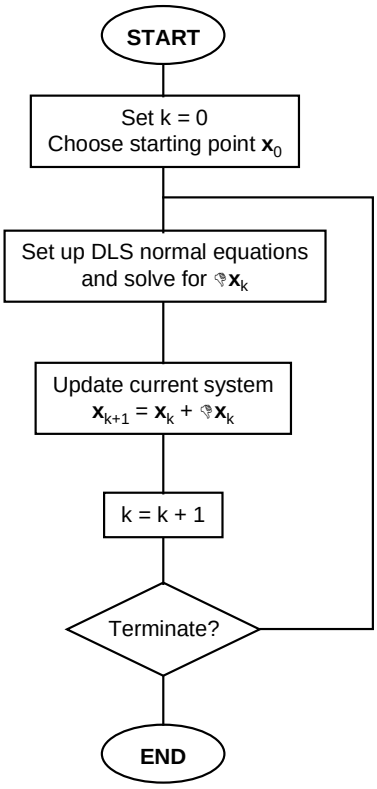
The last point is significant in that it implies that there is considerable freedom in the way that damping is applied. There is no particular need to restrict the damping term to a scalar constant times the identity matrix. Each element along the diagonal can have any desired value, without changing the solution point. This leads to different forms of damping, in particular the multiplicative damping used in OSLO, in which the identity matrix $\mathbf{I}$ is replaced by another matrix $\mathbf{D}$ whose diagonal elements are the same as the diagonal elements of the $\mathbf{A}^T\mathbf{A}$ matrix:

$$\mathbf{D} = \begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & d_{nn} \end{bmatrix}, \quad d_{jj} = \sum_{i=1}^{m} \left(\frac{\partial f_i}{\partial x_j} \right)^2 \tag{8.11}$$

OSLO also allows the damping factor μ_j, to be different for different columns of the $\mathbf{A}^T\mathbf{A}$ matrix, so that so that the user has control over the damping of individual variables.

The above derivation of the damped least squares equations has emphasized simple concepts, because in actual practice an experimental approach to optimization based on these concepts has

proved more successful than more elaborate theory based on mathematical elegance. It is up to the program, as directed by the designer, to develop an iterative technique for optimizing a system according to its particular requirements. The basic scheme for damped least squares iteration is shown in the figure below. This is the form of DLS that is invoked by the **ite std** command in OSLO. The damping factor is held fixed at the user-specified value and iteration is terminated when the specified number of iterations have been performed.



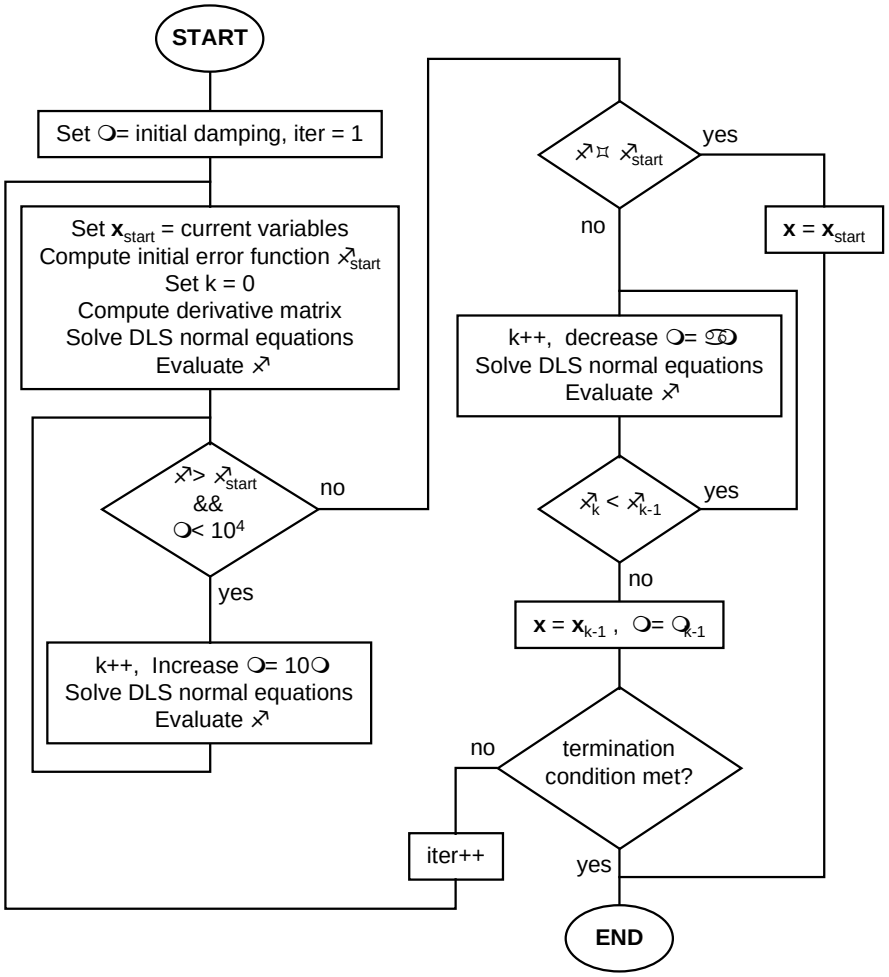
Damping

As mentioned above, the damping factor helps to stabilize the damped least squares algorithm by preventing the generation of large change vectors. Without damping ($\mu = 0$), if $\mathbf{A}^T\mathbf{A}$ is singular, no solution to the normal equations exists. Adding the damping term effectively adds a factor that is proportional to the square of the length of the change vector to the error function, thereby penalizing large steps.

The problem now becomes the choice of the damping factor. If μ is too small and $\mathbf{A}^T\mathbf{A}$ is nearly singular, the change vector will be very large and DLS will become unstable. On the other hand, if μ is too large the change vectors tend to be tiny steps in the direction of steepest descent (i.e., along the gradient of ϕ), and DLS becomes inefficient. Since proper choice of the damping factor is essential for stability and efficiency, OSLO contains an algorithm for automatically choosing the damping factor. This algorithm is invoked by the **ite ful** command, with each full iteration consisting of a search for the optimal value of the damping factor, as illustrated in the figure on the following page.

At the start an iteration sequence, the damping is reset to the value specified as the starting value of damping factor (**opds**) operating condition, normally set to a small value ($\approx 10^{-8}$). The algorithm increases the damping in large steps until it is above the optimum, then decreases it using the scaling factor for damping (**opdm**), which is indicated as α in the figure, until it finds the optimum.

It should be noted that this search for the optimal damping factor is relatively inexpensive since the derivative matrix need be computed only once, at the start of each full iteration; only the damping is varied within a single full iteration. Iteration is terminated when one of two conditions is satisfied: the specified number of full iterations have been performed or the relative change in error function is less than the "Percent improvement for continuing full iterations" (**opst**) operating condition value for two consecutive full iterations.



Constraints

Frequently it is necessary to maintain certain conditions as the lens changes during optimization. For example, a certain minimum back focal distance may be required, or the speed (*f*–number) of the lens may be required to remain constant. Such conditions can be maintained in three ways:

Solves. Solves allow the exact maintenance of paraxial properties. For example, placing an axial ray angle solve with value –0.1 on the last surface of the lens maintains a constant *f*–number of 5, assuming that the object is at infinity.

Penalty terms in the error function. By adding a term (operand) to the error function that is proportional to the deviation of a given quantity from its desired (or maximum or minimum) value, the optimizer is encouraged to maintain the desired conditions. For example, adding an operand "TH(7)>40.0" adds a penalty term to the error function if the thickness of surface 7 is less than 40.0.

Constraint operands. OSLO allows the explicit specification of constraints to be applied to the optimization process independently of the error function. These constraints are applied through the

method of *Lagrange multipliers*, in which the damped least squares equations are solved subject to the condition that certain of the operands have exact values (in the linear approximation).

Each of these methods has advantages and disadvantages. Solves are appropriate only when valid paraxial raytrace data are available at the surface containing the solve. Furthermore, solves can produce instability during optimization since the solved quantity may vary in a highly nonlinear fashion in response to changes in the variables.

Penalty terms can be difficult to implement since their effectiveness depends highly upon their weighting. If the weight is too small, the penalty may be dominated by the remainder of the error function and the optimizer will not enforce the penalty strictly. If the weight is too large, the penalty may dominate the remainder of the error function and the result will be a system that satisfies the constraints but has poor performance otherwise.

Constraint operands are useful when valid paraxial raytrace data are not available, when solves might be unstable, or when appropriate weights for penalties cannot be determined. However, the number of *active* (i.e. nonzero-valued) constraints may not exceed the number of variables. Furthermore, DLS works best when the constraints are satisfied before the error function is optimized; the behavior of DLS is less predictable if it is attempting simultaneously to satisfy the constraints and minimize the error function, particularly if the initial values of the constraints are far from their target values.

Variables

In the course of the design of an optical system, a subset of the system's constructional parameters is allowed to change. These parameters are called *variables*. There are two classes of variables: *independent* variables, whose values are changed directly by the user or by the optimizer, and *dependent* variables, whose values change as indirect results of changes to other parameters. In OSLO, dependent variables are variables that change as the results of solves and pickups; the term "variable" is used henceforth to refer exclusively to independent variables.

The set of variables is defined either by designating various quantities as variable in the Surface Data spreadsheet, or by adding entries to the Variables spreadsheet. The procedures for working with variables in OSLO are described in the Help system. There are, however, some specific issues about the way variables are handled in optimization that are discussed here.

Boundary conditions

In many optimization problems, it is necessary to specify bounds on the variables to prevent the generation of solutions that cannot be realized. For example, solutions may have lenses with negative edge thicknesses, more than one lens occupying the same physical space, etc. There are various ways to prevent the generation of such solutions, but before using them, you should realize that the overuse of boundary conditions can impede the optimization process. It is not unusual, during the course of optimizing a system, to generate intermediate solutions that are non-realizable, which spontaneously convert to realizable systems as the optimization progresses. If you disallow these intermediate solutions by imposing boundary conditions, you may not reach the desired optimum.

Boundary conditions can be imposed either in the definition of a variable, or by creating operands in the error function that serve to restrict the allowed values of the variable in question. In the variables spreadsheet, the columns labeled Minimum and Maximum are the lower and upper bounds, respectively, on the variable. If the value v of a variable is less than the lower bound v_{min}, a penalty term is added to the error function that is proportional to the boundary violation:

$$\text{penalty} = (\mathbf{opbw})(v_{\min} - v)^2 \tag{8.12}$$

Similarly, a penalty is added to the error function if the value of v is greater than the upper bound v_{max}:

$$\text{penalty} = (\mathbf{opbw})(v - v_{\max})^2 \tag{8.13}$$

The quantity **opbw** in the above equations is the operating condition labeled "Weight of boundary condition violations" in the variables spreadsheet. Note that if the maximum bound is less than or equal to the minimum bound, no penalty term is added to the error function.

In practice, thicknesses usually need more attention to boundary conditions than curvatures. Accordingly, default boundary conditions are applied to thicknesses when they are entered as variables. The values that are used can be set in the variables spreadsheet. The defaults are set up to define a wide range that essentially only prevents the generation of negative axial thicknesses, and need to be adjusted to fit particular design constraints.

Edge-thickness boundary conditions must be entered as operands. A common way to do this (which is used in the default error functions generated in OSLO) is to determine the ray with the greatest ray height on each surface, and require that the distance along this ray from the preceding surface be positive. Although this condition does not take into account the fact that the ray trajectory is not usually parallel to the axis, for most cases the calculation is close enough.

Boundary condition operands are formed using the *greater than* or *less than* operators, e.g., "TH(3)>0.5." Such an operand evaluates to 0.0 if it is true, and to the difference between the components if it is false. More complex relationships can be set up using cross-reference components, as described on p. 8-211. One-sided operands are restricted to minimization rather than constraint mode. By doing this, the weights can be adjusted to allow a small penetration into the disallowed region.

In ordinary DLS optimization, the best number of boundary conditions is zero. Boundary conditions are useful for defining a solution region, but generally impede the optimization process, and should be removed when they are not needed. If a variable persistently violates a boundary condition, it should be set to a fixed value at the edge of the allowed region.

In ASA global optimization, where thousands of systems are evaluated without designer intervention, boundary conditions play a different role. Here, boundary conditions define the search region, and are essential to the operation of the algorithm.

Derivative increments

When formulating the normal equations, damped least squares must compute the derivatives of the operands. This is done by using a finite-difference approximation:

$$\left.\frac{\partial f_i}{\partial x_j}\right|_{\mathbf{x}} \approx \frac{f_i\left(x_1, x_2, \ldots, x_j + \delta x_j, \ldots, x_n\right) - f_i\left(x_1, x_2, \ldots, x_j, \ldots, x_n\right)}{\delta x_j} \tag{8.14}$$

The default derivative increment (**opdi**) operating condition determines the step size δx_j that is used to compute the derivative. There is some question concerning the optimum size of δx_j. One argument considers that the increment should approximate the tangent to a nonlinear curve; another advocates that the increment should be based on the predicted magnitude of the solution vector.

The default conditions used in OSLO are that the derivative increments are chosen according to the type of variable and the entrance beam radius. The derivative increments are normally fixed at the values provided when the variables are created, but if the default derivative increment is set to zero, the program will adaptively set the actual derivative increment during optimization, for any variable whose nominal derivative increment is zero.

Usually, the results of optimization are not a strong function of the size of the derivative increments. An exception is the case where very high-order aspherics are used. In such a case, surface sag differences become proportional to the aperture raised to the n^{th} power, and careful attention must be paid to the derivative increments of the aspheric coefficients. Often, it is a good idea to scale such systems to a focal length of about one during optimization to provide optimum balance between different order aspherics. Similar reasoning applies to diffractive surfaces described by high-order polynomial phase functions.

Variable damping

As mentioned before (see Eq. (8.11)), it is not necessary to apply the same damping to each term on the diagonal of the $\mathbf{A}^T\mathbf{A}$ matrix. OSLO permits each variable to be damped individually. These damping factors on the individual variables allow more control over the behavior of the variables during optimization. Larger values of the damping for a given variable tend to prevent large changes in that variable during optimization.

The optimum value for each damping factor must be found experimentally. Unfortunately, the amount of computational work required to do this routinely is too large to justify the effort. That is, there are significant approximations made in formulating the problem as a piece-wise linear model, and it may be more efficient to compute a new derivative matrix at an approximate solution point than to go to extreme lengths to obtain an exact solution to an approximate model.

Operands

OSLO uses the term *operands* to denote the terms in the error function, that is, the error function is the (weighted) sum of squares of the operands. Although operands are primarily used in optimization, they are also useful for various other purposes, ranging from tolerancing to special-purpose evaluation.

Operands are compiled from operand definitions entered as source code (i.e. text) into an internal representation that can be computed with maximum efficiency. Thus operands in OSLO are *expressions* rather than *commands*. Each operand consists of one or two components and is of the form

$$f = [-1]\, c_1 \oplus c_2 \qquad (8.15)$$

where $\oplus$ is one of the mathematical operators from the following table and c_1 and c_2 are the *components*.

Operator	Description
+	Addition: operand value = component 1 value + component 2 value
–	Subtraction: operand value = component 1 value – component 2 value
*	Multiplication: operand value = component 1 value × component 2 value
/	Division: operand value = component 1 value / component 2 value
**	Exponentiation: operand value = component 1 value ** component 2 value
<	Less-than: operand value = component 1 value – component 2 value if component 1 value ≥ component 2 value; operand value = 0 otherwise
>	Greater-than: operand value = component 1 value – component 2 value if component 1 value ≤ component 2 value; operand value = 0 otherwise

Component classes

Using the two-component form shown above, it is possible to build operands that effectively consist of a greater number of components, since it is possible to define an operand component whose value is that of another operand. Each component is of one of the following classes:

System components measure physical properties of the lens such as curvature, thickness, aspheric and GRIN coefficients, etc. Any quantity that can be specified as an optimization variable can also

be used as a system operand component. Additionally, surface sag, edge thickness, power, and axial length can be specified.

Aberration and paraxial data components represent the values of third- and fifth-order aberration coefficients, seventh-order spherical aberration, paraxial chief- and axial-ray heights and slopes, and primary and secondary chromatic aberrations. Note that these components may be invalid for systems with certain types of special data.

Ray components are the values derived from aiming exact rays from specified field points at specified pupil points and tracing the rays through the system. The values of the field points are contained in the *field points set* and the values of the pupil points are contained in the *ray set*, each of which is described below.

Spot diagram components (in OSLO Premium) are the values of the *MTF* or RMS wavefront error, computed by tracing a spot diagram from specified field points. The field point values are taken from the field points set, and the grid size (aperture divisions) and wavelengths for the spot diagrams are taken from the spot diagram set.

CCL and SCP components are values computed by CCL or SCP operand functions. These components allow the computation of quantities not easily specified by the built-in component types.

External components (in OSLO Premium) are values computed in functions in dynamic-link libraries written in compiled, high-level languages such as C. These functions are much faster than the equivalent CCL or SCP functions, but require more effort to develop.

Cross-reference components are the values of other operands in the operands set. These components allow the effective construction of operands with more than two components.

Statistical components are the mean (average) or root-mean-square (RMS) values of one or more operands. They are typically used in computing RMS spot size or RMS wavefront error.

Constant components are simply components with constant numeric values.

The specific operand components in the various classes are enumerated in the Help system.

Operand component syntax

Each component contains one or more arguments, which are entered as integers separated by commas. When an operand definition is entered in OSLO, it is immediately compiled into an internal representation. The definition echoed on the display is a reconstructed version of the operand definition, which reflects how the entered operand was compiled by the program. The displayed version uses the most efficient syntax, which may not be the syntax that was used in entering the definition.

The syntax of operand components varies according to the class of operand component. With the exceptions of cross-reference and constant components, each component can take one or more integer arguments that specify such quantities as ray number (index into the ray set), wavelength number (index into the wavelengths set), surface number, configuration number, etc. Many of these arguments have default values that need not be entered explicitly by the user; indeed, any default arguments at the end of the arguments list for a given component will not be displayed by OSLO even if the user enters them explicitly. The following table shows the various operand component arguments and their default values.

Argument	Default
Surface number	0 (indicates image surface)
First surface number of surface range	0 (indicates object surface)
Last surface number of surface range	0 (indicates image surface)
Configuration number	1
Wavelength number	1
Field point number	none
Ray number	none
Spot diagram number	none
Spatial frequency	none

There are several conventions that apply to component arguments:

- A surface number of zero represents the image surface, and negative surface numbers represent surface numbers relative to that of the image surface. For example, the component "CV(–2)" is the curvature of the surface that is two surfaces before the image; if surface number 7 is the image surface, this is the curvature of surface 5. This convention allows the insertion and deletion of surfaces without affecting the actual surface represented in the component.
- Nonexistent wavelength numbers, field point numbers, ray numbers, and spot diagram numbers may not be entered. For example, if there are only two entries in the field points set, the component "Y(3, 1)", which is the *y*-intercept in the image plane of the ray from field point number 3, ray number 1, may not be entered.
- If a nonexistent configuration number is entered, the component value is zero. For example, if only one configuration is currently defined, the component "TH(5, 2)", which is the thickness of surface 5 in configuration 2, will have zero value. If a second configuration is later defined, "TH(5, 2)" will represent the actual value of thickness 5 in configuration 2.

The following sections describe the operand component classes and give examples of each component class.

System operand components

The focal length (EFL), transverse magnification (TMAG), angular magnification (AMAG), power (PWR), edge thickness (ETH), and axial length (LN), operand components have the syntax

<*component*>(*first surface #*, *last surface #*, *configuration #*)

where *surface #* is the surface number, *first surface #* and *last surface #* are the numbers of the first and last surfaces of a range of surfaces, and *configuration #* is the configuration in which the component is measured. The remaining components only include a single surface:

<*component*>(*surface #*, *configuration #*)

Example 1: Define a component that is the thickness of surface 5 in configuration 1. If you enter "TH(5, 1)" OSLO will display "TH(5)" since configuration 1 is the default. Since it is not necessary to enter default arguments, you may also simply enter "TH(5)".

Example 2: Define a component that is the thickness of surface 5 in configuration 2. If you enter "TH(5, 2)" OSLO will leave the definition unchanged since configuration 2 is not the default.

Example 3: Define a component that is the axial length from surface 1 to the image surface in configuration 1. Since the last two arguments are equal to their defaults, this component may be entered as "LN(1, 0, 1)", "LN(1, 0)", or simply "LN(1)". In any case, OSLO will display "LN(1)".

Example 4: Define a component that is the axial length from surface 1 to the image surface in configuration 2. Since the last argument is not equal to its default, all three arguments must be entered explicitly: “LN(1, 0, 2)”.

Aberration and paraxial data components

With the exceptions of primary axial color (PAC), primary lateral color (PLC), secondary axial color (SAC), and secondary lateral color (SLC), the syntax of aberration and paraxial data components is

<*component*>(*wavelength #, surface #, configuration #*)

PAC, PLC, SAC, and SLC components have the following syntax:

<*component*>(*surface #, configuration #*)

where *wavelength #* is the wavelength number (index in the wavelengths set), *surface #* is the surface number, and *configuration #* is the configuration in which the component is measured.

Example 1: Define a component to measure the third-order spherical aberration coefficient at the image surface (i.e. the third-order spherical aberration for the system as a whole) in wavelength 1 and configuration 1. If you enter “SA3(0, 1, 1)” OSLO displays simply “SA3” since all three arguments have their default values.

Example 2: Define a component to measure the total primary lateral color for the system (i.e. the primary lateral color at the image plane) in configuration 2. The only form in which this can be specified is “PLC(0, 2)” since the last argument is not the default.

Ray operand components

Each ray operand component is computed by tracing a ray in a specified wavelength from a specified field point (relative coordinates on the object surface), through a specified pupil point (relative coordinates on the entrance pupil), through the lens, and on to the image surface. The wavelength, object coordinates, and pupil coordinates are not specified directly in the operand component; rather, they exist as entries in the wavelengths, field points, and ray sets to which the operand components refer. This indirect representation enhances efficiency: Since information from a single ray is often used in multiple operand components, OSLO traces only as many rays as necessary. To create ray operands components, then, it is necessary to understand the field points set and the ray set (since the wavelengths set is used extensively outside optimization, it is not described here).

The field points set defines the field points that will be used for optimization. These are not necessarily the same as those used to draw rays on lens drawings, which are specified as the lens drawing operating conditions. Each field has a number and 10 data items that are used in conjunction with all rays traced from that field point. The data items are as follows.

FBY, FBX, and FBZ are the relative *y*-, *x*-, and *z*-coordinates, respectively, of the field point on the object surface. These are specified as fractional values relative to the object height. YRF and XRF are relative coordinates on the reference surface (normalized to the aperture radius of this surface) for reference rays traced from the field point. FY1, FY2, FX1, and FX2 are, respectively, lower- and upper-meridional and sagittal vignetting factors for the field point. These factors allow the pupil (or reference surface) coordinates of ray set rays to be adjusted to accommodate vignetting without changing the ray set. If FY and FX are the relative coordinates specified in the ray set (see below) for a given ray, the actual relative pupil or reference surface coordinates for rays traced from this field point are given by

$$\text{traced FY} = \frac{\left[\left(\text{rayset FY}\right)+1\right]\left[\text{FY2}-\text{FY1}\right]}{2}+\text{FY1} \tag{8.16}$$

$$\text{traced FX} = \frac{\left[\left(\text{rayset FX}\right)+1\right]\left[\text{FX2}-\text{FX1}\right]}{2}+\text{FX1} \tag{8.17}$$

The defaults for FY1, FY2, FX1, and FX2 are –1.0, 1.0, –1.0, and 1.0, respectively, so that the traced FY and FX are the same as the ray set FY and FX. The effect of the vignetting factors is to transform a square grid of rays on the pupil into a rectangular grid or to transform a circular

pattern of rays into an elliptical pattern, thereby approximating the shape of the vignetted pupil for off-axis field points. Normally, it is also desirable to set YRF to (FY1 + FY2)/2 and to set XRF to (FX1 + FX2)/2 so that reference rays pass approximately through the center of the vignetted pupil.

The last data item for a field point is its weight WGT, which is used only during automatic generation of error functions.

The second table of data needed to define optimization rays is the ray set. Each ray in the ray set has a number and data items that specify a ray type, fractional aperture coordinates, and weight. The weight is not currently used in OSLO. The ray type is either ordinary or reference. Ordinary rays are traced through the specified point on the entrance pupil and then through the lens system. Reference rays are iterated so that they pass through the specified point on the reference surface. Ordinary rays are faster than reference rays, and should be used for most optimization tasks. FY and FX are the relative coordinates on the entrance pupil (for ordinary rays) or the reference surface (for reference rays) through which the ray passes.

The type of a ray is an important factor in determining the types of components that can use it. The are essentially three classes of ray-based operand components: those that are computed for both ordinary and reference rays, those that are computed only for ordinary rays, and those that are computed only for reference rays.

The components computed for all types of rays are X, Y, Z, RVK, RVL, RVM, NVK, NVL, NVM, XA, YA, PL, and OPL (the definition of these components is described in the Program Reference manual, p. 156). These are data that involve only a single ray. The syntax for these components is

<*component*>(*field #*, *ray #*, *wavelength #*, *surface #*, *configuration #*)

The other two types of ray components are only available in image space, and have the syntax

<*component*>(*field #*, *ray #*, *surface #*, *configuration #*)

where *field #* is the field point number (index into the field points set), *ray #* is the ray number (index into the rayset), *wavelength #* is the wavelength number (index into the rayset), *surface #* is the surface number, and *configuration #* is the configuration number.

It should be noted that the field point set and the ray set do not by themselves impose any computational burden on the program during optimization. Only rays that are referenced in operand definitions are actually traced, and if there are multiple references to a particular ray from different components, that ray is traced only once.

The examples of ray components below are based on the following field point and ray sets.

```
*RAYSET
FPT       FBY/FY1       FBX/FY2       FBZ/FX1       FYRF/FX2      FXRF/WGT  CFG/GRP
F 1       1.000000        --            --            --            --        -
         -1.000000      1.000000     -1.000000      1.000000      1.000000     -
F 2       0.700000        --            --            --            --        -
         -1.000000      1.000000     -1.000000      1.000000      1.000000     -
RAY         TYPE          FY            FX            WGT
R 1       Ordinary     -1.000000        --          1.000000
R 2       Ordinary     -0.500000     -0.500000      1.000000
```

Example 1: Define an operand component that measures the optical path length from the entrance pupil to surface 5 of a ray from field point (FBY = 0.7, FBX = 0.0), pupil coordinates (FY = –0.5, FX = 0.5) wavelength 3, configuration 1.

Field point 2 is defined as (FBY = 0.7, FBX = 0.0). Ray 2 is defined as (FY = –0.5, FX = 0.5). The OPL operand that will be defined will use field point 2, ray 1, wavelength 3, and surface 5. Since the operand is to use configuration 1 (the default), there is no need to specify the configuration. The definition is thus OPL(2, 2, 3, 5).

Example 2: Create a one-sided operand that specifies that the path length from surface 3 to surface 4 along a ray from field point (FBY = 1.0, FBX = 0.0) through pupil point (FY = –1.0, FX = 0.0) be greater than 0.5 lens units. The PL operand uses field point 1, ray 1, wavelength 1, and surface 4, so the definition is PL(1, 1, 1, 4)>0.5.

Note that the wavelength number was specified explicitly despite the fact that it is the default, since the wavelength number argument is followed by the surface number argument, which does not have the default value here. The configuration number argument, however, need not be specified here because it has the default value (1) and is the last argument.

Spot diagram operand components

OSLO Premium allows the direct optimization of *MTF* through the use of spot diagram operands. The main purpose of these operand components is for the construction of specialized user-defined tolerancing error functions, or for differential optimization of *MTF* for a system that is close to meeting specifications. Their use in ordinary optimization is discouraged since *MTF* (and RMS wavefront error) are typically highly nonlinear functions of optimization variables, and damped least squares often performs poorly under such conditions.

There are only three spot diagram operand components: sagittal *MTF* (MTX), tangential *MTF* (MTY), and RMS wavefront error (WVF). MTX and MTY components have the following syntax:

<*component*>(*spot diagram #*, *spatial frequency*, *configuration #*)

WVF operand components have the syntax

WVF(*spot diagram #*, *configuration #*)

where *spot diagram #* is an index into the spot diagram set (see below), *spatial frequency* is the spatial frequency at which *MTF* is computed (in cycles/mm), and *configuration #* is the configuration number in which the component is evaluated.

As with ray operands, the information (field point, wavelengths, and aperture divisions) that determines the spot diagram to be traced is not specified directly in the component definition, but is specified in a separate data set called the *spot diagram set*.

The Spot Diagram set contains five data items. FPT is the field point number, or index into the field points set. The spot diagram will be traced from this field point. APDIV is the number of aperture divisions across the pupil for the spot diagram ray grid. FIRST WVL is the index into the wavelengths set of the first of one or more wavelengths in which the spot diagram is to be traced. NBR WVLS is the number of wavelengths in which the spot diagram will be traced. The wavelengths traced will therefore be (FIRST WVL, FIRST WVL + 1, ... , FIRST WVL + NBR WVLS – 1).

Normally, *MTF* (i.e. MTX or MTY) alone is not placed in the error function, but rather the deviation of the *MTF* from a target specification value, or the diffraction-limited *MTF* value.

Example: Create an operand that measures (0.9 – polychromatic tangential *MTF*) for the field point (FBY = 1.0, FBX = 0.0) at a spatial frequency of 10 cycles/mm.

Create the Spot Diagram set entry, referring to field point 1.
Since this is to be a polychromatic spot diagram, the first wavelength is number 1, and the number of wavelengths is 3 (assuming that three wavelengths are defined):

```
*SPOT DIAGRAM SET
   SD     FPT      APDIV    FIRST WVL    NBR WVLS
S   1      1     17.030000      1           3
```

Finally, create the operand using spot diagram number 1 and a spatial frequency of 10 cycles/mm: 0.9–MTY(1,10).

CCL and SCP components

CCL and SCP components allow the incorporation into the error function of terms that cannot be computed easily by the built-in operand components. Through CCL and SCP it is possible, for example, to retrieve numerical output of OSLO commands from the Spreadsheet Buffer.

CCL and SCP components have the following syntax:

OCM<*array element #*>(*configuration #*)

where <*array element #*> is an integer index into the CCL/SCP global array variable **Ocm** and *configuration #* is the configuration number in which the component is to be evaluated.

The procedure for setting up CCL/SCP components consists of three steps:

1. Write a procedure in CCL or SCP (or use one of the supplied procedures) that computes the value(s) of the term(s) that are to be added to the error function. The value of each term should be assigned to one of the elements of the Ocm array, which is a built-in global CCL/SCP variable. Below is a sample CCL command from the file *optim_callbaks.ccl* that traces a spot diagram from each of three field points and assigns to the global variables Ocm[0], Ocm[1], and Ocm[2] the values of the RMS spot size at each field point.

```
cmd oprds_spot_size(void)
{
        set_preference(output_text, off);
        trace_ref_ray(0.0);
        ssbuf_reset();
        spot_diagram(mon, 10.0);
        Ocm[0] = c4;
        trace_ref_ray(0.7);
        ssbuf_reset();
        spot_diagram(mon, 10.0);
        Ocm[1] = c4;
        trace_ref_ray(1.0);
        ssbuf_reset();
        spot_diagram(mon, 10.0);
        Ocm[2] = c4;
        set_preference(output_text, on);
}
```

2. Set the value of the “Command for CCL/SCP operands” optimization operating condition to the name of the CCL or SCP command from Step 1. If an SCP command is used to compute the added components, the command name entered here should start with an asterisk (*****). For the example above, the command name is **oprds_spot_size**.

3. Add OCM<*array element #*> operand components corresponding to the elements of the Ocm array that are assigned values in the CCL or SCP command: for the above case, add **ocm1**, **ocm2**, and **ocm3** to the operands.

Whenever the error function is evaluated, the designated CCL or SCP command is invoked once for each lens configuration. The CCL or SCP function can check the value of the cfg global variable to determine the current configuration number (if necessary). The same Ocm array is used by all configurations, so CCL/SCP operand-component commands need to ensure that each configuration uses a distinct subset of the Ocm array. For example, suppose that the operands set contains two CCL operands, one of which is calculated in configuration 1 and the other in configuration 2. The operands set would then contain the following two entries: OCM0 and OCM1(2), and the CCL command to compute the operand components would contain code such as the following:

```
        if (cfg == 1)
                Ocm[0] = <value of config. 1 operand>;
        else if (cfg == 2)
                Ocm[1] = <value of config. 2 operand>;
```

Normally, CCL/SCP operand-component commands that invoke OSLO commands that produce text output set the **output_text** preference to “Off” before invoking the OSLO commands and then restore the **output_text** preference to “On” (see the example on p. 8-210). Otherwise, a large amount of text output may appear during optimization as the CCL/SCP command is invoked repeatedly.

External components

External operand components provide a means for incorporating into error functions the results of lengthy or computationally intensive calculations that are impractical in CCL or SCP. This facility is implemented in a manner similar to CCL/SCP components, except that the calculations are performed in a *dynamic-link library* (DLL) that is written in a high-level compiled language, such as C. The advantages of this approach are speed and flexibility; the disadvantages are the

requirement of using a compiler separate from OSLO and the extra effort that is necessary to integrate the DLL with OSLO.

Cross-reference components

A cross-reference operand component is simply a component whose value is that of a previous operand. The syntax of cross-reference components is simply O<*operand #*>, where <*operand #*> is the index of a previous operand.

Cross-reference operands provide a means for building complex operand definitions that cannot be formed with only two components. For example, suppose that you wish to create an operand whose value is the sum of the third-, fifth-, and seventh-order spherical aberrations. OSLO does not allow the definition of a three-component operand "SA3+SA5+SA7", so the definition must be split into two parts: an "intermediate" operand with zero weight and definition "SA3+SA5", and a "total spherical aberration" operand which is the intermediate operand, plus SA7:

```
*OPERANDS
 OP  MODE    WGT      NAME          VALUE  %CNTRB DEFINITION
O 1   M    1.000000                --        --  SA3+SA5
O 2   M    1.000000                --        --  SA7+O1
MIN RMS ERROR:       --
```

Operand 1 is given zero weight so that it is not incorporated directly into the error function value calculation. The component "O1" in operand 2 has the *value* of operand 1 (irrespective of the weight of operand 1).

Statistical operands

Statistical operands are used to compute averages and standard deviations or RMS values of groups of consecutive operands. There are two statistical operand components: AVE and RMS. The value of an AVE operand is the weighted mean of the values of the *first components* of all the subsequent operands, up to, but not including, the next RMS operand. The value of an RMS operand is the root-mean-square value of all the operands (including *both* components) between the preceding AVE operand and the RMS operand. If operand number *a* is an AVE operand and operand *r* is an RMS operand ($r > a + 1$), then the value of operand *a* is given by

$$f_a = \sum_{i=a+1}^{r-1} w_i c_{1i} \Big/ \sum_{i=a+1}^{r-1} w_i \tag{8.18}$$

and the value of operand *r* is given by

$$f_r = \left[\sum_{i=a+1}^{r-1} w_i f_i^2 \Big/ \sum_{i=a+1}^{r-1} w_i \right]^{1/2} \tag{8.19}$$

The statistical operands are often combined with cross-reference operands to compute standard deviations of groups of operands, as follows. If each of the operands between an AVE operand and the corresponding RMS operand is of the form "<component 1 of operand *i*>–O*a*", where O*a* represents a cross-reference to the AVE operand, the value of the RMS operand is the standard deviation of the group of operands. In mathematical notation, this is expressed as

$$f_r = \left[\sum_{i=a+1}^{r-1} w_i \left(c_{1i} - f_a\right)^2 \Big/ \sum_{i=a+1}^{r-1} w_i \right]^{1/2} \tag{8.20}$$

Example: Construct an error function to measure the RMS *y*-intercept in the image plane of three rays from field point (FBY = 1.0, FBX = 0.0).

First, set up the ray set and the operands as follows:

```
*RAYSET
FPT       FBY/FY1       FBX/FY2       FBZ/FX1       FYRF/FX2      FXRF/WGT  CFG/GRP
F 1       1.000000        --            --            --            --        -
         -1.000000      1.000000     -1.000000      1.000000      1.000000     -
RAY         TYPE          FY            FX            WGT
R 1       Ordinary     -0.500000        --          1.000000
R 2       Ordinary        --            --          1.000000
R 3       Ordinary      0.500000        --          1.000000

*OPERANDS
```

```
 OP  MODE    WGT      NAME     DEFINITION
O 1   M      --               AVE
O 2   M    1.000000           Y(1,1)-O1
O 3   M    1.000000           Y(1,2)-O1
O 4   M    1.000000           Y(1,3)-O1
O 5   M      --               RMS
```

Notice that the AVE and RMS operands both have weights of zero. If they had nonzero weights, they would have been included directly in the error function; here, the error function measures the standard deviation of operands 2, 3 and 4 only.

When operands are listed in the text output command using the **operands** (**ope**) command, AVE operands and operands between AVE and RMS operand pairs are hidden, and the weight and percent contribution displayed for the RMS operand is actually the sum of the weights and percent contributions of the operands between the AVE operand and the RMS operand.

```
*OPERANDS
 OP  MODE    WGT      NAME          VALUE  %CNTRB DEFINITION
O 5   M    3.000000              0.037095 100.00 RMS
MIN RMS ERROR:     0.037095
```

If the operands are listed with the **operands all** command, however, all operands are displayed and the RMS operand is displayed with zero weight:

```
*OPERANDS
 OP  MODE    WGT      NAME          VALUE  %CNTRB DEFINITION
O 1   M      --                 18.278045    --  AVE
O 2   M    1.000000              0.050719  62.31 Y(1,1)-O1
O 3   M    1.000000             -0.013752   4.58 Y(1,2)-O1
O 4   M    1.000000             -0.036967  33.10 Y(1,3)-O1
O 5   M      --                  0.037095    --  RMS
MIN RMS ERROR:     0.037095
```

Constant components

Constant operand components are simply fixed numeric values, such as –27.6 and 1.772×10^{-3}. The only restriction on constant components is that a negative constant cannot be used as the second component of an operand; for example, OSLO will not accept the operand definition "SA3>–0.075". Usually, this restriction can be worked around by revising the operand definition: for example, "SA3>–0.075" can be rewritten as either "–SA3<0.075" or as "–0.075<SA3", and the operand "DY(1, 1)*–20.0" can be rewritten as "–20.0*DY(1, 1)". For cases in which the operand cannot simply be rearranged into a legal form, cross-reference operands can be used; for example, "CMA3**–0.5" can be rewritten as "CMA3**O1", where operand 1 is defined as the constant value –0.5, with zero weight.

Error function construction

Measures of performance that might be included in the error function include the following:

- optical performance (e.g., *MTF*, RMS wavefront error, RMS spot size)
- physical realizability (e.g., no negative edge thicknesses)
- cost (materials, fabrication, etc.)

Typical error functions include terms to measure optical performance and physical realizability. Cost is usually controlled by limiting the types of materials used in constructing the lens system, limiting the number of elements in the system, limiting the use of unusual surface types, and so on. Cost is also determined by the manufacturing tolerances in the system. In this discussion, only optical performance and physical realizability will be considered.

The type of error function required for a particular design task of course depends on the type of system and the specifications to be met. In the early stages of a design, the best error function is often one built from paraxial ray data and aberration coefficients. The reasons for this are that (1) such an error function is very robust because it doesn't require that rays can be traced through the system, and (2) for a successful final design it is essential that the pupil locations, apertures, and general aberration balance be controlled. Thus, provided that the symmetry properties of the

system permit aberration analysis, it is often a good idea to carry out a preliminary design using aberration coefficients.

For final designs, error functions built upon exact ray tracing are usually required to obtain satisfactory performance. For simple systems having only a few degrees of freedom, only a few rays are typically required to produce a finished design. For more complex systems, the RMS spot size and RMS OPD measures of image quality both estimate the ability of a lens system to image points in object space to points on the image surface with good accuracy, and form the basis of typical error functions.

RMS spot size

The RMS spot size is estimated by tracing a number of exact rays through the optical system from one or more field points and measuring the standard deviation of the positions at which the rays intersect the image surface. An ideal system will focus all rays from any given field point in the field of view to a single point on the image surface and will therefore have a zero spot size. Let $X(\mathbf{h}, \lambda, \boldsymbol{\rho})$ and $Y(\mathbf{h}, \lambda, \boldsymbol{\rho})$ represent the x- and y-intercepts on the image surface of a ray in wavelength λ from fractional object coordinates $\mathbf{h}$ ($\mathbf{h}$ means the object position (h_x, h_y)) that passes through fractional entrance-pupil coordinates $\boldsymbol{\rho}$ ($\boldsymbol{\rho}$ means the pupil position (ρ_x, ρ_y)). In these terms, an expression for the estimated mean-square spot size, averaged over the field, is given by

$$S^2 \approx \sum_{i=1}^{\#\text{ field points}} \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} \left\{ \left[X\left(\mathbf{h}_i, \lambda_j, \boldsymbol{\rho}_k\right) - \bar{X}\left(\mathbf{h}_i\right) \right]^2 + \left[Y\left(\mathbf{h}_i, \lambda_j, \boldsymbol{\rho}_k\right) - \bar{Y}\left(\mathbf{h}_i\right) \right]^2 \right\} \tag{8.21}$$

Here, w_{ijk} is the (normalized) weight of the ray, and the coordinates of the centroid of the spot from field point $\mathbf{h}$, are given by

$$\bar{X}\left(\mathbf{h}_i\right) \approx \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} X\left(\mathbf{h}_i, \lambda_j, \boldsymbol{\rho}_k\right) \Bigg/ \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} \tag{8.22}$$

$$\bar{Y}\left(\mathbf{h}_i\right) \approx \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} Y\left(\mathbf{h}_i, \lambda_j, \boldsymbol{\rho}_k\right) \Bigg/ \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} \tag{8.23}$$

To construct an error function that estimates the spot size, a scheme must be used to determine the *sampling* to be used (i.e. the number of field points and their positions in the field, the number of rays to be traced from each field point and their positions in the pupil, and the number of colors and their wavelength values) and the values of the weights. To this end, OSLO provides a set of methods based on Gaussian integration for automatically selecting the sample locations and weights and constructing an error function. This scheme is described in the "Automatic error function generation" section below.

RMS OPD

Another characteristic of optical systems that produce sharp point images is that the shape of the wavefront emerging from the system for a given field point is that of a sphere centered about the point where a reference ray from the field point intersects the image surface. The imaging quality of a system can then be measured by calculating the deviation from this *reference sphere* of the emerging wavefront (see p.196).

The optical path difference, or OPD, for a single ray from a given field point is the distance along the ray from the reference sphere to the wavefront, times the refractive index in image space. To measure the imaging performance of the system as a whole, we create an error function that measures RMS OPD, averaged over the field, as follows. Let $d(\mathbf{h}, \lambda, \boldsymbol{\rho})$ represent the OPD of a ray in wavelength λ from fractional object coordinates $\mathbf{h}$ that passes through fractional entrance-pupil coordinates $\boldsymbol{\rho}$. An expression for the estimated mean-square OPD, averaged over the field, is then given by

$$D^2 \approx \sum_{i=1}^{\#\text{ field points}} \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk}\left[d\left(\mathbf{h}_i,\lambda_j,\mathsf{r}_k\right)-\bar{d}\left(\mathbf{h}_i\right)\right]^2 \tag{8.24}$$

Again, w_{ijk} is the (normalized) weight of the ray, and the average OPD of the rays from field point **h** is given by

$$\bar{d}\left(\mathbf{h}_i\right) \approx \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} d\left(\mathbf{h}_i,\lambda_j,\mathsf{r}_k\right) \Bigg/ \sum_{j=1}^{\#\text{ colors}} \sum_{k=1}^{\#\text{ rays from field point } i} w_{ijk} \tag{8.25}$$

As with the estimation of the mean-square spot size, a scheme must be used to determine appropriate field, wavelength, and pupil samples and the appropriate weights. The Gaussian integration schemes available for this purpose in OSLO are described on the next page.

MTF

The modulation transfer function, or MTF, is often the specified measure of performance for imaging systems. However, the MTF has drawbacks that limit its use in error functions. First, it is computationally expensive. Second, the MTF is often a highly nonlinear function of the optimization variables, making it ill-suited to least-squares optimization. The nonlinearity also makes accurate numerical estimation of the derivatives of the MTF difficult. For these reasons, error functions used in optimization generally measure performance using quantities that correlate well with the MTF, such as RMS spot size or RMS OPD.

Notwithstanding the above, it is feasible to use MTF for *differential* optimization. In this case, a system is first optimized using a spot-size or OPD error function. Then a new error function based on MTF is used to incrementally change the MTF to meet target specifications.

Automatic error function generation

For routine optical design tasks, it is possible to generate an acceptable error function automatically, or at least with only general input parameters provided by the designer. OSLO contains two automatic error function generators, commonly known as the GENII error function generator and the OSLO error function generator because of their historical implementation. A knowledge of lens design is used in the construction of the GENII error function, which is why it is sometimes called an expert designer error function. The OSLO error function, on the other hand, makes no use of lens design knowledge, and sets up a mathematical function to optimize the rms spots size or rms OPD.

Genii error function

The GENII error function, so-called because it was originally used in the GENII program, uses multiple items of data from each traced ray to build a compact error function that is well-suited to interactive design. It uses only 10 rays to derive a 31-term error function that makes use of the relationships between classical aberrations and exact ray data. The individual terms are normalized so that a value of 1.0 represents a normal tolerance for the term, making it easy to see the significant defects in a design and apply appropriate weighting, if necessary. The user only needs to specify the spatial frequency at which the system is to be optimized, which makes it easy to use.

The GENII error function is designed to handle systems of moderate complexity, such as camera lenses and other systems having three to eight elements. It is not sufficiently comprehensive to handle very large systems, and is not efficient for handling singlets or doublets. The following description, extracted from the OSLO program documentation[1], describes its general construction.

The GENII error function uses terminology somewhat different from the rest of OSLO. The GENII error function ϕ is constructed from *targets* and is defined by

$$\phi = \sum_{j=1}^{N} \left\{ \frac{A_j - D_j}{T_j} \right\}^2 \tag{8.26}$$

where A_j is the actual value (the value for the existing lens) of the j^{th} target and D_j is the desired value (the value for the program to work to) for the j^{th} target. T_j is the tolerance of the j^{th} target (for example, the acceptable amount that the j^{th} target is permitted to deviate from its desired value in either direction). N is the number of targets. $A_j - D_j$ is the amount that the j^{th} target is in error and $(A_j - D_j)/T_j$ is the error measured in tolerances, so $(A_j - D_j)/T_j$ is the number of tolerances that the j^{th} target deviates from its desired value. In other words, $1/T_j$ is a weighting factor on the error $A_j - D_j$. M is the weighted sum of the squared errors. The use of tolerances simplifies the interpretation of the error function by effectively establishing a common unit of measure for different types of operands.

The default error function from GENII has a consistent set of targets and tolerances to control classical aberrations, i.e., it is assumed that the lens is rotationally symmetric. Color correction is performed using Conrady *D–d* operands and there is no control on secondary color. Since it is designed to balance aberrations in a focal plane shifted from the paraxial image plane, the image distance should be allowed to vary during optimization. If the lens is capable of being diffraction limited, this error function can usually drive it there. If the lens is not capable of being diffraction limited, a reasonable aberration balance can be achieved. If the lens is *f*/1.5 or faster, this error function may not work well because too few rays are traced.

The rays used in the error function are selected on the assumption that there will be some vignetting. If there is no vignetting, the off-axis rays should be moved further out in the aperture. For some lenses, better correction may be achieved by moving the axial marginal ray in to 0.9 or 0.95, rather than 1.0.

The GENII error function creates a field points set with three entries (FBY = 0.0, 0.7, and 1.0) and a ray set with 8 rays. Ray 1 (a real chief ray) is used to compute field curvature and distortion operands. Ray 2 is used for the axial marginal ray. Rays 3, 4, and 5 are the aperture rays for the 0.7 field point and rays 6, 7, and 8 are the aperture rays for the 1.0 field point. The aperture coordinates for these rays may need to be adjusted based on the desired vignetting.

The GENII error function command in OSLO generates a set with 43 operands, of which only 31 have non-zero weight. These 31 operands comprise the error function. The other operands (with zero weight) are used as intermediate steps in forming GENII-style target definitions. All tolerances are computed from the specified frequency (*design_spatial_frequency*) and the exit angle of the paraxial axial ray, which is held at its initial value. The basic tolerance from which the others are computed is the tolerance on the transverse ray error for the on-axis marginal ray. This tolerance, Dy, is set at .167 times the reciprocal of the design spatial frequency. The active operands are described in the table below.

Description of Active Operands	Field	Tolerance
Exit angle of paraxial axial ray, *u´*		0.0001
Focus shift penalty		3*Dy*
Marginal transverse ray error on-axis	0.0	*Dy*
Marginal *OPD* on-axis		*u´Dy*/3
Marginal *DMD* for axial color		*u´Dy*/3
Percent distortion		1
Tangential field curvature (transverse measure)		3(0.7)*Dy*
Sagittal field curvature (transverse measure)	0.7	3(0.7)*Dy*
Primary aperture coma exact in field		3.2*u´Dy*
Transverse ray error in upper aperture		4(0.7)*Dy*
OPD in upper aperture		*u´Dy*/3
DMD for lateral color in upper aperture		*u´Dy*/3
Transverse ray error in lower aperture		4(0.7)*Dy*
OPD in lower aperture		*u´Dy*/3
DMD for lateral color in lower aperture		*u´Dy*/3
x component of transverse ray error for sagittal ray	0.7	4(0.7)*Dy*
y component of transverse ray error for sagittal ray		*Dy*
OPD on sagittal ray		*u´Dy*/3
Percent distortion		1
Tangential field curvature (transverse measure)		3*Dy*
Sagittal field curvature (transverse measure)		*Dy*
Primary aperture coma exact in field		3.2*u´Dy*
Transverse ray error in upper aperture		4*Dy*
OPD in upper aperture	1.0	*u´Dy*/3
DMD for lateral color in upper aperture		*u´Dy*/3
Transverse ray error in lower aperture		4*Dy*
OPD in lower aperture		*u´Dy*/3
DMD for lateral color in lower aperture		*u´Dy*/3
x component of transverse ray error for sagittal ray		4*Dy*
y component of transverse ray error for sagittal ray (coma)		*Dy*
OPD on sagittal ray		*u´Dy*/3

OSLO error function

The OSLO error function takes a mathematical approach to computing the spot size or OPD of an optical system, averaged over the field of view and chromatic range of interest. One can imagine tracing a very large number of rays from a large number of field points, in a large number of wavelengths, to find the overall performance of the system. As the number of rays in the error function approaches infinity, the sum becomes an integral. Although feasible error functions obviously cannot contain an infinite number of rays, the integral formulation recasts the problem of determining the spot size into one of numerically evaluating an integral, which leads to efficient schemes for selecting the proper rays to include in the error function.

Consider a rotationally symmetric system working at wavelength λ that images objects at heights h. Consider points in the pupil of such a system to be specified in cylindrical coordinates (ρ, θ). Suppose that ρ and h are fractional coordinates. The height of a ray on the image surface of such a system will depend on all of these quantities: $y = y(\rho, \theta, \lambda, h)$ and $x = x(\rho, \theta, \lambda, h)$. The error function, defined as the mean-square spot size averaged over the pupil, field of view, and wavelength, can then be written as an integral.

Let us consider here only the integration over aperture; the integration over field and wavelength will follow the same principles. At a single field point, the integral for the y-component of the spot size becomes

$$\phi = \int_0^{2\pi} \int_0^1 \left[y(\rho, \theta, h) - \bar{y}(h) \right]^2 \rho \, d\rho \, d\theta \tag{8.27}$$

The evaluation of this integral must, of course, be done numerically instead of analytically for practical systems. The task of choosing appropriate sampling points for the integral is equivalent to creating an appropriate ray set. Forbes pointed out that Gaussian quadrature methods are well suited to solving this type of problem.

Gaussian quadrature, in addition to being an efficient way to do numerical integration, has a property that is intuitively helpful to optical designers: It gives an exact solution to the integral

$$\int_0^1 \left(a_1 \rho + a_3 \rho^3 + \cdots + a_{2n-1} \rho^{2n-1} \right) d\rho \tag{8.28}$$

using n sampling points. This implies, for example, that if we have an on-axis optical system that is known to have negligible ninth order (or higher) spherical aberration, the rms spot size can be calculated *exactly* using just four rays!

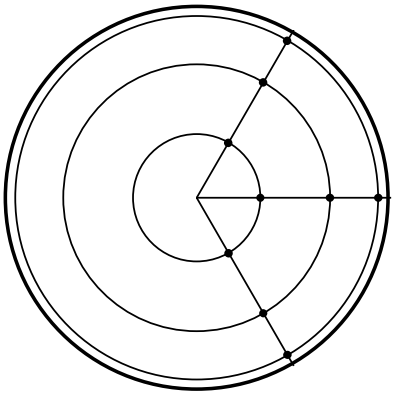
To evaluate Eq. (8.27) requires integrating over both ρ and θ. The angular part should be sampled at angles $\theta_k = (k - \tfrac{1}{2})\pi / N_\theta$, where N_θ is the number of angular sampling points in the range $(0 \ldots \pi)$. For the radial part, the sampling points are selected using Gaussian integration. The aperture integral is then written as the summation

$$\phi \approx \sum_{j=1}^{N_\rho} w_j \sum_{k=1}^{N_\theta} \left[y(\rho_j, \theta_k) - \bar{y} \right]^2 \tag{8.29}$$

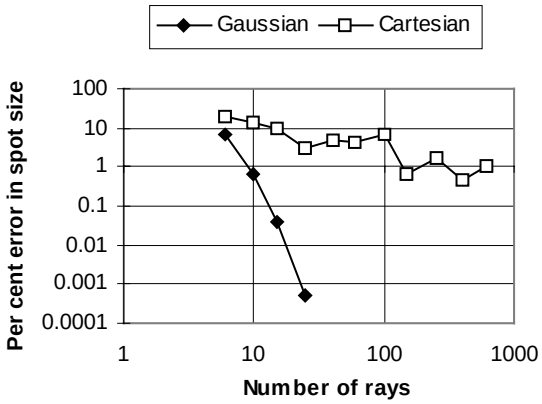
The required sampling points (ρ_j, θ_k) are shown in the following table for the first few values of N_ρ and N_θ.

N_ρ, N_θ	j, k	ρ_j	w_j	θ_k
1	1	0.7071	0.5000	90
2	1	0.4597	0.2500	45
	2	0.8881	0.2500	135
3	1	0.3357	0.1389	30
	2	0.7071	0.2222	90
	3	0.9420	0.1389	150
4	1	0.2635	0.0870	22.5
	2	0.5745	0.1630	67.5
	3	0.8185	0.1630	112.5
	4	0.9647	0.0870	157.5

The figure below shows a typical ray pattern for Gaussian quadrature with three *rings* and three *spokes*. Note that rays need to be traced through only one-half of the aperture, because of the rotational symmetry of the system. Note also that all rays are skew rays; there are no rays along the *y* axis.



The figure below shows typical accuracy of Gaussian quadrature, as compared to Cartesian (i.e. square grid) methods for evaluating the rms spot size. It should be noted, however, that it is necessary to have a circular entrance pupil to achieve this accuracy, and also the method is restricted to systems that have rotational symmetry. If these factors are missing, Cartesian methods compare more favorably to Gaussian quadrature.



For optical design purposes, the Gaussian quadrature method is not totally satisfactory, because it does not produce a sample point at the center of the aperture, where the reference ray is traced, nor at the edge of the aperture, where the marginal ray is traced. Two other quadrature schemes called Radau and Lobatto quadrature produce the desired sampling. The Radau scheme includes the ray at the center of the aperture, and the Lobatto scheme includes both the central and marginal rays.

The automatic error function in OSLO gives the user the choice of a square grid or Gaussian, Radau, or Lobatto quadrature. Lobatto quadrature is the default, since it is most convenient for optical analysis. Integration is carried out over aperture, field, and wavelength. The program allows you to specify the number of rings and spokes, as well as to choose either rms spot size or rms wavefront error. An option is provided to align one of the spokes with the *y* axis, so that meridional rays are traced.

In addition to the options for setting up ray patterns, the automatic error function generator allows a choice of color correction methods between tracing rays in each of the defined wavelengths, or using a simplified methods based on Conrady *D–d* operands. Separate operands can be generated to control distortion and edge thickness, which are of course not included in the Gaussian integration scheme.

In a system where the aperture is vignetted, the quadrature schemes must be modified to approximate the actual pupil by an ellipse defined by the vignetting factors included in the field point set. The circular pupil is mapped on the vignetted ellipse to determine the final ray coordinates used in the error function computation.

The automatic error function generated by OSLO can be classified as a high-accuracy, medium efficiency error function. For typical systems, it is much more accurate than error functions based on a square grid of rays. On the other hand, compared to custom error functions that have been used in the past for optimization, it uses too many rays. If the highest efficiency optimization is desired, it is usually possible for a skilled designer to create an error function that surpasses the OSLO default. It is interesting to note, however, that few designers choose this option anymore, probably because the high speed of contemporary computers makes it unnecessary.

Multiconfiguration optimization

A multiconfiguration system is one in which a portion of the lens data changes from one configuration to the next. The zoom lens is the most common multiconfiguration system, but there are several other types, including lens attachments, systems in which the ray paths vary from one configuration to the next, and even systems that are apparently single configurations, but which are optimized simultaneously under different operating conditions. Multiconfiguration optimization refers to the process of optimizing a system so that its performance in any one configuration is not optimum, but the performance of the ensemble of configurations is optimum.

OSLO has a special data structure for storing multiconfiguration data in a binary format that can be switched extremely rapidly. The base configuration is referenced as configuration 1. There is no upper limit on the configuration number. If a configuration is referenced that has no defined configuration data, the data for the system is the same as the base configuration.

Ordinary configuration data (radii, thickness, apertures, and glasses, as well as basic paraxial operating conditions) can be entered directly on the surface data spreadsheet by setting the current configuration field to the desired configuration. Special configuration data is entered using the configuration data spreadsheet.

At any given time, the system is in its current configuration, which can be set by the user, or automatically by the program. Lens setup is always performed in the base configuration, and solves are normally only carried out in this configuration (this can be overridden by an operating condition). Certain actions, e.g. opening the surface data spreadsheet or computing the operands, cause the program to automatically reset to the base configuration.

Variables can be specified in any configuration, and can also be specified to be in configuration 0, meaning that they are variable, but have the same value in all configurations (these are sometimes called global variables).

Operands in a multiconfiguration system must reference the configuration in which they are to be evaluated; there are no global operands. If the configuration number for an operand is not specified, it is understood to apply to the base configuration.

Optimization of multiconfiguration systems proceeds similarly to optimization of single configuration systems, the only difference being that the program cycles through all the defined configurations when computing the error function and setting up the derivative matrix.

Global optimization

The most widely employed optimization scheme consists of the selection of a starting point by the designer, followed by the application of the DLS method to locate a nearby optimum. If the local optimum thus found is not of sufficient quality, a new starting point is chosen and the process is repeated until a satisfactory local minimum is found. The success of such a scheme is dependent upon the choice of the starting point, and the resulting designs are typically similar in form to the starting point.

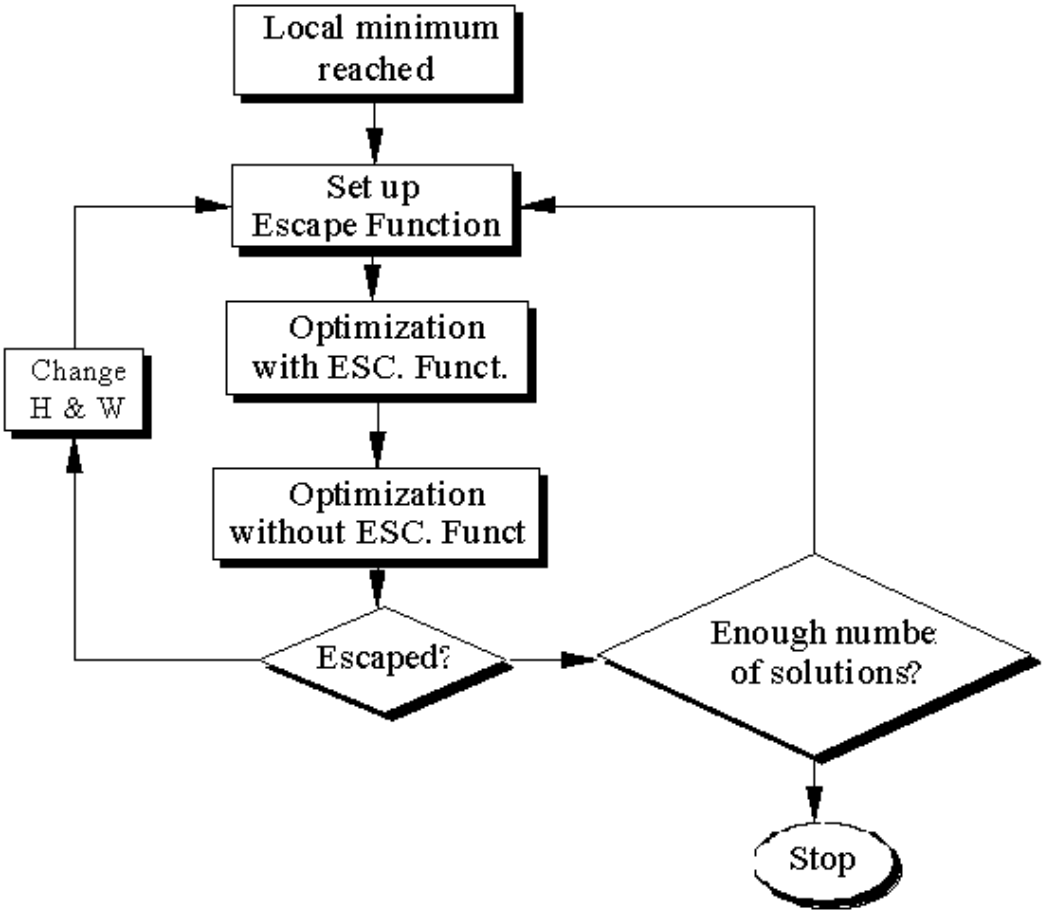
The goal in the optimization stage of design is to determine the configuration of the lens system for which the merit function is smallest over the region of interest defined by the constraints–that is, to locate the *global minimum* of the merit function. OSLO Premium contains two optimization methods useful for this task: Global Explorer (GE) and Adaptive Simulated Annealing (ASA). Strictly speaking, GE should be called a multiple solution generator, while ASA is a true global optimization method. The difference between the two is that GE carries out successive local optimizations, keeping track of several solutions. The outcome depends on the starting point, and the solutions will be the same if the method is run several times. ASA, on the other hand, produces (ultimately) only a single solution which is the best estimate of the global optimum in the specified search region, subject to the time constraint imposed on the optimization.

One conceptually simple global optimization scheme is the *grid search*: the merit function ϕ is evaluated at points on a regular grid aligned to the coordinate axes, and the sample with the lowest value of ϕ is taken as an estimate of the global minimum. Such a simple scheme, however, rapidly becomes unworkably inefficient with increasing dimensionality. Consider a problem in which five samples are required in each coordinate direction to locate the global minimum to the desired accuracy and in which the merit function requires 1 μs to evaluate. Thus, the time needed to perform a grid search in N dimensions is 5^N μs: for $N = 10$, the search is completed in less than ten seconds, but for $N = 15$, more than eight hours is needed, and in just 20 dimensions, over *three years* is required. Clearly, even with such sparse sampling and with such unrealistically favorable estimates of evaluation speed, a grid search is impractical; the sampling of the region of interest in the configuration space must be more selective.

Global Explorer

The solution obtained from damped least squares is no more than a local minimum that happens to be near the starting design. Once the design is trapped there, it is impossible to get out of that place, because damping factor becomes too large around the local minimum and this prevents the design from jumping out of the trap. This is one of the most serious defects of the DLS method.

The Global Explorer method developed by Isshiki defines an escape function (or penalty function) that forces the DLS algorithm out of a local minimum. The method is illustrated in the following a flow chart and also by the following description of steps.



1. When the design falls into a local minimum, the program automatically sets up an escape function there, in which initial values are given for *H* and *W*.
2. Optimization is performed for the merit function including the escape function.
3. After removing the escape function, optimization is done again, the solution thus obtained is another local minimum of the merit function.
4. If the newly found solution is not identical with any of the already found ones, the escape was regarded as a success, and that solution is saved in the file.
5. When the escape is not successful, two escape parameters *H* and *W* are changed by a predetermined rule, and the process (2) to (4) is repeated until the new and independent solution is found.

In the step (4), there must be a criterion to judge whether the escape was successfully made or not. In Global Explorer, the distance of two solutions is defined as

$$D_p = \sqrt{\sum_j w_j (x_j - x'_j)^2} \tag{8.30}$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are positions of the local minima in the parameter space. If D_p is larger than a threshold value D_t, these two solutions are regarded as independent. If this relation does not hold between a newly found solution and each of the already filed solutions, the escape is judged as a failure.

The following figure illustrates a model of merit function ϕ having a design parameter x_j. When the design falls into a local minimum at x_{jL}, an escape function f_E is set up there which is to be added to the error function. The escape function is defined by

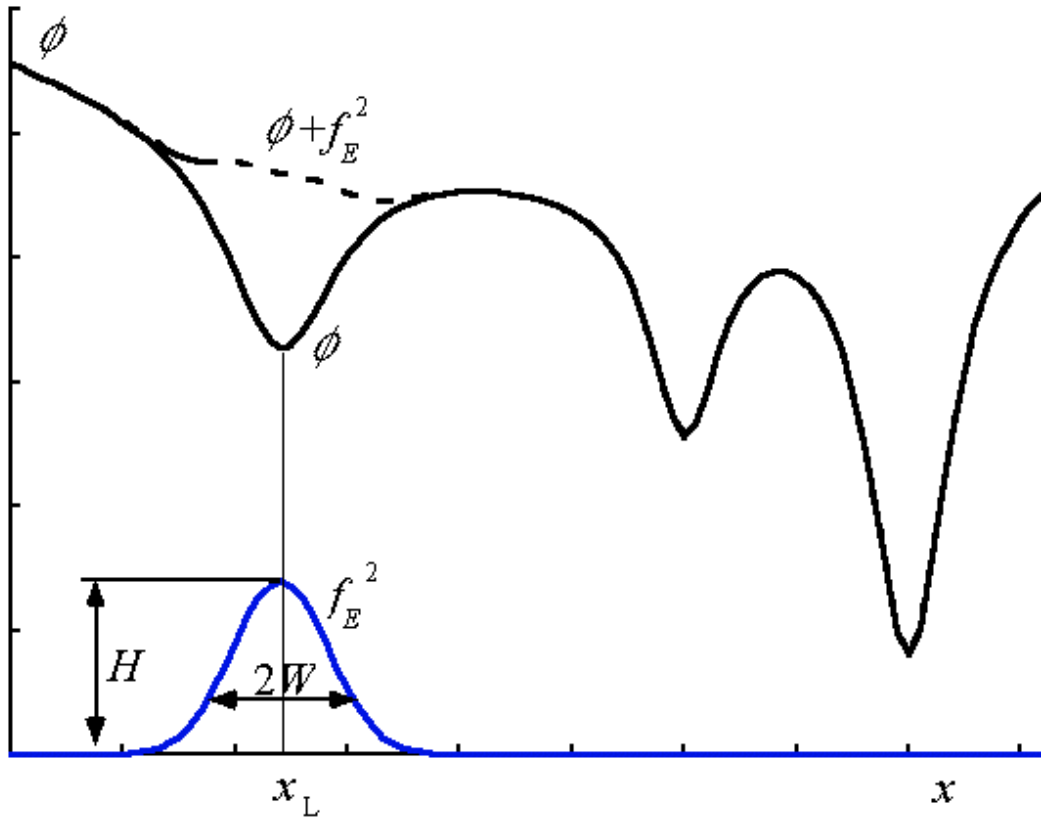
$$f_E = \sqrt{H}\exp\left\{-\frac{1}{2W^2}\sum_j w_j\left(x_j - x_{jL}\right)^2\right\} \tag{8.31}$$

where

x_{jL}: Local minimum from which the design is to escape.

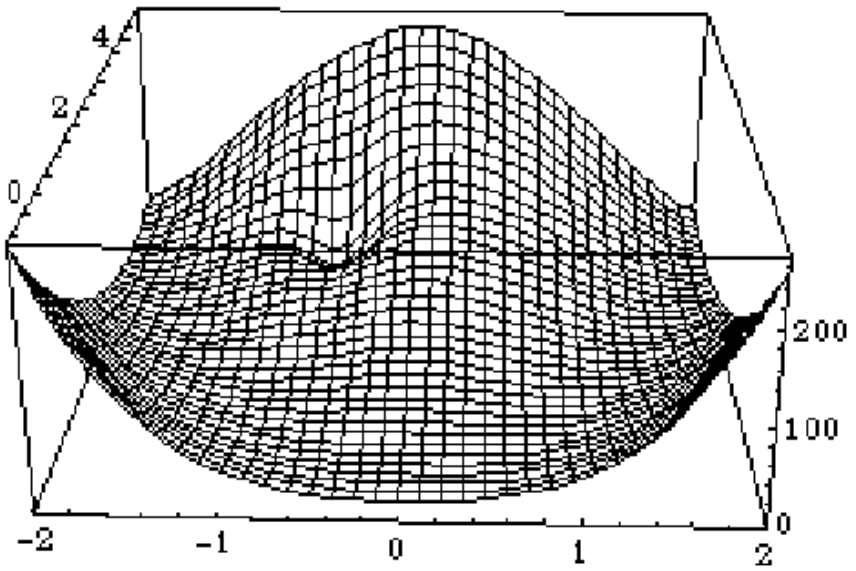
w_j: Weights for design parameters.

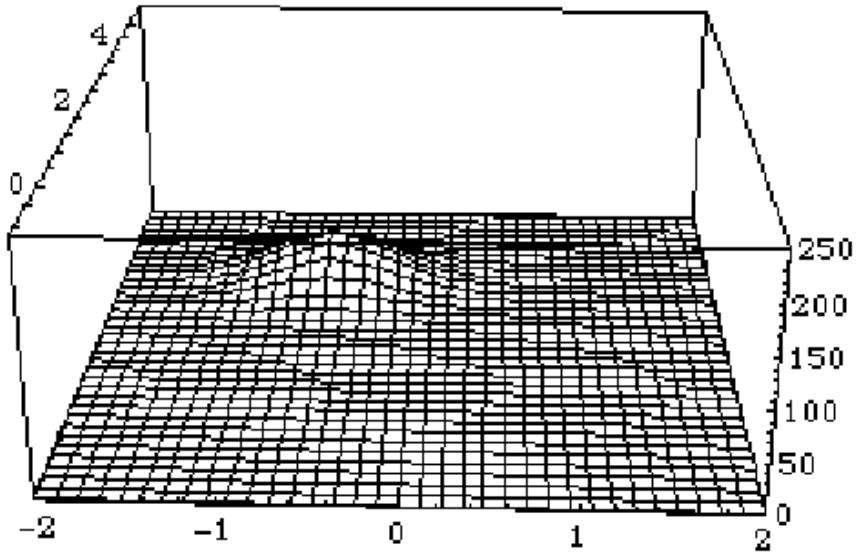
H and W: Escape parameters as shown in figure below.

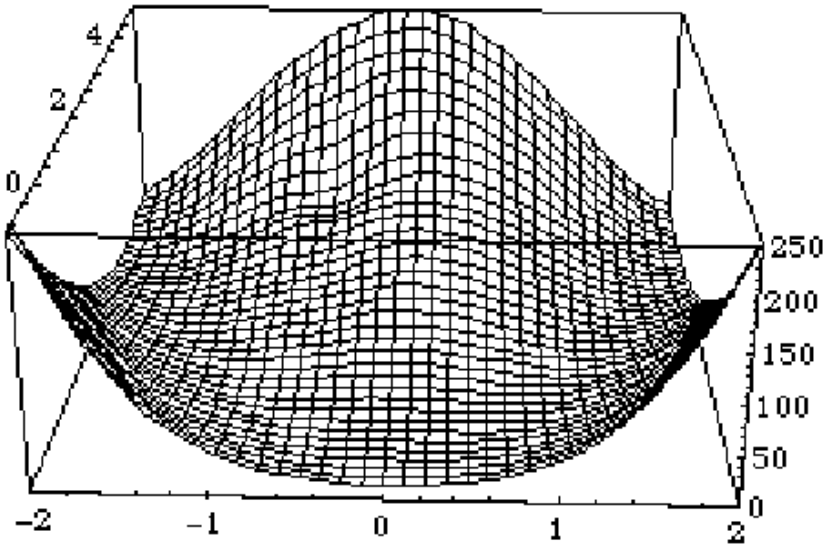


The shape of merit function ϕ around its local minimum changes with the escape function (see figure); ϕ is raised by an amount of f_E^2 when an escape function is added to the error function. This enables the design to escape from the trap. Repeating this process, you can automatically find a predetermined number of local minima. In the next section, the program named 'Global Explorer' is explained in detail.

From the above graph, it may look very difficult to select appropriate values for the two parameters H and W. However, in practical cases where the number of parameters is large enough, this problem is not so delicate; a rather crude choice of these two values would be acceptable in most cases. In the above figure, the number of parameters is only one, i. e. the picture shows a model in one dimensional parameter space. A model in two dimensional space is shown below.


4
2
0
200
100
0
−2
−1
0
1
2

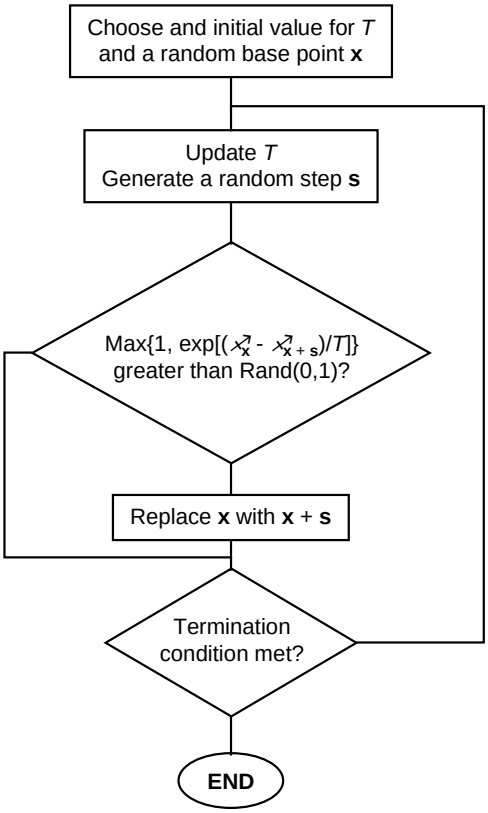

4
2
0
250
200
150
100
50
0
−2
−1
0
1
2


4
2
0
250
200
150
100
50
0
−2
−1
0
1
2

Adaptive Simulated Annealing (ASA)

Since the performance of global optimizers is essentially independent of the particular starting point, they are especially effective in situations in which suitable starting points are difficult to determine. One attractive global optimization scheme is the *simulated annealing* algorithm. Simulated annealing belongs to a class of global optimizers that are called *controlled random search* methods because the merit-function space is sampled randomly according to a scheme in which the values of several parameters determine the distribution of the random samples. Although these so-called *Monte Carlo* methods might at first seem as promising as throwing darts while blindfolded, it turns out that they are much more efficient than grid search. Monte Carlo methods are routinely used for solving multidimensional integrals, for example.

The simulated annealing algorithm derives its name from the fact that its behavior is controlled principally by a parameter *T*, called "temperature," that is analogous to the temperature in the thermal annealing process. We call simulated annealing a "true" global optimization algorithm because each run attempts to search the entire region of interest for the global minimum rather than performing multiple downhill optimization runs in which the selection of the various starting points is automated.



In simulated annealing, the optimization process is not required to proceed uniformly downhill, but is allowed to make occasional uphill moves. The typical increase in ϕ that is acceptable in an uphill move is determined by the value of *T*. At the start of the annealing process, *T* has a relatively large value compared to the standard deviation of the merit function over the region of interest, and the random walk effectively traverses all of this region. As the random walk progresses, *T* is lowered, the allowed increases in ϕ are then typically smaller, and the walk is effectively constrained to ever lower valleys. If *T* is reduced sufficiently slowly, the random walk can escape the higher valleys during its earlier stages, and it can be expected to terminate at the global minimum. If *T* is lowered more quickly, however, the random walk is more likely to become trapped in one of the higher valleys. This follows the analogy with the thermal annealing process: if a hot solid is cooled slowly, its final state is likely to be one of lower potential energy

(usually, a more ordered arrangement of atoms); if it is cooled more quickly, the final state is likely to be one of higher potential energy.

There exist numerous variants of simulated annealing, each of which represents an attempt to address the following concerns:

- How is the initial value of T chosen, and how is T lowered?
- How are the steps generated?

Most variants of simulated annealing require the user to answer these questions by providing the values of a set of parameters that control the course of optimization; for example, for the temperature, the user may be required to specify an initial value, a factor between 0 and 1 by which T is multiplied periodically, and a parameter used to control the length of this period. The parameters for tuning the step generation process are even more important. Determining appropriate values for all of these parameters is often accomplished through trial and error, which is prohibitively expensive for all but the simplest problems. Furthermore, many variants of annealing do not possess the desirable properties of invariance under linear transformations of the coordinates or of the merit function.

For these reasons, we have developed adaptive controls for all of the parameters in simulated annealing. The performance of the resulting algorithm, which we call Adaptive Simulated Annealing (or just ASA), is invariant under linear transformations of both the merit function and, more importantly, of the coordinates in the configuration space. ASA requires the specification of only a single parameter: the annealing rate written as ε, which determines the average rate at which T is lowered. Given a value for this parameter, all else is automatically tuned to the particular problem at hand. Reducing the value of ε slows the cooling (hence increases the run time) and makes the search for the global minimum more thorough.

The effectiveness of any adaptive mechanism is crucially dependent upon the automatic control of the statistics of the random step generator. First, a fundamental requirement of simulated annealing is that the step distribution must be symmetric. That is, a step $\mathbf{s}$ is just as likely as the step $-\mathbf{s}$. We have chosen to use a Gaussian distribution for the steps. In two dimensions, a general Gaussian distribution resembles an elliptical cloud that becomes less dense away from its center. The proportions, orientation, and size of the ellipse are the key parameters in this case. It is clear, that, as the value of T is lowered, the region being explored in the configuration space is reduced. It therefore seems reasonable to expect that the optimal Gaussian for the steps should also be modified as the annealing process advances. If the step size is too large for the current value of T, almost all of the attempted moves are rejected and the process stagnates. On the other hand, if the steps are too small, the walk may cover only a fraction of the configuration space that should be explored at the current temperature for optimal efficiency. It also appears reasonable to expect that the relative shape and orientation of the Gaussian cloud will need to be adjusted continually in order to maintain optimal efficiency. We have proposed a simple procedure that automatically adjusts the scale, shape, and orientation of the n-dimensional Gaussian at each stage during annealing.

The basic idea for step control in ASA is based on what is called the *central limit theorem*. This theorem states that, if you average a collection of independent random numbers, the result is a random number with a distribution that is roughly a (one-dimensional) Gaussian. This holds regardless of the distributions of each of the individual random numbers and it also follows that the variance (i.e., the mean-square spread) in the resulting random number is just the average of the variances of the individual random numbers. As a result, it turns out that a Gaussian multidimensional step generator can be realized by taking linear combinations of a collection of random vectors.

In ASA, the idea is to keep a record of the last m steps that have been accepted and to generate new steps by taking random linear combinations of these steps and scaling the result by an appropriate (dynamically adjusted) expansion factor. In this way, the statistics of the generated steps are directly coupled to both the behavior of the merit function and the current value of T. This scheme turns out to be not only effective but simple: without expensive matrix operations, the algorithm automatically adjusts to the multidimensional terrain in a way that is invariant under

arbitrary linear transformations that scale and stretch the space in a manner that warps rectangular prisms to rotated, elongated parallelepipeds. Within the context of lens design, this invariance is of utmost importance since there is no natural measure of distance in the configuration space. (Recall that the coordinates are normally a mixture of refractive indices, thicknesses, curvatures, aspheric coefficients, etc., and there is no intuitive way to define the distance between two points in such a space.)

Another important aspect of the guided random walk is the means for dealing with steps that take the system outside the region of interest. That is, if one or more constraints are violated by the proposed system, the step cannot be accepted. Efficiency is enhanced if, instead of simply rejecting such a system and requesting a new random step, the point at which the current step first crosses a constraint boundary is found and the step is then reflected off this constraint back into the region of interest. Without this reflection process, the regions near constraints turn out to be undersampled and this can adversely affect the overall efficiency.

For optimal efficiency it is important that the details of this process of reflection take a form that ensures invariance under linear transformations of the coordinates. Since our intuition is based in a space where there is a natural measure of distance, it seems unambiguous to state that a step should be reflected in a given plane as if it were a mirror. This is not so, however. The difficulty can be appreciated by observing that the idea of a normal to a plane surface is not invariant under linear transformations. To see this, consider two perpendicular lines on a page and imagine a scale transformation that stretches the plane along a direction that is not parallel to either of the lines. As a result, the lines will no longer be perpendicular. It follows that, to complete the specification of the reflection process, it is necessary to introduce a particular measure of distance – i.e., a "metric" – to the coordinate space.

In ASA, there is a natural metric defined by the elliptical Gaussian cloud of the step distribution: the distance between two points can be measured in terms of the number of steps (of mean size in that direction) required to move from one point to the other. Notice that, due to the coupling to the step distribution, the form of this metric evolves during the annealing process. Now, if the space is redrawn by reference to this metric (which amounts to a linear transformation) the elliptical Gaussian cloud takes the form of a sphere and this gives the natural representation in which to bounce off constraints as if they were mirrors. With this, the essential pieces of the crucial step-generation component of ASA are completely determined.

The details of the temperature control aspects of ASA are more sophisticated, but it is intuitive that, for a given merit function, there will be particular ranges for the value of T during which a relative reduction in the rate of decrease of T will lead to better overall efficiency. This aspect of the adaptiveness in ASA is controlled by monitoring certain statistics of the recent stages of the search. The process is terminated when the typical relative change in the current merit function value falls below a certain level. On any given problem, ASA should be run as many times as is workable and the designs that emerge can then be reoptimized and fine-tuned by using the **iterate full** or **iterate standard** commands.

The general outline presented here should give some appreciation of the philosophy underlying the design of ASA(1). It is important not to forget that typical design problems are challenging tasks and the success of an automated algorithm is still dependent upon the designer's guidance. No longer is the designer asked for a starting point, but now the designer is asked for the specification of the region to be explored.

---

1 A. E. W. Jones and G. W. Forbes, Journal of Global Optimization **6**, 1-37 (1995)

Chapter 9 Tolerancing

An optical design is not ready for manufacturing until tolerance limits have been designated for each construction parameter of the optical system. These construction parameters are all of the data that are used to specify the system: radii of curvature, element thicknesses, air spaces, refractive indices, etc. Tolerance schemes range from computing sensitivity information, to statistical analysis techniques, to simply specifying limits that have been successful in the past or have been prescribed by another source. The methods used vary from designer to designer and from design to design. In this chapter, we will look at tolerance analysis from the point of view of characterizing system performance.

Default tolerances

If you display the current tolerance values for any lens, you will see that OSLO has assigned default tolerances to certain construction items. These tolerance values are taken from the ISO 10110 standard. The standard specifies tolerances that will be assumed if specific values are not indicated on a drawing. If you do not enter any overriding tolerance data, OSLO will assign these default tolerances to your lens. You are free, of course, to change some or all of these tolerance assignments. The assigned default tolerances are a function of the aperture of the surface, as shown in the table below. Note that for the two larger aperture classes, the surface form (designated 3/ and given in fringes) tolerances are given for a test diameter that is smaller than the aperture of the surface. OSLO will compute the appropriate change in the surfaces based on fringes measured over the test diameter, not the aperture diameter, if the default tolerance is used in these cases.

Depending on your particular optical performance requirements, it is more or less likely that you will find these default tolerances to be appropriate for the optical system that you are evaluating. The defaults do, however, provide convenient starting points for examining the relative sensitivities of the various construction parameters of the lens. As is the case with any parameter that is assigned by default, it is up to you to make sure that the default value is acceptable for your particular circumstances.

Property	**Maximum dimension of part (mm)**			
	Up to 10	**Over 10 Up to 30**	**Over 30 Up to 100**	**Over 100 Up to 300**
Edge length, diameter (mm)	± 0.2	± 0.5	± 1.0	± 1.5
Thickness (mm)	± 0.1	± 0.2	± 0.4	± 0.8
Angle deviation of prisms and plate	± 30′	± 30′	± 30′	± 30′
Width of protective chamfer (mm)	0.1 – 0.3	0.2 – 0.5	0.3 – 0.8	0.5 – 1.6
Stress birefringence (nm/cm)	0/20	0/20	–	–
Bubbles and inclusions	1/3x0.16	1/5x0.25	1/5x0.4	1/5x0.63
Inhomogeneity and striae	2/1;1	2/1;1	–	–
Surface form tolerances	3/5(1)	3/10(2)	3/10(2) 30 mm diameter	3/10(2) 60 mm diameter
Centering tolerances	4/30′	4/20′	4/10′	4/10′
Surface imperfection tolerances	5/3x0.16	5/5x0.25	5/5x0.4	5/5x0.63

Statistics background

In the usual course of design and analysis, we are interested in what the construction data for our system should be. The radius of the third surface is 78.25 mm; the axial thickness of the second element is 3.5 mm, etc. When we get around to building our optical system however, we must deal with the fact that we cannot make our parts with the precision that these numbers might imply. In a batch of lenses, we'll find that the radius of the third surface is 78.21 in one case, 78.38 in another, and 79.1 in a third. The obvious question is: how close to the nominal value of 78.25 do we have to be in order to still satisfy our performance requirements? We will try to answer this question statistically, i.e., the radius of any given surface 3 is random. This may seem a hopeless proposition, but we will also assume that we know something about the statistical distribution of radius 3. This will allow us to make a statistical prediction about the resulting systems. Our goal is to determine the expected range of performance for our collection of assembled systems. The statistics of the construction parameters may be a result of knowledge about the underlying fabrication process or known from building many such systems and measuring the parameters. The terminology used in this section follows the development in Goodman(1).

Let X be a random variable. In our case, X may be a radius of curvature, a refractive index, etc. The *probability distribution* function $F(x)$ is defined by the probability that the random variable X is less than or equal to the specific value x, i.e.,

$$F(x) = \text{Prob}\{X \le x\} \tag{9.32}$$

where Prob{z} means that z occurs. The probability is the fraction of the time that a particular outcome is expected, relative to all possible outcomes. Since X must have some value, $F(-\infty) = 0$ and $F(\infty) = 1$. Also, $F(x)$ must be non-decreasing for increasing x; for example, the probability that X is less than 10 can not be less than the probability that X is less than 15.

A very useful quantity derived from the probability distribution is the *probability density* function $p(x)$ defined by

$$p(x) = \frac{d}{dx} F(x) \tag{9.33}$$

Using the definition of the derivative, we can show that $p(x)dx$ is the probability that X lies in the range $x \le X < x + dx$. From the properties of the probability distribution $F(x)$, it follows that $p(x)$ has the properties

$$p(x) \ge 0 \tag{9.34}$$

$$\int_{-\infty}^{\infty} p(x)\,dx = 1 \tag{9.35}$$

$$\text{Prob}\{a < X \le b\} = \int_{a}^{b} p(x)\,dx \tag{9.36}$$

A prime example of the usefulness of the probability density function is the computation of *statistical averages* (also called *expected values*). Consider a function $g(x)$. If x represents a random variable, then $g(x)$ is also a random variable. The statistical average of $g(x)$ (which we shall denote by angle brackets $\langle g(x) \rangle$) is defined by

$$\langle g(x) \rangle = \int_{-\infty}^{\infty} g(x)\,p(x)\,dx \tag{9.37}$$

The statistical averages that we shall make the most use of are the *moments*, which are found by using $g(x) = x^n$. Generally, we are primarily interested in the first moment (also known as the mean, expected, or average value)

$$\langle x \rangle = \int_{-\infty}^{\infty} x p(x)\, dx \tag{9.38}$$

and the second moment (or mean-square value)

$$\langle x^2 \rangle = \int_{-\infty}^{\infty} x^2 p(x)\, dx \tag{9.39}$$

We are often interested in the variation of the variable around its mean value. In this case, we can use the *central moments*, which are found by using $g(x) = (x - \langle x \rangle)^n$. The most widely used central moment is the second, which is called the *variance* and is denoted by σ^2:

$$\sigma^2 = \int_{-\infty}^{\infty} \left(x - \langle x \rangle\right)^2 p(x) dx \tag{9.40}$$

It is often easier to calculate the variance using the first and second moments via

$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 \tag{9.41}$$

The square root of the variance, σ, is the *standard deviation*. σ is often used as a measure of the "size" of X or the spread of values taken on by X. We will use the standard deviation to characterize the range of the deviation in performance of our assembled systems from the nominal (design) value. Although, for simplicity, we have only considered a single random variable X, all of the concepts described above have extensions to two or more random variables.

Effect of tolerances on system performance

For our problem of assessing optical system performance, we will have to consider the effects of many random variables. For example, in tolerancing a simple Cooke triplet, we must (at least) consider 6 curvatures, 6 surface irregularities, 3 element thicknesses, 2 air spaces, 3 refractive indices, and 3 element centerings. Let the number of construction parameters to be considered be denoted by n. We are always interested in the deviations of the construction parameters from their nominal values; so let x_i denote the (random) deviation of construction parameter i from its nominal value. The *tolerance limit* is the maximum allowed perturbation and is denoted by Δx_i. (The tolerance limit is the value in the tolerance data spreadsheet.) We will select some measure of system performance, denoted by S, whose sensitivity to changes in the construction parameters we wish to study. S may be a simple first-order quantity, like focal length, or a performance measure such as spot size or MTF. We are really interested in the change in S (i.e., δS) from its nominal value S_0, i.e., $\delta S = S - S_0$. The question of whether the nominal performance S_0 is appropriate or adequate is usually a problem of *optimization*, not tolerancing. In general, δS will be some function f of the construction parameter deviations

$$\delta S = f(x_1, x_2, \ldots, x_n) \tag{9.42}$$

Since x_i is a deviation from a nominal value, if $x_i = 0$ for all i, $\delta S = 0$. We are generally interested in small values of the construction parameter perturbations x_i. Let us assume that the contribution of parameter i to δS can be expressed as a linear function of the perturbation, i.e., the first term in the Taylor expansion of f, so $\delta S_i = \alpha_i x_i$. Then the total change in performance is just the sum of these linear contributions

$$\delta S = \sum_{i=1}^{n} \delta S_i = \sum_{i=1}^{n} \alpha_i x_i \tag{9.43}$$

The first and second moments of δS are

$$\langle \delta S \rangle = \sum_{i=1}^{n} \alpha_i \langle x_i \rangle \tag{9.44}$$

and

$$\left\langle \delta S^2 \right\rangle = \sum_{i=1}^{n} \alpha_i^2 \left\langle x_i^2 \right\rangle + \sum_{i \neq j} \alpha_i \alpha_j \left\langle x_i x_j \right\rangle \tag{9.45}$$

From Eqs. (9.41), (9.44) and (9.45), we can compute the variance of δS as

$$\sigma_{\delta S}^2 = \sum_{i=1}^{n} \alpha_i^2 \left\langle x_i^2 \right\rangle + \sum_{i \neq j} \alpha_i \alpha_j \left\langle x_i x_j \right\rangle - \left(\sum_{i=1}^{n} \alpha_i^2 \left\langle x_i \right\rangle^2 + \sum_{i \neq j} \alpha_i \alpha_j \left\langle x_i \right\rangle \left\langle x_j \right\rangle \right) \tag{9.46}$$

We now make the assumption of *statistical independence* for x_i and x_j, i.e., knowledge about x_i does not influence the probabilities associated with x_j. This a reasonable assumption in our case, since it is unlikely that, for example, the radius of surface 3 affects the thickness of surface 6. Assuming independence, then, x_i and x_j are uncorrelated and $\langle x_i x_j \rangle = \langle x_i \rangle \langle x_j \rangle$. Thus, Eq. (9.46) reduces to

$$\sigma_{\delta S}^2 = \sum_{i=1}^{n} \alpha_i^2 \left(\left\langle x_i^2 \right\rangle - \left\langle x_i \right\rangle^2 \right) = \sum_{i=1}^{n} \alpha_i^2 \sigma_{x_i}^2 \tag{9.47}$$

where

$$\sigma_{x_i}^2 = \left\langle x_i^2 \right\rangle - \left\langle x_i \right\rangle^2 \tag{9.48}$$

is the variance of construction parameter *i*. We see from Eqs. (9.44) and (9.47) that the mean and variance of δS are simply weighted sums of the means and variances of the construction parameters.

We can simplify Eq. (9.47) for the common case where the variance can be expressed as a simple function of the tolerance limit Δx_i. Consider the case where

$$\sigma_{x_i} = \kappa_i \Delta x_i \tag{9.49}$$

where κ_i is a constant. Now,

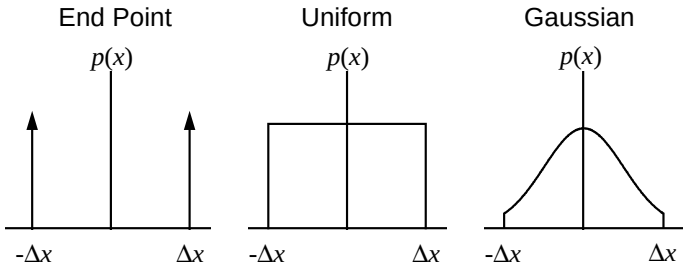
$$\sigma_{\delta S}^2 = \sum_{i=1}^{n} \alpha_i^2 \left(\kappa_i \Delta x_i \right)^2 = \sum_{i=1}^{n} \kappa_i^2 \left(\Delta S_i \right)^2 \tag{9.50}$$

where ΔS_i is the change in performance when the construction parameter is equal to its tolerance limit Δx_i, i.e., $\Delta S_i = \alpha_i \Delta x_i$. If all of the construction parameters have the same type of probability distribution, then κ_i is the same for each parameter (call it κ) and the standard deviation of the performance change is

$$\sigma_{\delta S} = \kappa \sqrt{\sum_{i=1}^{n} \left(\Delta S_i \right)^2} \tag{9.51}$$

Equation (9.51) provides the basis for converting tolerance values into an expected range of system performance: given a set of tolerance limits Δx_i, we compute the change in performance for each perturbation, ΔS_i. Then, we use an appropriate value of κ to compute the standard deviation of system performance $\sigma_{\delta S}$ using Eq. (9.51).

Three commonly used probability density functions for x_i are shown in the figure below. When the fabrication errors occur only at the ends of the tolerance range centered on the design value, we have an *end point* distribution. When the errors are equally likely to occur anywhere within the tolerance range centered about the design value, the probability density function is a *uniform* distribution. If the fabrication process tends to concentrate errors near the center of the tolerance range, we can model the distribution as a *Gaussian* (or *normal*) distribution, truncated at the 2σ level. All of these distributions are symmetric about the design value, so $\langle x_i \rangle = 0$, and the average change in performance is also 0 (see Eq. (9.44)).



We can easily compute the variances of the three distributions pictured above and hence the corresponding values of κ:

Distribution	κ
End point	1.0
Uniform	0.58
Gaussian	0.44

From the above table, we see that the often used *RSS rule* (square *R*oot of the *S*um of the *S*quares; use $\kappa = 1$ in Eq. (9.51)) for tolerance budgeting results from implicitly assuming an end point distribution function, along with the other assumptions we have made (linear relationship between perturbation and performance change; statistical independence). It would appear rather unlikely that real manufacturing processes result in end point distributions of manufacturing errors. Since the value of κ is smaller for the more realistic choices of probability density functions (uniform, Gaussian), the RSS rule results in a pessimistic estimate of the performance change standard deviation. It should be noted, however, that this is not necessarily a bad thing. Also note that, regardless of κ, the squaring operation in Eq. (9.51) means that the larger tolerance effects will dominate in the computation of the standard deviation.

We can use Eq. (9.51) to compute the standard deviation in system performance, but this does not tell us about the form of the distribution in performance of the resulting systems. We need to know this form in order to make predictions about the probability of "success", where success occurs when a fabricated system has a performance change within specified limits. Fortunately, we can make use of another concept from statistics: the *central limit theorem*. This theorem states that for a set of independent random variables $x_1, x_2, \ldots, x_n$, with *arbitrary* probability density functions, the probability density for the random variable $z = \Sigma x_i$ approaches a Gaussian density as $n \to \infty$. For our purposes, this means that we should expect that our resultant optical systems should have a nearly Gaussian distribution in system performance. Among the many mathematically "nice" features of Gaussian random variables is the fact that the distribution is completely specified by knowledge of the mean μ and standard deviation σ:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(z-\mu)^2}{2\sigma^2}\right] \tag{9.52}$$

All of the probability density functions that we are considering have zero mean value, so $\langle \delta S \rangle = 0$. Thus, from Eq. (9.52), we would expect that the probability density function for δS is

$$p(\delta S) = \frac{1}{\sqrt{2\pi}\sigma_{\delta S}} \exp\left[\frac{-(\delta S)^2}{2\sigma_{\delta S}^2}\right] \tag{9.53}$$

From Eqs. (9.36) and (9.53), we can compute the probability that the system performance is in the range $\pm\, \delta S_{max}$ as

$$\text{Prob}\left\{\left|\delta S\right| \le \delta S_{\max}\right\} = \int_{-\delta S_{\max}}^{\delta S_{\max}} p(\delta S)\, d(\delta S) = \text{erf}\left(\frac{\delta S_{\max}}{\sqrt{2}\sigma_{\delta S}}\right) \tag{9.54}$$

where erf(x) is the error function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{9.55}$$

Given a standard deviation in performance change $\sigma_{\delta S}$ and a maximum acceptable performance change δS_{max}, we can use Eq. (9.54) to calculate the probability of success. For example, approximately 95% of the resulting systems will have a performance change in the range $\pm 2\sigma_{\delta S}$. A desired level of probable success is obtained by distributing the construction parameter tolerances such that the resulting performance changes yield the required standard deviation. The table below gives the probability of success, computed from Eq. (9.54), for several values of the ratio of acceptable performance change δS_{max} to standard deviation $\sigma_{\delta S}$.

$\delta S_{max} / \sigma_{\delta S}$	Probability of success
0.67	0.50
0.8	0.58
1.0	0.68
1.5	0.87
2.0	0.95
2.5	0.99

This statistical analysis in computing the probability that optical system performance will meet or exceed a designated performance level can be calculated from performance changes computed in a *sensitivity* analysis or targeted in an *inverse sensitivity* computation. Sensitivity analysis starts from either predefined (i.e., input by the designer) or default (e.g., from ISO 10110) tolerance limits assigned to each construction parameter. Data for a sensitivity table (or change table) is generated by perturbing each construction parameter of the design and computing the resulting change in system performance (i.e., ΔS_i). Inverse sensitivity analysis starts from a predefined or default change in system performance that is permitted for each construction parameter. The analysis then determines the tolerance limit for each construction parameter that will cause the permitted performance change.

One method of establishing tolerances is to distribute the tolerances to produce performance changes that balance the contributions to the tolerance budget. This prevents the standard deviation, hence the probability of success, from being dominated by a single item. We can use Eq. (9.51) to develop a starting point. Let us assume that we have established a target value of the standard deviation of system performance change $\sigma_{\delta S}$. Further assume that we have n construction parameters and that each parameter has the same probability density function. If each construction parameter is to contribute equally to the overall performance standard deviation, then ΔS_i is the same for each i. Designate this target value of ΔS_i by ΔS_{tar}. In this case, Eq. (9.51) takes the form

$$\sigma_{\delta S} = \kappa \sqrt{\sum_{i=1}^{n} \left(\Delta S_{tar}\right)^2} = \kappa \sqrt{n \left(\Delta S_{tar}\right)^2} = \kappa \sqrt{n} \Delta S_{tar} \tag{9.56}$$

Thus, the allowed performance change for each construction parameter is found from Eq. (9.56) to be

$$\Delta S_{tar} = \frac{\sigma_{\delta S}}{\kappa \sqrt{n}} \tag{9.57}$$

This suggests that one way to begin the creation of a tolerance budget is to start with a prescribed probable success rate and maximum allowed performance change and use Eq. (9.54) to compute the allowed standard deviation in system performance change. Then, Eq. (9.57) can be used to compute the targeted performance change for each construction parameter. An inverse sensitivity analysis with this requested performance change for each parameter yields the allowed tolerances.

The discussion in this section has concentrated on presenting the analysis of tolerance effects from a statistical point of view. It should be stressed, however, that a complete automation of the tolerancing process is probably an unrealistic goal. Coupling the fact that there is rarely an optical system that can unambiguously be called the "right answer" to an optical design problem with the inherent variations in manufacturing processes and techniques, there are many aspects of a complete tolerancing analysis that require the skill and experience of the designer.

User-defined tolerancing

The user-defined tolerancing routines perform the tolerance calculations using the current optimization error function. The motivation for this approach is to provide general flexibility in specifying the performance criterion: anything that can be computed as an operand can be used as a tolerance criterion. With the flexibility provided by the built-in operand types along with CCL and SCP operands, the user-defined tolerancing provides the capability to "tolerance on anything." Another advantage of using the optimization error function is the ready implementation of compensation schemes for reducing the effects of the perturbations. The optimization, in this case, is done in an effort to restore the nominal system performance.

The error function that is used for the tolerance analysis may be the one that was used in the design of the lens or it may be one that has been developed specifically for tolerance evaluation. This flexibility in defining the tolerance criterion places a responsibility on the designer to specify optimization conditions that either directly describe a system specification or provide a mapping from one to the other. Compensators are designated by the current optimization variables. During the tolerancing process, the compensators will be varied in an attempt to make the change in the error function caused by the tolerance perturbation as small as possible.

Change table tolerancing

A change table is a tabulation of the changes in optical aberrations and system quantities that are produced by perturbing the construction parameters of the lens. The changes are calculated by simply perturbing each parameter by the tolerance value and recomputing all of the items contained in the change table. An advantage of the change table tolerancing is that there is no need to set up an error function, as is required for the user-defined tolerancing. The same change table items are computed for each lens, so it is easy to compare tolerance sensitivities across different lenses. Breaking down the system performance changes by aberration allows you to locate potential "trouble spots" during the tolerancing process.

The aberrations in the change table are reported in *tolerance units*. The unit for the particular aberration depends upon whether the aberration is measured as a transverse (*T*) quantity, a longitudinal (*L*) quantity, or a wavefront (*W*) quantity. In the default case, these units correspond to the quarter-wave or Strehl tolerance limit of 0.8. Thus, by default, the performance changes are reported in "times-diffraction-limit" values. You can, of course, scale the tolerance units by any desired value if this is more convenient. For example, to work with the wavelength (rather than quarter-wavelength) as the basic aberration unit, scale the tolerance units by 0.25.

MTF/RMS wavefront tolerancing

All tolerancing calculations involve computing the change in some performance measure for a small perturbation of each construction parameter of interest. It is not hard to see that this requires the evaluation of many different optical systems. For example, it was mentioned earlier in this chapter that the tolerance analysis of a simple Cooke triplet requires the consideration of (at least) 23 construction parameters. If we combine this fact with a performance measure that requires a relatively computationally intensive evaluation (tracing many rays and/or a complex evaluation),

we find that it may require an objectionably long period of time to perform a tolerance analysis. Hopkins and Tiziani, in a study of lens centering errors, introduced a method of quickly computing differential optical path changes for a perturbed optical system, without reevaluating the entire system. The advantage of this method is that the effects of the tolerance perturbations can be calculated from data that is obtained by ray tracing only the nominal system. This method is many times faster than methods that require that all necessary rays be retraced for each perturbed system in order to compute tolerance operands. This speed and efficiency make it practical to compute tolerance effects using a numerically intensive computation such as MTF.

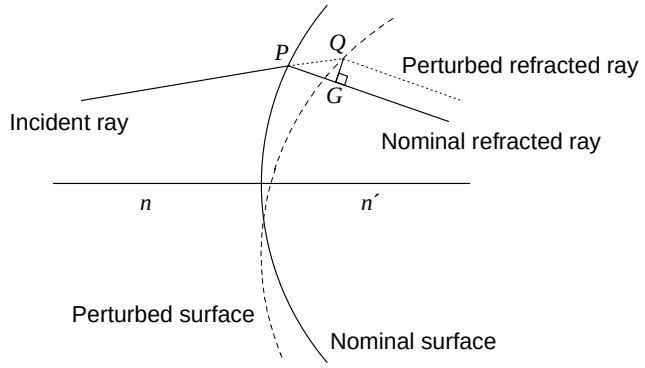
The change in optical path introduced by a perturbed surface is illustrated in the figure below. The construction line *QG* is perpendicular to the nominal ray at *G* and passes through the intersection of the perturbed ray with the perturbed surface at *Q*. Hopkins and Tiziani used Fermat's principle to conclude that, to a first approximation, the optical path length from *Q* to the final wavefront and the optical path length from *G* to the final wavefront are the same. Thus, the change in optical path δW due to the perturbation is

$$\delta W = nPQ - n'PG \tag{9.58}$$

where n and n' are the refractive indices on the incident and refracted sides of the surface. Ray trace data from evaluation of the nominal system provides most of the data necessary to evaluate the perturbed system. Only the physical distance PQ needs to be computed for the ray at the perturbed surface, since Eq. (9.58) may be written as

$$\delta W = (\delta \mathbf{r} \bullet \mathbf{g})(n\mathbf{i} \bullet \mathbf{g} - n'\mathbf{i}' \bullet \mathbf{g}) \tag{9.59}$$

where $\delta \mathbf{r}$ is the displacement of the surface, **g** is the surface normal vector, **i** is the incident ray vector and **i′** is the refracted ray vector. The vectors **g**, **i**, and **i′** are available from the ray trace of the nominal system and $\delta \mathbf{r}$ can be computed for each type of surface perturbation of interest.



It should be pointed out that the savings in time comes both at the expense of storing the original ray trace data at each surface and at the expense of the approximation of the perturbed wavefront. Also, not all tolerance criteria involve the computation of optical path or wavefront errors (e.g., focal length, magnification, etc.), in which case Hopkins and Tiziani's method is not applicable. The interested reader is referred to the original paper by Hopkins and Tiziani2 and to the extensive development of tolerancing methods using this technique by Rimmer3 for more details on the computation of the wavefront differential terms.

Using the analytic wavefront differential calculation, it is possible to compute tolerance operands for MTF and RMS wavefront error. In both of these cases, the change in system performance is expanded in a general quadratic equation in the construction parameters. This is in contrast to the linear functionality used earlier in this chapter. The change in system performance can be expressed as

$$\delta S = \sum_{i=1}^{n} A_i x_i^2 + \sum_{i=1}^{n} B_i x_i + \sum_{i \neq j} C_{ij} x_i x_j \tag{9.60}$$

We can compute the mean and variance of δS using a procedure similar to that used earlier in this chapter when δS was assumed to be a linear function of x_i. The resulting expressions are, as might be expected, rather complicated. However, if we assume that the probability density functions associated with all x_i are symmetric and have zero mean value, the mean and variance of δS have the (comparatively) simple forms

$$\langle \delta S \rangle = \sum_{i=1}^{n} A_i \sigma_{x_i}^2 \tag{9.61}$$

and

$$\sigma_{\delta S}^2 = \sum_{i=1}^{n} A_i^2 \left(\langle x_i^4 \rangle - \sigma_{x_i}^4 \right) + \sum_{i=1}^{n} B_i^2 \sigma_{x_i}^2 + \sum_{i \neq j} C_{ij}^2 \sigma_{x_i}^2 \sigma_{x_j}^2 \tag{9.62}$$

Examination of the A and B coefficients reveals how the relative magnitudes of A_i and B_i affect the statistics of δS. When the nominal lens is well-corrected with respect to S for x_i, S_0 is (or is close to) an extremum, so the A_i coefficient is much larger than B_i. We would expect then, from Eqs. (9.61) and (9.62), that the main effect is a change in the average performance. On the other hand, if the nominal lens is not well-corrected, then the B_i term dominates the A_i term and the main effect is an increase in the standard deviation, i.e., a larger range in performance for the resulting systems. A complete discussion of this general quadratic transformation can be found in the paper by Koch.4

Obviously the quadratic approximation of Eq. (9.60) is only valid for small performance changes. Fortunately, this is exactly the situation in which we are usually interested when performing a tolerance analysis. It is entirely possible, however, to specify tolerance limits that result in A_i and B_i coefficients that predict non-realizable performance changes, e.g., MTF values less than zero or greater than unity. This means that the quadratic approximation is not valid in this regime and the MTF or RMS wavefront can not be computed with any accuracy using this technique. The usual implication of results of this type is that the current tolerance values are too large and the performance degradation is probably quite serious and unacceptable. You should consider reduction of the tolerance assignments and/or designating more compensators if you notice unphysical MTF or RMS wavefront predictions.

Monte-Carlo tolerancing

Monte Carlo analysis uses random numbers to generate a sequence of perturbed lenses, where the maximum magnitude of the perturbations is determined by the current values of the tolerances. Each random realization of the lens is constructed by generating random numbers having a prescribed probability density function and then using these random numbers along with the tolerances to perturb the construction parameters of the system. An advantage of Monte Carlo analysis is that all of the construction parameters may be perturbed simultaneously. After all of the perturbations are applied, the compensators (if any) are varied in an attempt to restore the performance of the lens as close as possible to its nominal state of correction. Analysis of the performance of the resulting systems provides a statistical prediction of the distribution of the final fabricated lenses. Because of the stochastic nature of the process, depending upon the lens and its sensitivity to its construction parameters, the Monte Carlo analysis may converge slowly to the true value of the performance statistics. Also, since all of the parameters are varied simultaneously, it can be difficult to locate which parameters are the most sensitive. However, Monte Carlo analysis can be quite useful when used in conjunction with other tolerancing techniques.

The setup procedure for a Monte Carlo analysis is identical to that for user-defined tolerancing: construction of an error function and designating compensators (variables). The current error function will be used as the performance measure, so it is important that the error function accurately reflects the system performance, even in its perturbed state. For example, if you are tolerancing decentrations and tilts, the error function should contain rays on both sides of the entrance pupil, and both positive and negative field points, even for a nominally rotationally symmetric lens. If the current error function appears to assume rotational symmetry, a warning message to that effect will be issued before the analysis is begun.

Just as for user-defined tolerancing, the error function value will be the prime measure of the change in the lens performance. However, you can also perform a statistical analysis of any selected operands by giving them the name “tolop”. These operands may have zero weight if you don’t want them included in the error function.

OSLO provides several controls for the computation of statistical tolerance data:

Number of random systems to evaluate – This is the number of different random lenses that will be generated using the current tolerance values. As with all simulations, the more systems that are evaluated, the closer the resulting statistics should be to the “real world.” Of course, this accuracy means that the analysis takes a longer time to perform. After beginning the computations, you may interrupt the routine before all of the specified systems have been evaluated by pressing the ESCAPE key.

Perturbation distributions – This option controls the distribution of the random perturbations applied to the lens. The default distributions are obtained from the tolerancing operating conditions. The other three options (uniform, Gaussian, end-point) will apply the specified distribution to all perturbations.

Plot error function distribution – If you wish, you can plot both the cumulative probability (continuous curve) and a relative distribution histogram (vertical lines) for the error functions of the resulting ensemble of Monte Carlo systems.

After all of the requested systems have been evaluated, a statistical summary of the error function, compensators, and all “tolop”-named operands is presented. The definitions of the statistical quantities may be found in any statistics reference; we use the notation and terminology of Press, et. al., to which the reader is referred for more detail. In the following, N denotes the number of systems evaluated, and xj denotes either the change in the error function or the change in an operand value.

The mean change is given by

$$\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j \tag{9.63}$$

The standard deviation s, is the square root of the variance s^2, where

$$\sigma^2 = \frac{1}{N-1}\sum_{j=1}^{N}\left(x_j - \bar{x}\right)^2 \tag{9.64}$$

The average deviation (or mean absolute deviation is)

$$\text{AVG DEV} = \frac{1}{N}\sum_{j=1}^{N}\left|x_j - \bar{x}\right| \tag{9.65}$$

The skewness is a nondimensional version of the third central moment, and measures the shape and asymmetry of the distribution. A positive skewness means that the distribution has a "tail" extending toward positive values of xj from the mean. Conversely, a distribution with a negative skewness has a tail extending toward negative values of xj.

$$\text{SKEWNESS} = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{x_j - \bar{x}}{\sigma}\right]^3 \tag{9.66}$$

The kurtosis is a nondimensional version of the fourth central moment, and has a value of zero for a Gaussian (normal) distribution. A distribution with a positive kurtosis is more "peaked" than a Gaussian, while a negative kurtosis distribution is "flatter" than a Gaussian.

$$\text{KURTOSIS} = \left[\frac{1}{N}\sum_{j=1}^{N}\left\{\frac{x_j - \bar{x}}{\sigma}\right\}^4\right] - 3 \tag{9.67}$$

The "+/–" ranges displayed for the mean and standard deviation are, respectively, the standard deviation of the mean

$$\left(\frac{\sigma}{\sqrt{N}}\right) \tag{9.68}$$

and the standard deviation of standard deviation

$$\left(\frac{\sigma}{\sqrt{2(N-1)}}\right) \tag{9.69}$$

Also displayed for the error function and the "tolop" operands is a cumulative probability table. The cumulative probability (or probability distribution) is the probability that the value of the random variable is less than or equal to the specific value. So, the cumulative probability of the minimum change in the quantity is 0 % and the cumulative probability of the maximum change is 100 %. Also, the median change in a quantity is given by the 50 % cumulative probability value.

---

1 J. W. Goodman, *Statistical Optics*, Wiley, 1985, Chap. 2.
2 H. H. Hopkins and H. J. Tiziani, "A theoretical and experimental study of lens centring errors and their influence on optical image quality," Brit. J. Appl. Phys. **17**, 33-54 (1966).
3 M. P. Rimmer, "Analysis of perturbed lens systems," Appl. Opt. **9**, 533-537 (1970); "A tolerancing procedure based on modulation transfer function (MTF)" in *Computer-Aided Optical Design*, Proc. SPIE Vol. 147, pp. 66-70 (1978).
4 D. G. Koch, "A statistical approach to lens tolerancing," in *Computer-Aided Optical Design*, Proc. SPIE Vol. 147, pp. 71-82 (1978).

Chapter 10 Examples

Often the best way to learn how to do something in a new program is to see how a similar task was performed using the program. Accordingly, this chapter consists of a number of systems that make use of one or more of the features of OSLO. Some are included to serve as base designs that you can use to develop enhanced systems; others are included for their tutorial value. Most examples correspond to files in the public\len\demo directory in OSLO. Each contains a listing, drawing, and some comments on the system.

For most examples, you should be able to get a good idea of the purpose by reading the description here and experimenting with the file using the toolbar icons and menus. If you have trouble understanding something, try tracing a few single rays through the system, or making a change to the system and seeing its effect. If you really want to see the details, you can open the file in the text editor.

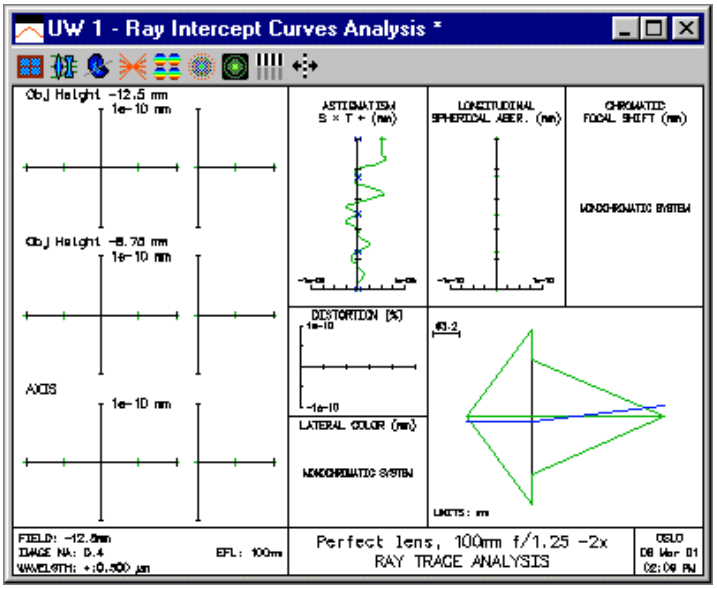
Using OSLO to maximum advantage requires an understanding of both optical design and how the program works. These examples concentrate on the middle ground. No specific attempt is made here to teach optical design, which is better presented in textbooks, nor how OSLO works, which is better taught in the help system. We assume that you know something about both, but everything about neither. The best use of OSLO is to fill in the gaps in your understanding.

Standard lenses

Perfect lens

A perfect lens is one that forms a sharp undistorted image of an extended object on a plane surface. If the aperture of the lens is infinitesimal, the rays passing from the object to the image will follow the laws of paraxial optics. If the aperture of the lens is finite, however, this will not be the case. Abbe's sine law states that for finite rays, the ratio of the sines of the object and image axial ray angles must be constant. This conflicts with the laws of paraxial optics, which state that the ratio of the tangents of the angles must be constant.

OSLO uses true perfect lenses, i.e. ones that obey the laws of optics, to model idealized systems. As a result, OSLO results will be different from programs that use paraxial lenses to model idealized systems. The perfmag2.len file is included in the demo directory to illustrate some properties of perfect lenses that may seem curious to those who are not familiar with them. The lens is purposely chosen to be very fast (N.A. 0.8 on the short conjugate side), and to work at a magnification of -2x, so that the differences between a perfect lens and a paraxial lens are obvious. The drawing below shows a basic ray analysis of the lens at its nominal magnification.



All the ray aberrations are essentially zero, according to the definition of a perfect lens (the residual wiggles are due to round-off errors). The drawing of the lens is unusual. In OSLO, a perfect lens is modeled as a single surface. If the lens is to obey the sine law, however, this means that the rays must emerge from the surface at a different height than they enter, which accounts for the strange drawing. The paraxial data for the perfect lens are shown below.

```
*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:     200.000000    Image axial ray slope:     -0.666667
   Object num. aperture:       0.800000    F-number:                   0.250000
   Image num. aperture:        0.400000    Working F-number:           1.250000
 FIELD
   Field angle:                4.763642    Object height:            -12.500000
   Gaussian image height:     25.000000    Chief ray ims height:      25.000000
 CONJUGATES
   Object distance:          150.000000    Srf 1 to prin. pt. 1:          --
   Gaussian image dist.:     300.000000    Srf 2 to prin. pt. 2:          --
   Overall lens length:          --        Total track length:       450.000000
   Paraxial magnification:    -2.000000    Srf 2 to image srf:       300.000000
 OTHER DATA
   Entrance pupil radius:    200.000000    Srf 1 to entrance pup.:        --
   Exit pupil radius:        200.000000    Srf 2 to exit pupil:           --
   Lagrange invariant:       -16.666667    Petzval radius:          1.0000e+40
   Effective focal length:   100.000000
```

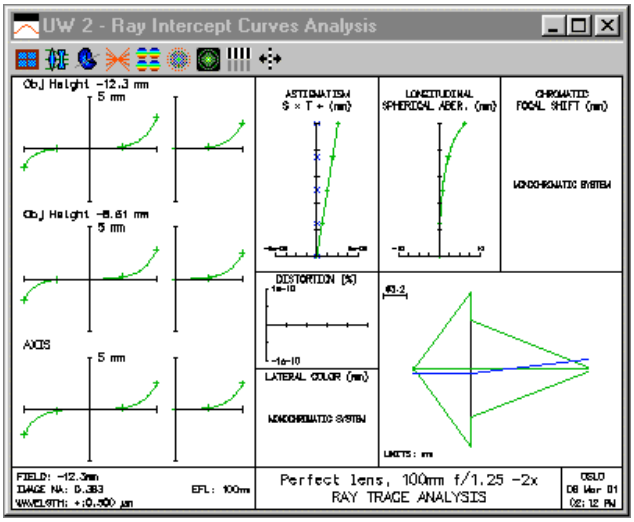
It is worth spending a little time to see how the values are calculated. First, the focal length of the lens is specified to be 100mm, the magnification is specified to be -2, the numerical aperture is specified as .8, and the object height is specified as -12.5mm; these are all given data.

The focal length and magnification specifications imply that the object distance must be 150mm, and the image distance must be 300mm. The magnification and numerical aperture specifications imply that the numerical aperture on the image side must be .4. If the numerical aperture is .8, the axial (marginal) ray in object space must make an angle arcsin(.8) = 53.130102 degrees with the optical axis, and must strike the lens at a height of 150*tan(53.130102) = 200mm, which is the entrance beam radius. The *f*-number, defined as the focal ratio f/D, must then be 100/400 = .25. The working *f*-number, defined as 1/(2*NA), is then 1/(2*0.4) = 1.25. The image axial ray slope (a paraxial quantity) is -200/300 = -.666667.

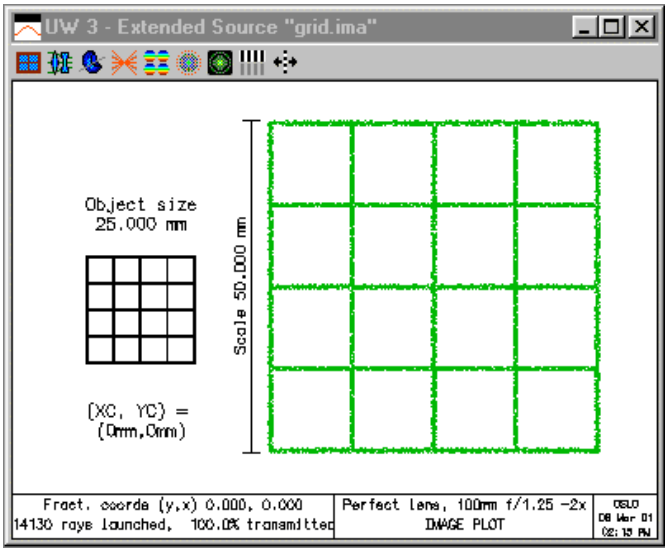
Since the numerical aperture in image space is 0.4, the angle of a real ray emerging from the lens must be -arcsin(.4) = -23.578178 degrees. If this ray is to pass through the paraxial image point, it must emerge from the lens at a height 300*tan(23.578178) = 130.930734mm. This is easily confirmed by tracing a real ray from an on-axis object point, as shown below. The difference in ray height between the input and output rays is thus caused by the fact that we are dealing with a real perfect lens, not a paraxial approximation. Since an actual ray could obviously not follow the displayed trajectory, the implication is that a real perfect lens cannot be infinitely thin.

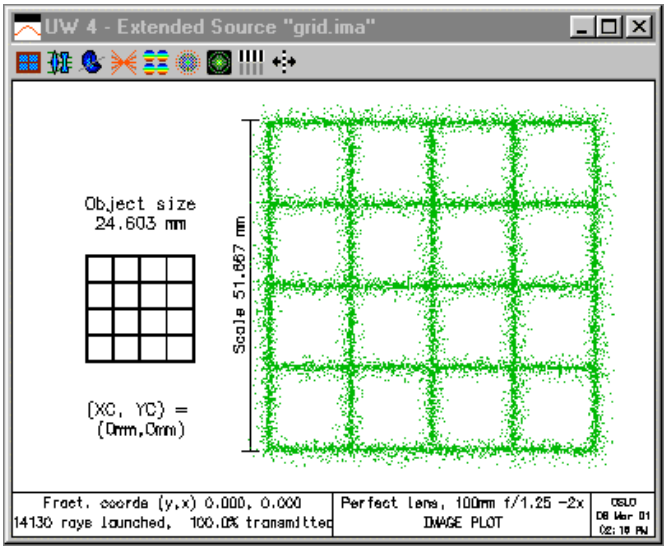
```
*SET OBJECT POINT
         FBY          FBX          FBZ
          --           --           --
        FYRF         FXRF          FY           FX
          --           --           --           --
          YC           XC          YFS          XFS          OPL     REF SPH RAD
          --           --           --           --  300.000000  299.999993
*TRACE RAY - LOCAL COORDINATES
 SRF         Y            X            Z           YANG         XANG         D
  1     200.000000       --           --        53.130102       --           --
  2     130.930734       --           --       -23.578178       --      -127.326835

  3     5.6843e-14       --           --       -23.578178       --       327.326835
 PUPIL       FY           FX                                                  OPD
          1.000000        --                                                 --
```

A perfect lens cannot be characterized solely by a focal length. It is necessary also to specify a magnification at which it is to be used. This is because a lens cannot form a perfect image at two magnifications (Herschel's rule). The perfect lens used here can be used to demonstrate this by changing the magnification by 5% to -2.1. Then the object distance becomes 147.619048mm, and the image distance 310mm. The ray analysis shows that even with this small change in operating condition, there is a substantial overcorrected spherical aberration.



The effects of using the perfect lens at the wrong magnification are sufficiently dramatic that they can easily be seen in the image of a rectangular grid (obtained using the ***xsource** command).


UW 3 - Extended Source "grid.ima"
Object size
25.000 mm
Scale 50.000 mm
(XC, YC) =
(0mm,0mm)
Fract. coords (y,x) 0.000, 0.000
14130 rays launched, 100.0% transmitted
Perfect lens, 100mm f/1.25 -2x
IMAGE PLOT
OSLO
08 Mar 01
02:10 PM

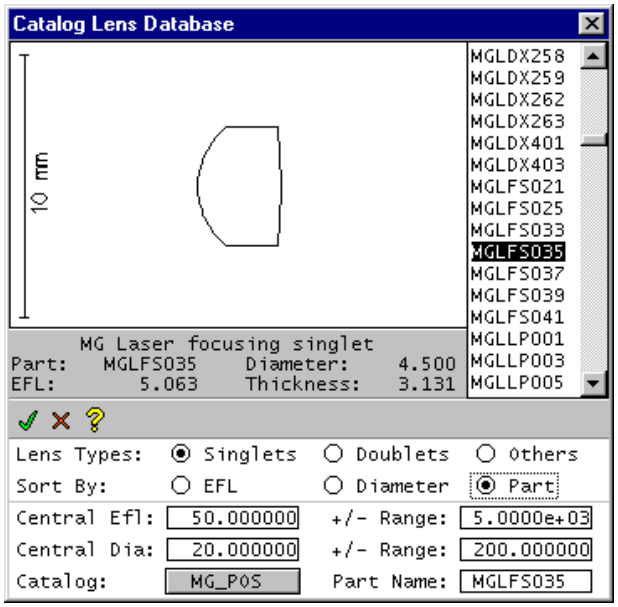

UW 4 - Extended Source "grid.ima"
Object size
24.603 mm
Scale 51.867 mm
(XC, YC) =
(0mm,0mm)
Fract. coords (y,x) 0.000, 0.000
14130 rays launched, 100.0% transmitted
Perfect lens, 100mm f/1.25 -2x
IMAGE PLOT
OSLO
08 Mar 01
02:10 PM

Catalog lens

The output data from the ray trace routines in OSLO can best be explained by reference to an example. The example uses a singlet catalog lens to illustrate the meaning of ray output data. Explicit instructions are given so you can reconstruct the output using OSLO. You should, of course, get the same results as shown here.

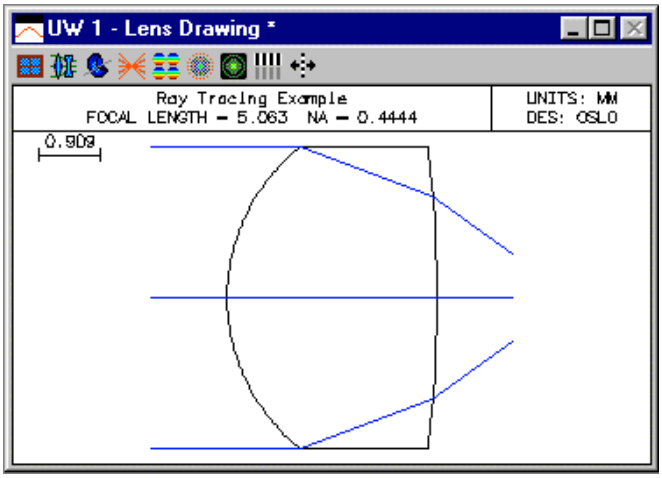
1. Enter the lens data. Click File >> New, enter the file name mglfs035.len, select Catalog lens, enter the magnification –0.5, then click Green check to dismiss the dialog box.

2. The catalog database spreadsheet opens. Switch to the MG_POS catalog if it is not already the current catalog. Click the Part Name field, then enter the part number MGLFS035. Click the Green check button to close the spreadsheet.



3. In the Lens id field of the surface data spreadsheet, change “No name” to “Ray trace example”. Click the Group radio button, which should change to Surfs. Click the Object numerical aperture field, then enter 0.1. Click the Object height field, then enter –4.0. Click the *Save the current lens* toolbar icon to save the lens data. Click Green check to close the spreadsheet.

4. In the current graphics window, Click the Draw 2D plan view toolbar icon. The graphics window should appear as follows.



5. In the current text output window, click the Len button, then click the Pxc button. The text window should contain the following.

```
*LENS DATA
Ray Trace Example
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ       --         14.903256      4.000000            AIR

 AST     2.865000 F    3.131000 F    2.250000 AF          BK7 F *
  2    -18.932000 F    5.709113 S    2.250000 F           AIR

 IMS       --            --          2.000000 S

*PARAXIAL CONSTANTS
   Effective focal length:    5.062846    Lateral magnification:    -0.500000
   Numerical aperture:        0.200000    Gaussian image height:     2.000000
   Working F-number:          2.500000    Petzval radius:           -7.303485
   Lagrange invariant:       -0.4020156.
```

6. To simplify the following steps, clear the contents of the text window by right-clicking in the window and selecting the bottom command, *Clear window and SS buffer*. Next, define an object point at the edge of the field of view and trace the chief ray from this point by clicking the **Chf** button in the text window. The text window should contain the following.

```
*SET OBJECT POINT
        FBY          FBX          FBZ
       1.000000      --           --
        FYRF         FXRF         FY          FX
         --           --         0.037489     --
         YC           XC          YFS         XFS          OPL    REF SPH RAD
       1.981388      --         -1.061876   -0.542210   10.703427    8.085073

*TRACE RAY - LOCAL COORDS - FBY  1.00, FBX  0.00, FBZ  0.00
 SRF         Y            X           Z          YANG        XANG         D
  1       -0.055755      --          0.000543   10.103536      --        -0.013834
  2        0.500883      --         -0.006627   14.606424      --         3.173037

  3        1.990406      --            --       14.606424      --         5.906637
 PUPIL       FY           FX          RAY AIMING                          OPD
             --           --        CENTRAL REF RAY                       0.053803
```

The first line just prints the fractional object point coordinates as they were entered in the dialog box. The first two numbers on the second line also echo input data (these are the fractional heights of the reference ray on the reference surface, normalized to the reference surface aperture radius). The last two numbers on the second line are the fractional object space coordinates of the reference ray. The third line contains output data for the reference ray on the image surface. YC and XC are the ray heights (Y_{chief} and X_{chief}) measured from the origin of the image surface coordinate system. YFS and XFS are the longitudinal distances from the image surface to the differential focus in the *yz* and *xz* planes, respectively. YFS and XFS are measured parallel to the *z*-axis, not along the reference ray. OPL is the optical distance along the reference ray from the object sphere to the reference sphere, and REF SPH RAD is the radius of the reference sphere (used to compute optical path difference). The next task is to trace the reference ray as an ordinary ray (i.e., by giving its object-space coordinates). The proper way to do this is to obtain the required data from the spreadsheet buffer.

7. In the text window, click on the value of FY. You should see the full-precision value echoed in the message area, next to the command line (c2 = 0.0374888201129). Click the **Tra** button in the text window. In the dialog box, change the Output format control to Full. Change the Surface selection option to All. In the field for FY, enter **c2**; this will be echoed as 0.037489, but the cell will contain the full precision value. Now execute the command by clicking OK. The text window will contain the following data.

```
*TRACE RAY - LOCAL COORDS - FBY  1.00, FBX  0.00, FBZ  0.00
 SRF         Y/L          X/K          Z/M       YANG/IANG    XANG/RANG      D/OPL
  1     2.5232e-14       --       1.1111e-28     9.840230       --       6.5408e-15
          0.170901       --         0.985288    15.023975     9.840230   6.5408e-15
  2       0.541737       --        -0.007752    14.134625       --         3.169882
          0.244201       --         0.969725     8.200494    12.494889     4.808078
```

```
  3       1.981388      --          --       14.134625      --        5.895349
          0.244201      --        0.969725   14.134625   14.134625   10.703427
 PUPIL      FY          FX          RAY AIMING                          OPD
          0.037489      --       CENTRAL REF RAY                   -3.0233e-12
```

The output of the trace ray command has been described in chapter 6. Note that the height of the reference ray on the reference surface (surface 1) is essentially zero, as is the optical path difference. You may recall that the non-zero value of FY needed to accomplish this comes from the fact that the object numerical aperture of the lens is large enough to require aplanatic ray tracing.

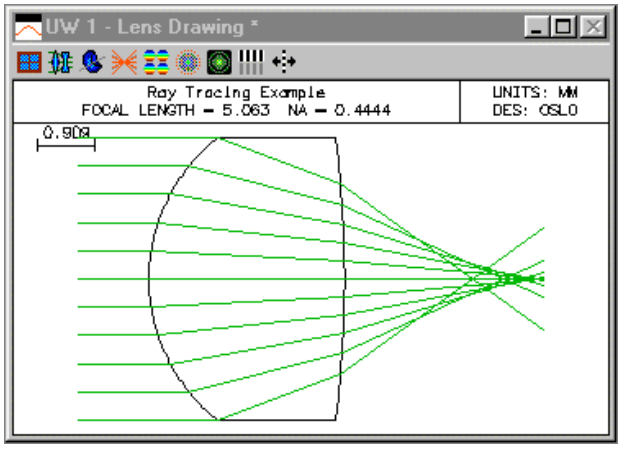
8. There is another command in OSLO, related to the **set_object_point** command, that provides additional output of differential information. The **trace_ray_derivs** (**trd**) command is only available from command mode. It requires the same input data as the **sop** (or **trr**) command. Type the command **trd** in the command line, then press ENTER. In the text output window, you will see the same output as before, with the following additional lines that give the derivative information for the reference ray.

```
*TRACE REFERENCE RAY WITH DERIVATIVES
         FBY          FBX          FBZ
       1.000000       --           --
        FYRF         FXRF          FY           FX
         --           --         0.037489       --
         YC           XC           YFS          XFS          OPL     REF SPH RAD
       1.981388       --        -1.061876    -0.542210    10.703427     8.085073

       D(YC/FY)     D(XC/FY)     D(YC/FX)     D(XC/FX)
      -0.250976       --           --        -0.117678

      D(YC/FBY)    D(XC/FBY)    D(YC/FBX)    D(XC/FBX)
       1.944771       --           --         1.981388
```

The first number gives the change in chief ray height per unit change in fractional aperture height, and so forth. It is important to remember that these are derivatives, and accurately describe only rays that have infinitesimal displacements from the reference ray. These differential rays are sometimes called *parabasal* rays. If the reference ray is the optical axis, the differential rays become equivalent to paraxial rays.

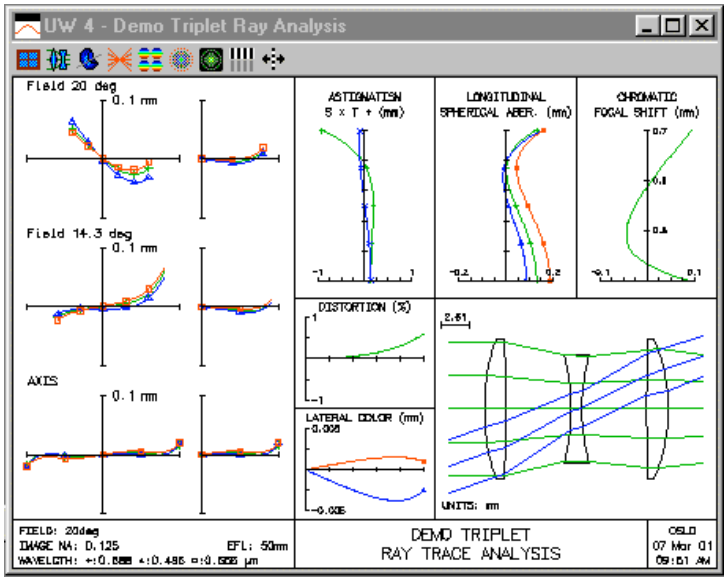
It is a good idea to remember that OSLO is primarily a numerical optimization program, not a drawing program. Drawings are used in OSLO to aid your understanding of the numerical data. If the drawing does not correspond with what you expect, it may be because the numerical data is not what you think it is, or it may be because the drawing routines are not sufficiently powerful to handle the actual data for your system. There are lots of cases of complex optical systems that are computed properly by OSLO, but not drawn properly.

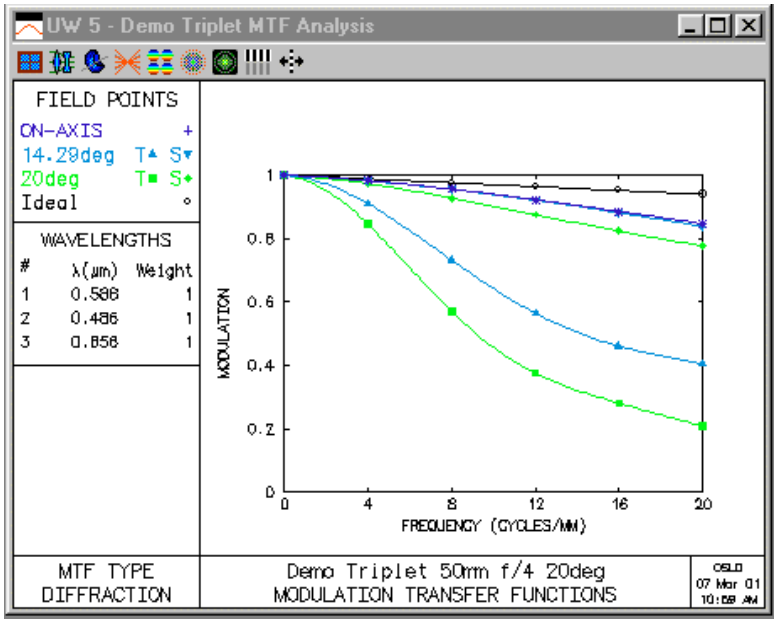
Lens drawings can be tailored to fit many special situations by setting the proper lens drawing operating conditions, which can be opened in several ways, e.g. from the menu that pops up when you use the drawing buttons in the standard graphics window tools. In the present situation, you may wish to reset the default drawing rays to eliminate the off-axis field points, since they don't show on the drawing because the field angle is too small. Setting the *Image space rays* field to *image srf*, the number of field points equal to 1 and the number of rays to 9 from an on-axis point results in the following drawing.



Cooke triplet

This lens has been used for many years to demonstrate OSLO. It is discussed throughout the manual in a wide variety of contexts. As a historical note, this lens was not designed using OSLO, but is rather an adaptation of a design done by R. Kingslake and included in his book “Lens Design Fundamentals”, pp 286-295, (Academic Press 1978, ISBN 0-12-408650-1). As another historical note, this design form was actually invented by H. Dennis Taylor, who worked for Cooke. According to Kingslake, the lens was not manufactured by Cooke, but rather by Taylor-Hobson, another optical firm whose principals were not related to Taylor, despite the name.
In any case, this form has been used in millions of low cost cameras, projectors, and various other instruments. The design here is typical and can be used as a starting system for adaptation. You can compare its performance to the dblgauss.len and petzval.len designs included in the OSLO demo library. The basic ray and MTF analyses for this lens are as follows.





Digital Triplet

Lenses like the above Cooke triplet are typically designed for film cameras. The above lens has a 50mm focal length, which suits the 24x36mm format used for 35mm cameras. The lens is not optimum for a typical digital camera, which has a smaller image size, because the focal length is too long. In this example we consider the use of the GENII error function to redesign the lens to have a focal length of 10mm, more suited to contemporary digital cameras. In addition, we will make the lens faster by 3dB, so that the paraxial specifications become 50mm efl, f/2.8, 20 degrees field angle.

The steps are the following:

- Open the demotrip lens, open the surface data spreadsheet, and right-click in the spreadsheet to pop-up the edit menu. Scale the lens to a focal length of 10mm. Then, set the entrance beam radius to 1.78571mm. Note that this makes the speed f/2.8 (use Pxc in the text window).

- Change the apertures of all the surfaces to 1.8 mm so the larger beam can get through. With the scaled lens, the elements are too thin to support an aperture of 1.8. Increase the thickness of the front and back elements to 0.7mm, and the thickness of the center element to 0.3mm. Now the on-axis beam can get through, but the focal length is no longer 10mm because the thicknesses have been changed.

- To fix the focal length, put an axial ray angle solve on the last refracting surface (6), setting the angle to -0.178571. This will adjust the curvature so the focal length is exactly 10mm again. As soon as you have adjusted the curvature, remove the solve by clicking the button in the radius cell and selecting *Direct Specification*. The curvature will not be changed, but the constraint will be removed, and the curvature can then be used as a variable in optimization.

The GENII error function is designed to hold the paraxial properties of the system at the values that exist at the start of optimization. It is essential that you have the right paraxial properties (focal length, aperture, and field) prior to beginning to optimize. Fortunately, you have just set up the system so that it has the correct propertures - efl = 10, fnb = 2.8, ang = 20.

- Click on the Variables button in the surface data spreadsheet and make all the curvatures variable. Also make the air spaces variable. Close the variables spreadsheet but not the surface data spreadsheet. You will see "V's" on the appropriate buttons.

- Use Optimize>>Generate Error Function>>GENII Ray Aberration to enter the error function. Accept all the defaults in the dialog box without change.

- Change the Lens ID to something more descriptive of the current system, and save the lens in your private directory under a new name. The final solution in the public demo library is called digitrip.len, so pick a related but different name. Then close the Surface data spreadsheet using the Green check and immediately re-open it. This allows you to cancel unfortunate changes that might be made during optimization by canceling using the Red X button. After you have re-opened the spreadsheet (turn on Autodraw so you can see what is happening to your lens), you are ready to optimize it.

- Use the Ite button in the Text window to iterate the design until it doesn't improve anymore. This should produce an error function of about 0.64.

- Change the center element from F4 to a model glass, leaving the index and v-number unchanged, by clicking the button in the Glass column and selecting Model. Then click the button again and make it a special variable (RN only). In the variable spreadsheet, set boundaries of 1.5 to 1.8 for the RN variable. Close the variables spreadsheet.

- Change the glass in the front and back elements to LAK33.

Later, you can experiment yourself with varying these glasses, but for now, accept that LAK33 is a reasonable choice. The glass variation procedure will be simplified to finding the correct matching flint for these crown elements. If and when you experiment with varying the front and back elements, use both RN and DN. In the variables spreadsheet, put boundaries of 1.5 to 1.8 on the RN's, and 0.0 to 1.0 on the DN's.

- Close the surface data spreadsheet and re-open it (so you can return to this state if things go awry). Then re-iterate the design until it doesn't improve (error function ~ 0.39).

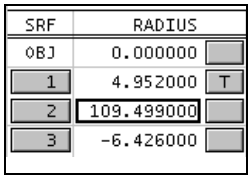
- Use the Fix option on the first glass button to find the closest Schott glass to the variable glass. You should find SF1. Click OK to accept this result.

- Re-optimize after you have fixed the glass.

- After you have fixed the glass, work through the variables, one at a time, choosing a nearby rounded off value, then making it a direct selection instead of a variable. Start with TH[2] = 1.2, re-optimize, TH[4] = 1.1, re-optimize. Don't fix the last thickness yet.

- To fix the radii, you can either round off to nearby values, or use a test glass list. If you round off, try to use only one place after the decimal for the thicknesses, and 2 places after the decimal for the radii. Re-iterate after each step.

- To use a test glass list, first convert the radius to a Direct specification, then hold down the SHIFT key while clicking on a radius value twice. The field will change into an outline box, and the nearest radius from the current test glass list will be shown. Click on the button to accept the value, or SHIFT+Click above or below the box to see the next higher or lower value in the list (specified by the preference *tglf*). If the error function cannot be restored close to its original value, experiment with fixing other radii first. Fitting to a test glass list involves a certain amount of trial and error, as well as user discretion as to what is most important. In fitting the current lens, we were able to use the same test glass for three surfaces (see the final listing below).



- Adjust the aperture sizes so they just transmit the axial beam. The front element will be 1.8, the center 1.5, and the back 1.6.

- Set TH[6] to some nominal value (TH[6] = 8.1), and click the thickness button for the image surface to establish the final focus position using the minimum polychromatic axial spot size criterion.

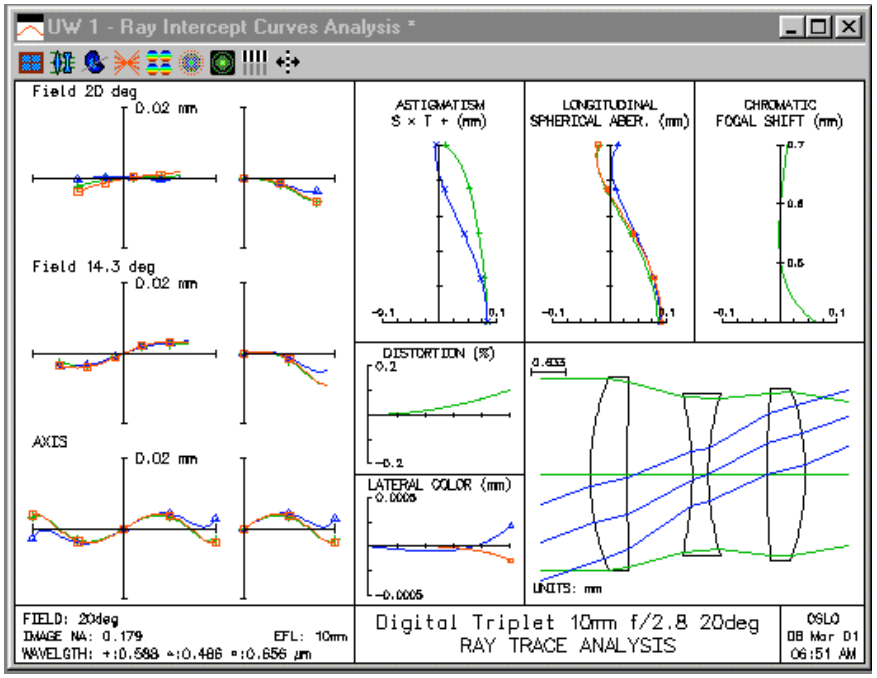
When you are done with this process, you will have completed the design. The solution that you obtain should be generally similar to (but possibly not identical to) the following. Note that the scales have been adjusted by a factor of 5.0 to account for the difference in focal length. In spite of the increased speed, the new design is generally similar to or better than the old. The final design is included in the OSLO demo/Lt library as digitrip.len.

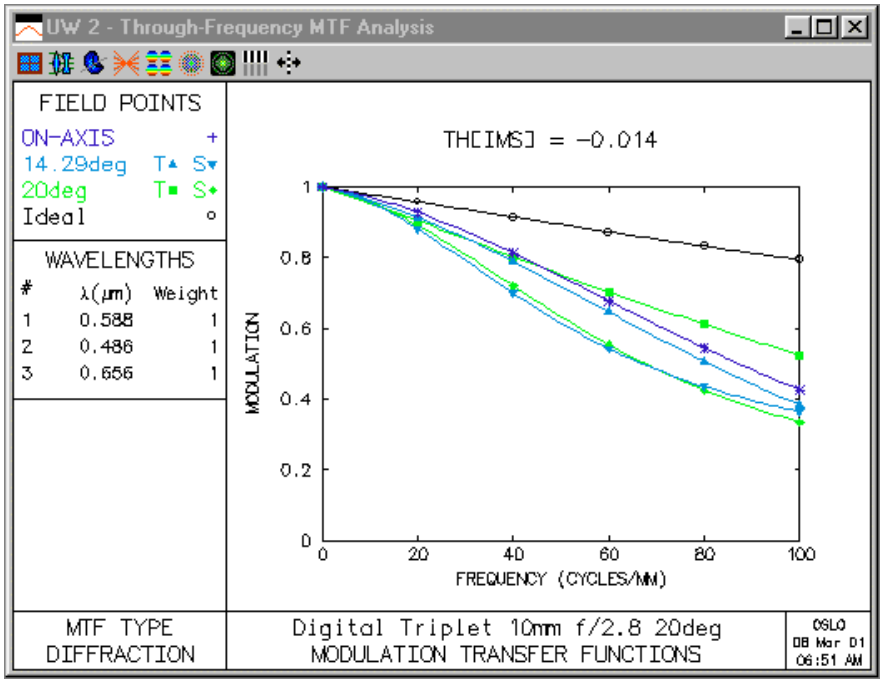
Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens:Digital Triplet 10mm f/2.8 20deg | Zoom 1 of 1 | Efl 9.999385

Ent beam radius 1.785710 | Field angle 20.000000 | Primary wavln 0.587560

SRF	RADIUS	THICKNESS	APERTURE RADIUS		GLASS		SPECIAL
OBJ	0.000000	2.0000e+19	7.2793e+18		AIR		
1	4.952000	0.700000	1.800000	K	LAK33	C	
2	109.400000	1.200000	1.800000	P	AIR		
3	-6.426000	0.300000	1.500000		SF1	C	
AST	4.952000	1.100000	1.500000	AP	AIR		
5	25.022000	0.700000	1.600000		LAK33	C	
6	-4.952000	8.100000	1.600000	PK	AIR		
IMS	0.000000	-0.014000	3.624407	S			

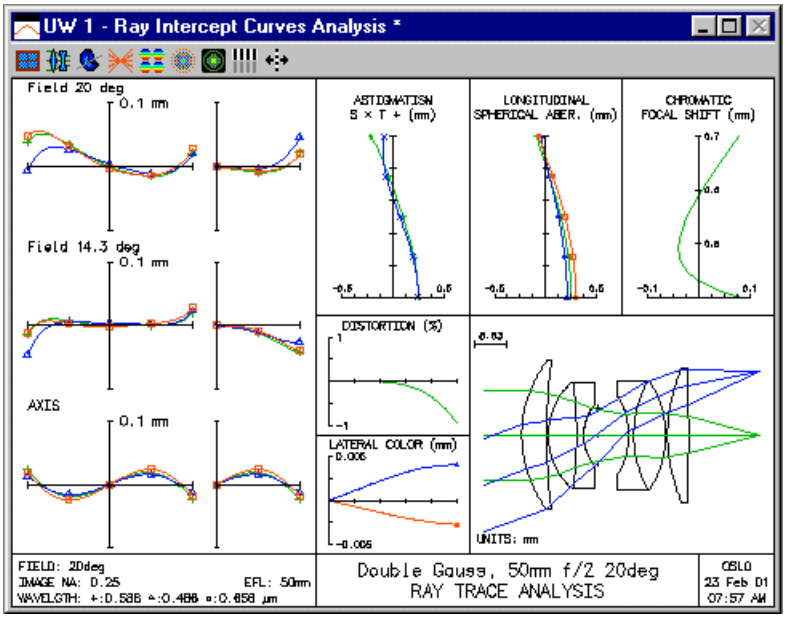




Double-Gauss objective

The double-Gauss design form has dominated the class of photographic lenses for many years. There are literally thousands of adaptations of this form, which has an apt combination of aperture, field, and design complexity. The design shown here is just an ordinary double gauss, using conservative specifications, that you can use as an indicator of typical performance as well as a starting system for a new design. It is set up to work at 50mm efl, f/2, with a field of about ±20 degrees. You can compare this to the Cooke triplet (demotrip.len) and Petzval lens (petzval.len) to see the relative performance.

One possibly interesting aspect of the current design is that it has no vignetting. The present system, and the design procedure described below, are designed to produce optimum performance under the nominal conditions. Most photographic lenses are not designed to have peak performance at full aperture. In photographic applications, vignetting is usually used to improve performance by truncating the oblique spherical aberration that limits the field coverage of the double Gauss lens.



The double-Gauss objective shown above serves as a starting system to illustrate the techniques for lens optimization with the GENII error function. The lens is to be optimized for f/2, 50mm focal length, ±20 degrees field coverage. The results will be different from the above system because the example system was created using a different optimization procedure.

The example is presented as a series of explicit steps that you should duplicate on your computer. The OSLO user interface has been especially constructed to provide an easy-to-use interaction during optimization, if you follow the recommended procedure. As you progress through the design, you should save your lens so that you can recover to a given point rather than start over, if you make a mistake. The public/len/demo/light directory contains lens files showing the design at various stages. You should expect your lens data to be similar, but not necessarily identical, to that in the tutorial files.

Starting Design (dblgauss0.len)

- Click File >> Open and open the lens public/len/demo/light/dblgauss.len.
- Click File >> Save As and save the lens as private/len/dblgauss0.len. This will make a backup copy in your private directory that you can reopen if necessary.

Curvature Solution (dblgauss1.len)

- Click the F5 tool bar icon to open the surface data spreadsheet. Make sure that Autodraw is set to On, that the entrance beam radius is 12.5mm, the field angle 20 degrees, and the focal length 50.000041.

- Click on the aperture buttons for surfaces 1 and 11 and set the aperture specification to Not Checked.

- Click Optimize >> Variables and click on the Vary All Curvatures button. Click the OK button (or the check button in the spreadsheet) to close the Variables spreadsheet.

- Click Optimize >> GENII Error Function to enter the error function with default values (Note: If you enter the command geniierf_lt ?, OSLO will allow you to enter custom values for the default ray data). Note that the Target icon (Shift+F10) is enabled after you give the command, signifying that the lens has operands.

- Click the Target icon (Shift+F10) and observe the initial error function is 2.35.

- Close (i.e. OK) the lens (surface data) spreadsheet and immediately reopen it. This has the effect of saving the current lens to a temporary file, which you can recall using the Revert capability of OSLO (If you click Cancel in a spreadsheet, the program will prompt "Undo all changes ?"). You should always optimize a system with the lens spreadsheet open so that you can back up if something goes awry during the optimization process. Also, if the spreadsheet is open and Autodraw is on, the program will update the spreadsheet and draw a picture of the lens after each series of iterations.

- Click the Shift+F9 (Bow & Arrow) icon to iterate the design by 10 steps. Repeat this five or six times until the merit function goes down to about 1.44.

- Change the lens identification to "CV Solution".

- Click the Ray Analysis report graphics toolbar icon (Shift+F3) to display a ray trace analysis in the graphics window. Note that the lens still looks similar to the starting system, and that the ray curves, although different, are still reasonable.

- Click File >> Save As and save the lens as dblgauss1.len. If you want, you can compare your system with the lens of the same name in the public/len/demo/tutorial directory.

At this stage, we have only used curvatures as variables. This has the advantage that the system is still in the same general solution region as the starting system, and the disadvantage that the performance is not improved very much. The next stage of the design is to add the thicknesses as variables. When you add thicknesses, you must take extra care to provide boundary conditions to prevent the system from "blowing up", i.e. wandering off to a solution that is either non-physical or in a totally different solution region from the starting system.

Thickness Solution (dblgauss2.len)

- Click Optimize >> Variables. At the top of the spreadsheet, change the data fields so the air-space thickness bounds run between 0.5 and 25mm, and the glass thickness bounds run between 1 and 15mm.

- Click the Vary All Thicknesses button, then close (accept) the Variables spreadsheet.

- Close (accept) the lens spreadsheet, then immediately reopen, to provide a revert file.

- Click the Bow & Arrow (Shift+F9) icon once (only). Look at the lens picture made by Autodraw. Note that the first element is much too thin, resulting in a "feathered" edge, even though the axial thickness is within its bounds. Now look at the spreadsheet data and notice that the thickness of surface 2 is less than its minimum boundary (0.5mm). The feathered edge problem requires us to put a boundary condition on edge thicknesses,

which we will do using "Edge Contact" solves. The minimum axial thickness violation requires us to put increased weight on boundary conditions.

- Click the Cancel button in the lens spreadsheet. When the program pops up the "Undo all Changes?" box, click OK in the box. The lens will be restored to its state when the spreadsheet was opened (i.e. right before the iteration). Reopen the lens spreadsheet, and note that this is so.

We are going to put edge contact solves on all the positive elements (1,3,8,10). By choosing the aperture height at which we specify edge contact, we can start off with roughly the same axial thicknesses as the original system. Note that this is only one of the ways to control edge thickness, but it is quick and easy. Later in the design, you can free up the thicknesses and see if they will move to physically reasonable values.

- Click the Thickness button for surface 1, then click Solves, and then Edge Contact. In the box that asks for the edge contact radius, enter 23.0.

- Click the Thickness button for surface 3, then click Solves, and then Edge Contact. In the box that asks for the edge contact radius, enter 16.0.

- Click the Thickness button for surface 8, then click Solves, and then Edge Contact. In the box that asks for the edge contact radius, enter 16.0.

- Click the Thickness button for surface 10, then click Solves, and then Edge Contact. In the box that asks for the edge contact radius, enter 22.0.

- Click Optimize >> Operating Conditions. Find the "Weight of boundary condition violations" field, and enter 1.0e4. This will force any boundary violations to have much more importance than other operands, and hence to be well controlled.

- Click OK to close the Operating conditions spreadsheet.

- Click OK to close the lens spreadsheet, then immediately reopen it.

- Click the Bow & Arrow icon about ten times, which should lower the error function to about 1.05.

- Change the lens identification to "CV/TH Solution".

- Click File >> Save As and save the lens as dblgauss2.len.

- Click the Ray Analysis Report Graphics (Shift + F3) icon. Note that although the error function is better than before, that the various ray curves are still not very good. In order to make the design better, it will be necessary to use different glasses.

For the design exercise in this tutorial, we will just vary the glasses in the inner doublets. After trying this, you can proceed on your own to make a final design by varying all the glasses. To vary glasses, it is necessary to first replace the catalog glasses with model glasses.

Variable Glass Solution (dblgauss3.len)

- In the lens spreadsheet, click on the glass button for surface 3, then click on Model. In the dialog boxes that appear, click OK three times to accept the Glass name, base index, and V-number for the model glass. After you have finished, note that the letter M appears on the glass button. Click the glass button for surface 3, and select Variable from the pop-up menu of actions.

- Repeat the above step for surfaces 4, 7, and 8.

OSLO normally treats glasses as one-dimensional variables. When you vary a glass, its index and dispersion are simultaneously varied so that the glass stays on the "glass line". For many types of lenses, this is an adequate scheme for finding an optimum glass quickly. For a lens like the double Gauss, however, the glasses in the interior doublets need to have both the refractive index and dispersion varied independently. To do this, you enter additional variables "dn" on each of the surfaces.

- Click Optimize >> Variables. Select Row 21, then click the "Insert After" toolbar icon in the Variables spreadsheet. Then select Row 22, and hold down the Shift key while clicking on the row button three more times. This will create 4 new rows at the bottom of the variables spreadsheet.

- In the Surf column of the new rows, insert surfaces 3, 4, 7 and 8 respectively.

- In the Type column, insert DN for all 4 rows.

For both refractive index and dispersion, it is necessary to enter explicit boundary conditions. Otherwise, the program may try to create glasses with infinite index and zero dispersion. The refractive index should be set between 1.5 and 2, and the dispersion should be allowed to range from 0 to 1.0. The dispersion factor DN is normalized so that 0 corresponds to the left-hand edge of the glass chart, and 1.0 corresponds to the right hand edge of the glass chart.

- Enter the above boundary conditions in the appropriate cells in the eight rows that specify the glass variables.

- Close the Variables spreadsheet, then close the lens spreadsheet and immediately reopen it.

- Click the Bow & Arrow toolbar icon several times until the error function decreases to about 0.47.

- *There is a substantial reduction in the error function from varying the glasses, but the glasses are now fictitious. You can update the graphics window to check the current Ray Analysis.*

- Click Show >> Surface Data, then select Refractive indices and close (accept) the dialog box. The current refractive index data will appear in the Text output window. Note that the glass on surface 8 has a refractive index of 2.00 and a V-number of 50. This will be a difficult glass to match.

- Change the lens ID to "CV/TH/RN Solution", then click File >> Save As to save the current lens as dblgauss3.len. Close the lens spreadsheet and reopen it immediately.

The next task is to find real glasses to replace the model glasses. OSLO has a special command, accessible from the lens spreadsheet, that substitutes the nearest real glass for a model glass.

Real Glass Solution (dblgauss4.len)

- Click the glass button on surface 8, and select Fix >> Schott from the list of options. The program should offer to substitute LASFN31. Accept this suggestion. Close and reopen the lens spreadsheet. Click the Ite button many (~40?) times until the error function decreases to about 0.49.

- Click the glass button on surface 7, and select Fix >> Schott from the list of options. The program should offer to substitute SF9. Accept this suggestion. Close and reopen the lens spreadsheet. Click the Ite button until the error function decreases to about 0.50.

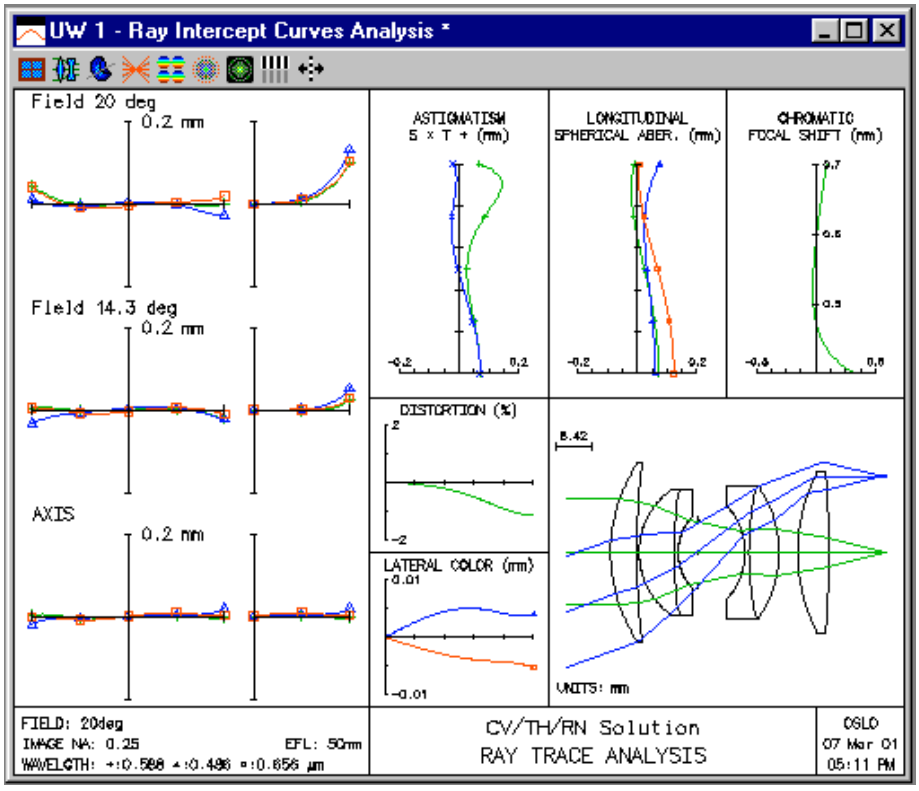
- Click the glass button on surface 4, and select Fix >> Schott from the list of options. The program should offer to substitute SF59. Accept this suggestion. Close and reopen the lens spreadsheet. Click the Ite button until the error function decreases to about 0.50.

- Click the glass button on surface 3, and select Fix >> Schott from the list of options. The program should offer to substitute LASF18A. Accept this suggestion. Close and reopen the lens spreadsheet. Click the Ite button until the error function decreases to about 0.70.

Note that it is not possible to get a good match with LASF18A glass. Possibly this indicates that additional optimization should be carried out with the other glasses as variables.

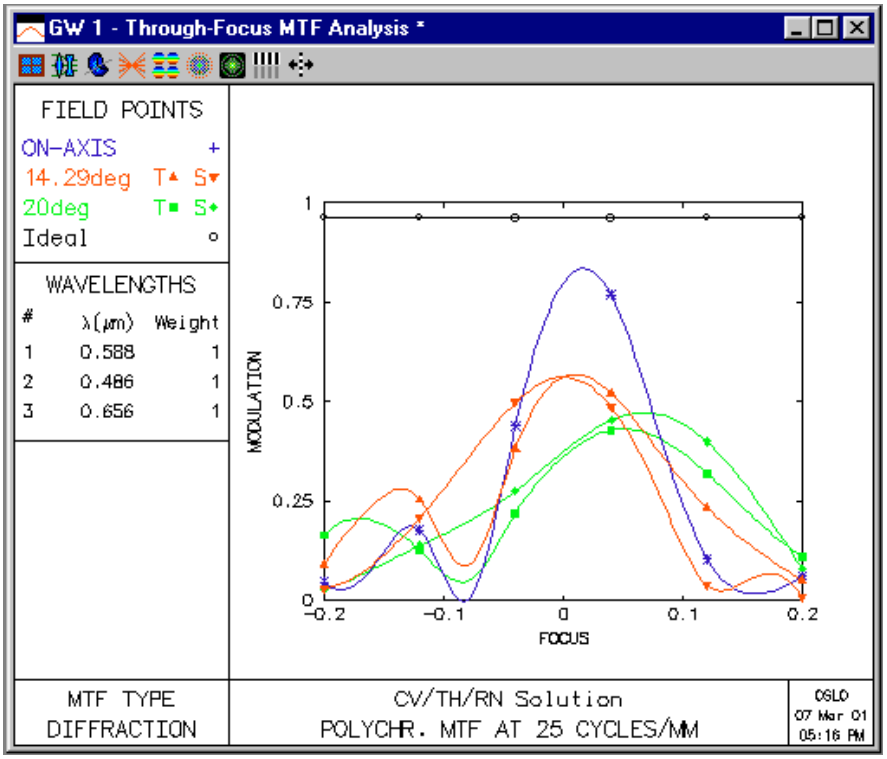
- Click File >> Save As to save the current lens as dblgauss4.len. Close the lens spreadsheet and reopen it immediately.

- Double-click the report graphics window to bring the ray analysis up to date. The curves should be similar to the following. Note that the sagittal ray-intercept curve at full field is not well controlled. Note also that the astigmatism curves at the edge of the field do not come together.



Open a second graphics window using Window >> Graphics >> New. Click the report graphics Through-focus MTF (Shift F6) icon. The curves should be similar to the following. Note that

despite the fact that the ray analysis astigmatism curves do not come together at the edge of the field, the actual MTF curves are reasonably coincident, at least at 25 cycles/mm.

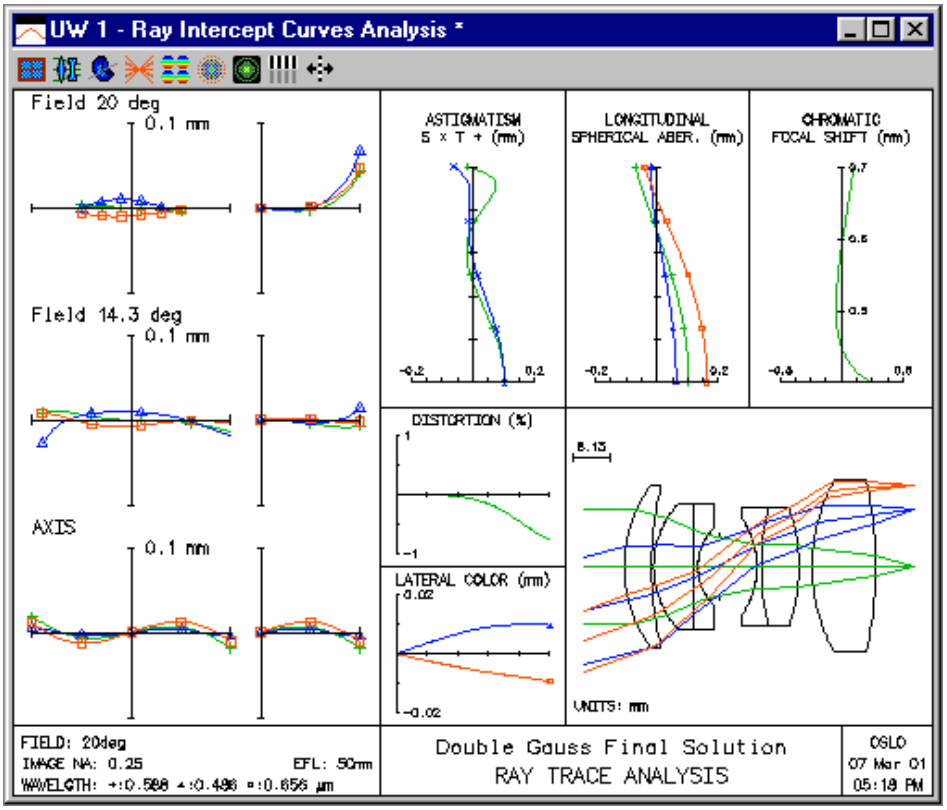


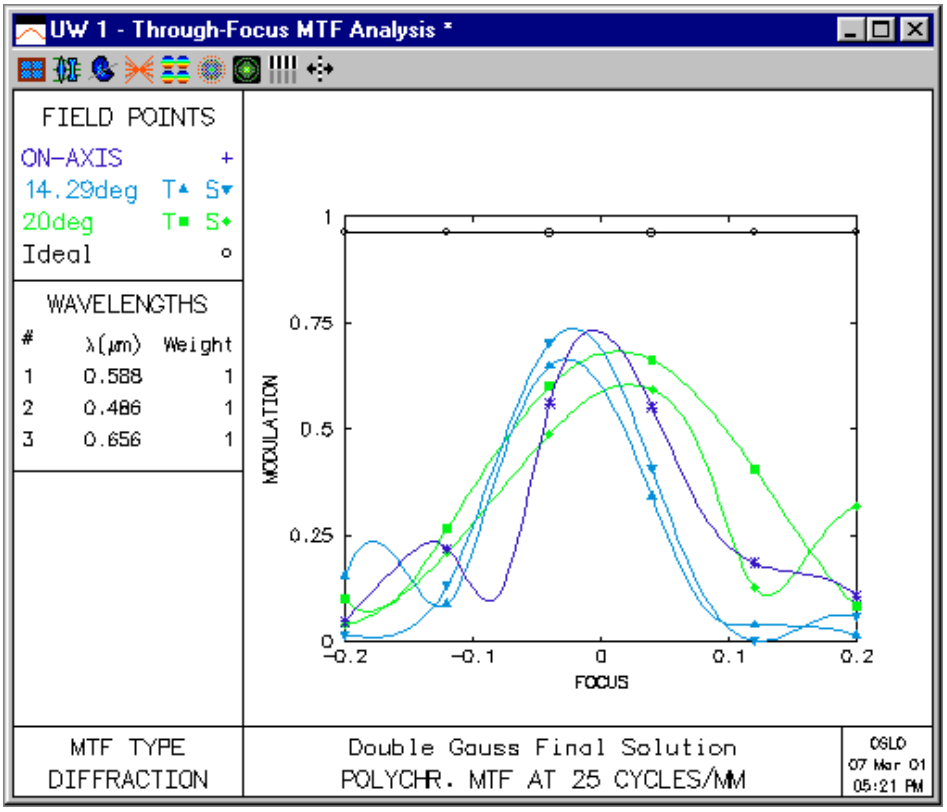
At this point, the optimization has proceeded to the point of a default solution. Now, various trials can be conducted to improve the design, such as trying additional glasses. Other possibilities for additional optimization are to remove the edge contact solves to see whether the positive elements still want to get too thin, to change the weights on selected terms in the error function, or to re-enter the error function with different rays.

The general approach to optimization using OSLO is illustrated by the steps of this example: you should approach the optimization cautiously and change only a few things at a time, working interactively, until you are confident that the combination of variables and operands can be trusted to produce a system of high quality. You should always maintain a way to restore your system to an earlier state, such as using the revert capability of OSLO spreadsheets.

Final Solution (dblgauss5.len)

The lens shown below (public/len/demo/tutorial/dblgauss5.len) is the result of about an hour's investigation of various options for improving the design using the GENII error function (with different weights on selected operands). It is not feasible to trace the course of this optimization explicitly. You should try to see if you can match, or improve on, the final design shown below. Note that checked apertures have been inserted to provide vignetting at the edge of the field.


UW 1 - Ray Intercept Curves Analysis *
Field 20 deg
0.1 mm
Field 14.3 deg
0.1 mm
AXIS
0.1 mm
ASTIGMATISM
S x T + (mm)
LONGITUDINAL
SPHERICAL ABER. (mm)
CHROMATIC
FOCAL SHIFT (mm)
DISTORTION (%)
LATERAL COLOR (mm)
UNITS: mm
FIELD: 20deg
IMAGE NA: 0.25
EFL: 50mm
WAVELGTH: +:0.588 *:0.486 ▪:0.656 µm
Double Gauss Final Solution
RAY TRACE ANALYSIS
OSLO
07 Mar 01
05:18 PM


UW 1 - Through-Focus MTF Analysis *
FIELD POINTS
ON-AXIS +
14.29deg T▲ S▼
20deg T■ S◆
Ideal ○
WAVELENGTHS
λ(µm) Weight
1 0.588 1
2 0.486 1
3 0.656 1
MODULATION
FOCUS
MTF TYPE
DIFFRACTION
Double Gauss Final Solution
POLYCHR. MTF AT 25 CYCLES/MM
OSLO
07 Mar 01
05:21 PM

The data for the final system are shown below. Note that the "outer" glasses have been changed. Note also that the radii and thicknesses have been rounded to meaningful values. This should be the final step in an optimization run. Each variable in turn is set to a rounded off value and removed from the variable list. The lens is then re-optimized using the remaining variables. A similar procedure is used to fit radii of curvature to test glass values.

Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: Double Gauss Final Solution | Zoom 1 of 1 | Efl 50.009117

Ent beam radius 12.500000 | Field angle 20.000000 | Primary wavln 0.587560

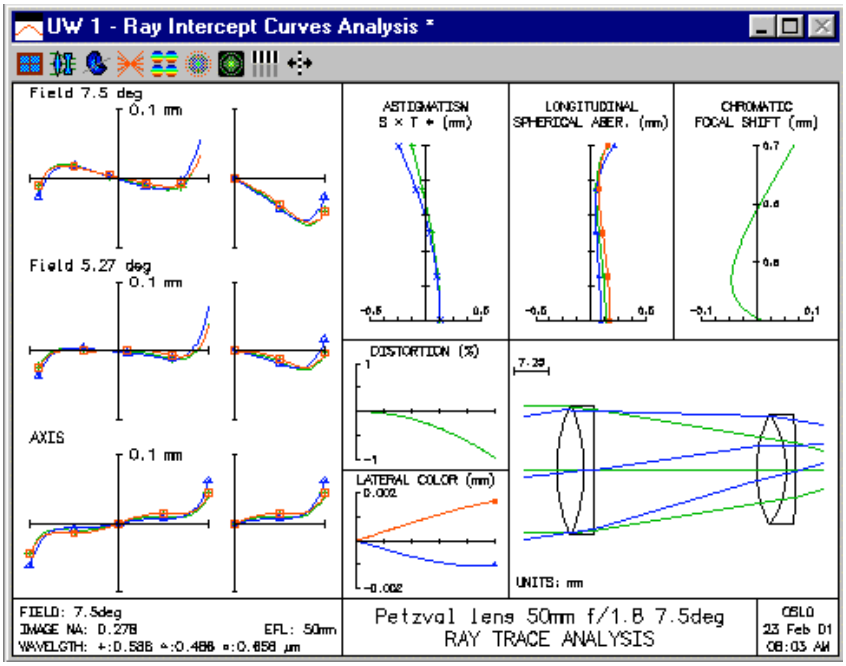
SRF	RADIUS	THICKNESS	APERTURE RADIUS		GLASS		SPECIAL
OBJ	0.000000	1.0000e+20	3.6397e+19		AIR		
1	30.660000	6.000000	18.000000	K	LASF35	C	
2	77.800000	0.500000	18.000000		AIR		
3	20.600000	8.750000	14.000000		LASFN31	C	
4	0.000000	1.000000	14.000000		SF59	C	
5	12.420000	4.800000	8.500000		AIR		
AST	0.000000	8.200000	6.400000	A	AIR		
7	-18.930000	1.000000	11.000000		LF5	C	
8	59.600000	8.300000	13.000000		LASFN31	C	
9	-40.490000	2.900000	13.000000		AIR		
10	40.000000	14.000000	19.000000		BASF52	C	
11	-87.900000	8.700000	19.000000	K	AIR		
IMS	0.000000	0.000000	18.500000				

Petzval Lens

The Petzval lens is a very old design form (> 150 years!) that is still a mainstay in lens libraries. The original Petzval Portrait lens used a cemented doublet and air-spaced doublet, but the term Petzval lens is now generally applied to lenses containing two separate groups (usually doublets) in which both groups contribute positive power.

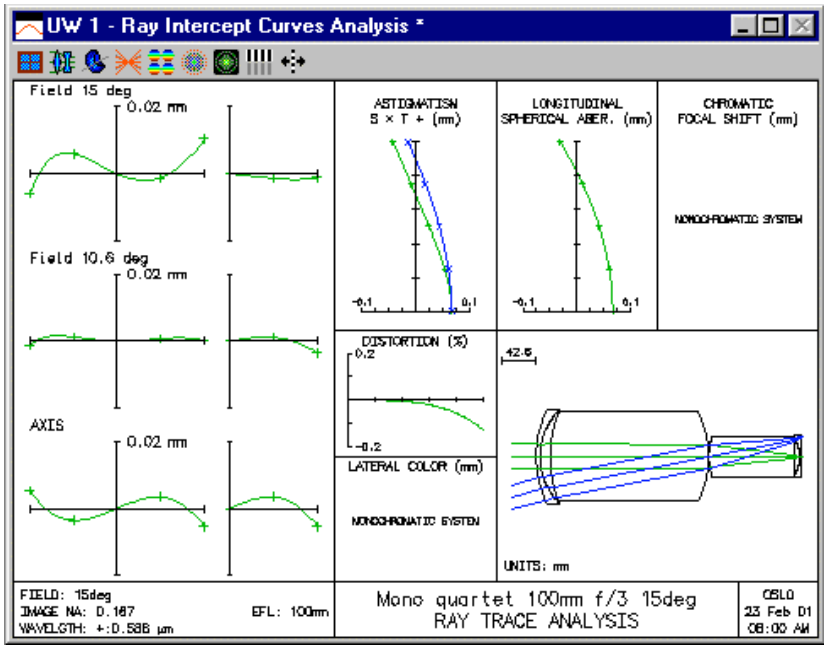
The Petzval lens is a good design form for high-aperture narrow field applications. Curiously, although invented by Petzval to improve the field coverage of high-aperture systems, it makes no attempt to correct the Petzval curvature. In modern designs, the Petzval lens often incorporates a negative element near the image plane to flatten the field.

The lens included here is a typical design that can be used as a starting design for specific modification. It is scaled to 50mm focal length, so you can compare its performance to demotrip.len and dblgauss.len.



Monochromatic quartet lens

The monochromatic quartet is the name given by organizer D.C. O'Shea, to a lens that was designed according to the rules of the 1990 International Lens Design Conference contest. The lens was specified according to the requirement that it be the best design having a focal length of 100mm, a speed of f/3, and a field of 15 degrees, using four elements of BK7 glass.
Of the 44 designs submitted, the top five were essentially the same (three were designed using OSLO), and it is speculated that this design is the global solution to the problem as stated. Since then, global optimization algorithms have often been tested to ensure that they find this solution.

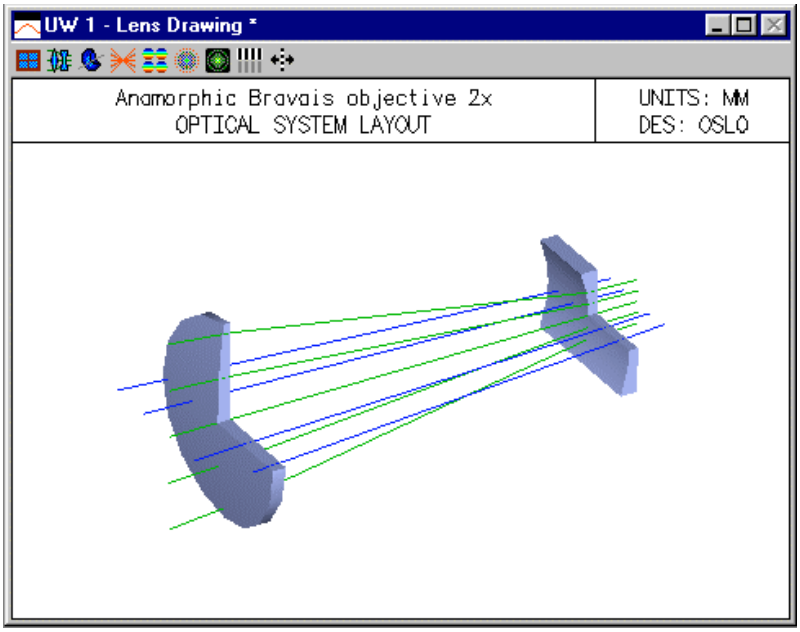


Anamorphic telextender

This example shows a simple Bravais system, which is a system where the object coincides with the image. The system goes on the back end of an ordinary lens to create a sharp image with different magnification in the *x* and *y* directions. This is accomplished using cylindrical lenses with power in the *yz* meridian.

The Bravais condition is imposed using a **pk lnm** command on surface 4. Actually, the condition cannot be *precisely* satisfied using cylindrical lenses because the thickness of the lenses displaces the image in the *xz* meridian. Therefore, the lens powers were chosen to make the image in the *yz* meridian lie on the same surface as the *xz* image.

Note the use of special apertures on surfaces 3 and 4 to model the rectangular lens.

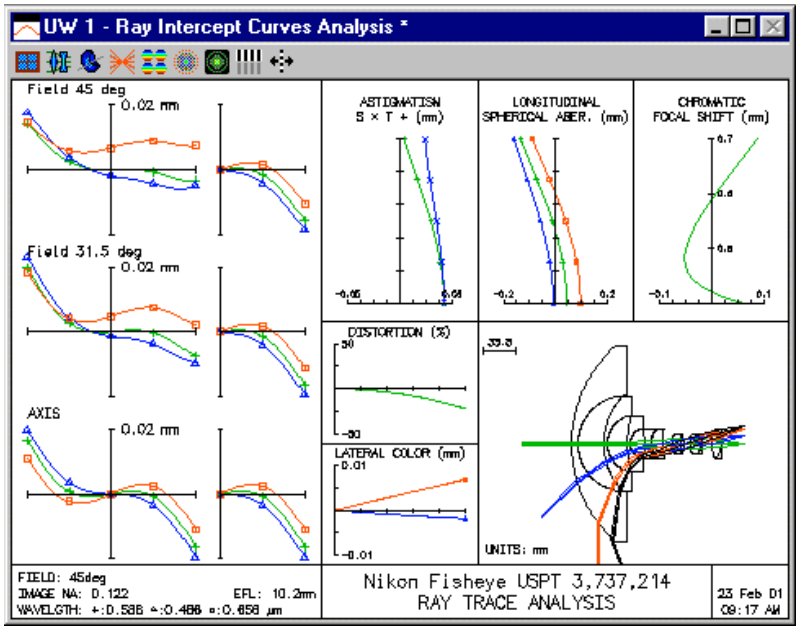



```
*PARAXIAL CONSTANTS
   Effective focal length:   -4.160855    Lateral magnification:     2.019103
   Numerical aperture:        0.049281    Gaussian image height:    -1.009551
   Working F-number:         10.145865    Petzval radius:            4.269440
   Lagrange invariant:        0.050000

*PARAXIAL CONSTANTS - XZ PLANE
 FRACTIONAL XZ APERTURE       1.000000    FRACTIONAL XZ OBJECT       1.000000
   Effective focal length:  5.0000e+39    Lateral magnification:     1.000000
   Numerical aperture:        0.099504    Gaussian image height:    -0.500000
   Working F-number:          5.024938    Petzval radius:          1.0000e+40
   Lagrange invariant:        0.050000
*LENS DATA
Anamorphic Bravais objective 2x
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        -10.000000      0.500000             AIR

  1      8.462000      0.150000      1.100000 A          SK16 C *
  2        --          4.100000      1.100000             AIR

  3     -1.370000      0.100000      0.943398 X          SK16 C *
  4        --          5.650000 P    0.943398 PX          AIR

  5        --            --          1.050000

*SURFACE TAG DATA
  1     CVX       --
  3     CVX       --

*PICKUPS
  4     LNM    1   4   10.000000
```

```
  4     AP     3
  4     SAP  A   3 A

*APERTURES
 SRF   TYPE APERTURE RADIUS
  0     SPC     0.500000
  1     SPC     1.100000
  2     SPC     1.100000
  3     SPC     0.943398
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -0.800000  AX2     0.800000  AY1    -0.500000  AY2     0.500000
  4     PKP     0.943398
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -0.800000  AX2     0.800000  AY1    -0.500000  AY2     0.500000
  5     SPC     1.050000
```

Fisheye lens

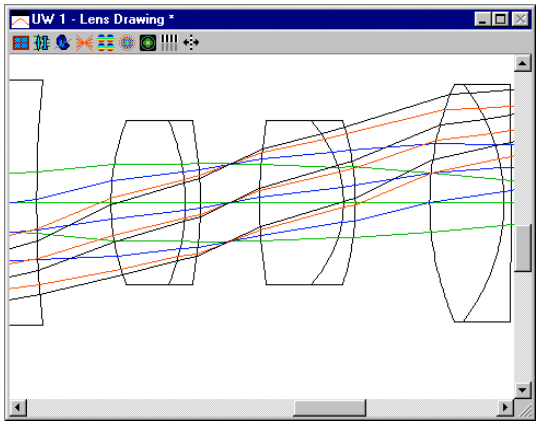
Lenses that have fields of view of greater than 90 degrees (half-angle) require special attention in setup, since the conventional methods for specifying field of view are awkward to handle for this case. In addition, special care must be taken within the ray trace routines to ensure stability. Finally, care must be taken in evaluating such lenses, since many of the built-in evaluation procedures (e.g. report graphics) implicitly assume a field of view of less than 90 degrees. The nikofish.len file is an example of such a lens. It is based on a Nikon patent, and is implemented here as a 10mm efl lens that covers a field of view of 108 degrees with an aperture of f/4.



As you can see from the drawing, the aperture stop of the lens is well back in the lens, on surface 14. Obviously the pupil position is a function of the field angle, moving almost to the first surface at the edge of the field. In order to accommodate this type of system and ensure that the rays pass through the desired positions on the aperture stop, OSLO uses the **warm** (wide-angle ray-aiming mode) general operating condition.

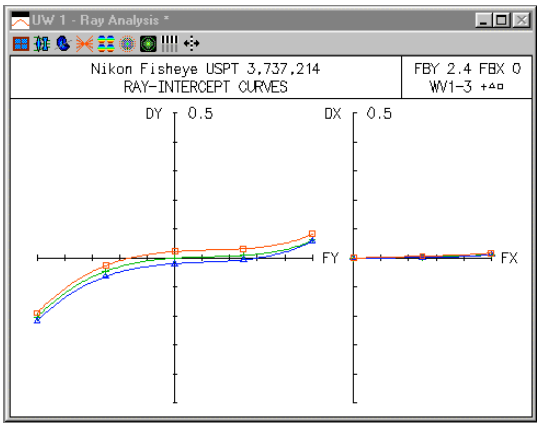
When **warm** is on, fractional object coordinates (for infinite conjugate systems) represent angles in degrees, relative to the nominal field angle of the system. For the present system, the field angle is specified as 45 degrees, so a fractional object height of 1.0 refers to a point 45 degrees off axis. The edge of the field of view of this system occurs at a fractional object height of 2.4.

In addition, when **warm** is on, fractional pupil coordinates refer to positions in the reference surface, normalized to the reference surface radius (not the entrance beam radius). For the nikofish lens, the reference surface (aperture stop) radius is 5.2 mm, so a fractional aperture coordinate FY = 0.5 refers to a point in the aperture stop 2.6 mm from the optical axis. The figure below shows a zoomed drawing of the lens near the stop surface. You can see that the rays always go through their prescribed points in the aperture stop. In this connection, it should be mentioned that it is not necessary to have **warm** on to have the chief ray go through the center of the stop: this always happens (unless the ray trace fails for some reason). The **warm** condition only affects ordinary rays.



To evaluate fisheye systems, you can use all the routines that use the current field point, which can be set to a value greater than 1.0 using the **sop** command. For example, a ray intercept curve at the edge of the field of view can be obtained as follows:

```
*SET OBJECT POINT
         FBY          FBX          FBZ
       2.400000       --           --
         FYRF         FXRF         FY           FX
          --           --           --           --
          YC           XC           YFS          XFS          OPL     REF SPH RAD
      18.702850        --         0.219863     0.062672   182.326549    Infinite
```



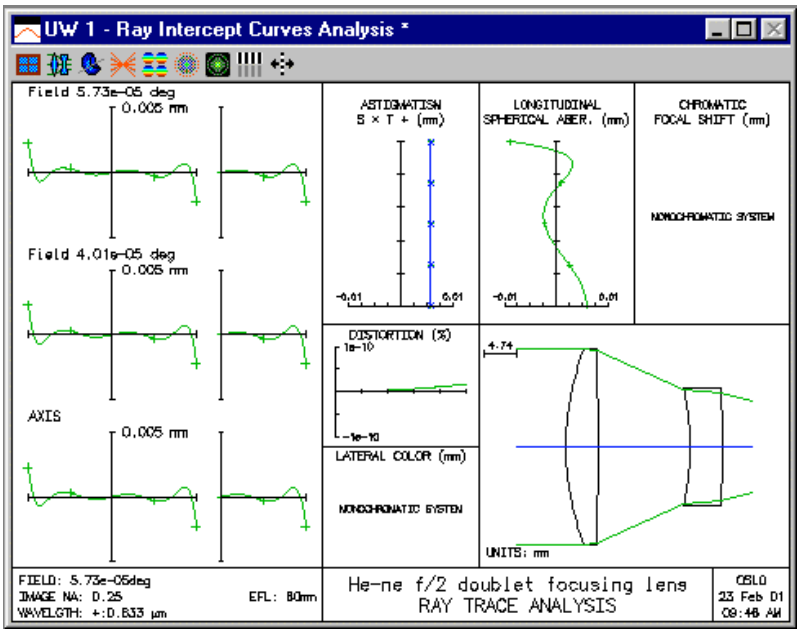
There is some problem in obtaining a graphical field analysis, because all the internal routines extend only to FBY = 1.0. However, the star command *field accommodates two arguments that give the number of field points and the maximum fby, respectively. The command ***field 12 2.4** produces the following output.

```
*FIELD ANALYSIS
 WAVELENGTH 1
      FIELD         YC           YFS          XFS          % DIST      LAT COLOR
       --           --          0.042414     0.042414       --           --
    0.200000     1.606980      0.040257     0.041150     -0.858303    -0.001824
    0.400000     3.210651      0.034107     0.037644     -3.444843    -0.003608
    0.600000     4.807926      0.024917     0.032730     -7.795877    -0.005325
    0.800000     6.396158      0.014212     0.027698    -13.976597    -0.006981
    1.000000     7.973349      0.004033     0.024151    -22.088918    -0.008626
    1.200000     9.538334     -0.003140     0.023787    -32.283884    -0.010379
    1.400000    11.090926     -0.004421     0.028098    -44.780543    -0.012438
    1.600000    12.631948      0.003501     0.037920    -59.894417    -0.015109
    1.800000    14.163058      0.024490     0.052703    -78.080634    -0.018818
    2.000000    15.686059     -0.177583     0.069185   -100.000000    -0.024115
    2.200000    17.201146      0.125809     0.078828   -126.621245    -0.031608
    2.400000    18.702850      0.219863     0.062672   -159.380288    -0.041751
```

Monochromatic air-spaced doublet

The lasrdblt.len file is designed to focus light (on axis) from a helium-neon laser. You can see from the ray analysis below that the lens far outperforms the Cooke triplet, double Gauss, and

Petzval lenses shown in these examples. On the other hand, it has essentially no field of view, and works only in monochromatic light. These limitations distinguish the present design from normal doublets, which are usually designed to cover an extended spectral range. In the present lens, the degrees of freedom normally used to extend the spectral range were instead used to balance high-order aberrations.



The air-spaced doublet provides a good system to illustrate the thermal or environmental analysis features in OSLO. When the temperature of the lens and its environment is changed, OSLO changes the lens data according to the thermal expansion of the materials (glass and spacers) used to construct it. In addition, refractive indices are recalculated to account for the thermal variation of refractive index.

The temperature of the lens and its surroundings can be set by changing the Temperature value in the General Operating Conditions spreadsheet.

To see the effect of temperature changes on the laser doublet, open the lasrdblt.len file and click the Pxc button in the text window to print the paraxial constants

```
*PARAXIAL CONSTANTS
   Effective focal length:   60.000049    Lateral magnification:  -6.0000e-19
   Numerical aperture:        0.250000    Gaussian image height:   6.0000e-05
   Working F-number:          2.000002    Petzval radius:         -205.107372
   Lagrange invariant:     -1.5000e-05
```

Now open the general operating conditions spreadsheet (click GEN in the surface data spreadsheet) and change the temperature of the system to 100 degrees. Repeat the above command:

```
*PARAXIAL CONSTANTS
   Effective focal length:   59.970699    Lateral magnification:  -5.9971e-19
   Numerical aperture:        0.250122    Gaussian image height:   5.9971e-05
   Working F-number:          1.999023    Petzval radius:         -205.091070
   Lagrange invariant:     -1.5000e-05
```

Now change the temperature back to 20 degrees. You will see that the lens data do not return exactly to their previous values. This is caused by round-off effects in the expansion calculations. You should always save the original system in a file before changing the temperature, or leave the operating conditions spreadsheet open during your analysis and cancel out of it at the end. If you plan extensive thermal analysis, you should consult the **tem** command in the help system or see the Program Reference manual for additional options.

Athermalization

In OSLO, air spaces are expanded by computing the thermal change in the corresponding edge thickness (defined at the aperture radius), and adding this change to the axial thickness. Note that this implies that the axial thickness itself does not affect the thermal expansion.

Radii of curvature are expanded according to the solid material bounding the radius, if the surface separates a solid and AIR. If a surface separates two solid materials, the average TCE of the two solids is used to expand the radius of curvature. This is obviously an ad hoc assumption, and for accurate thermal analysis, an extra surface should be used so that all solids are separated by AIR.

In the case of a mirror, OSLO uses the TCE of the following space to compute the thermal expansion. If the spacer in the following space is not made from the same material as the mirror, it is necessary to add an additional surface in contact with the mirror to accommodate the extra data.

The computation of thermal effects is fairly involved, partly because the data supplied by manufacturers are non-linear and not entirely consistent (e.g. some data are relative to air, and other data are relative to vacuum). One result of this is that it is not possible to exactly reverse a thermal change; there are residual round-off errors in the system parameters. While these are ordinarily not large enough to affect performance, it is a good idea to save a copy of the lens data prior to carrying out a thermal analysis.

The term *athermalization* refers to the process of making a lens insensitive to changes in temperature. When the temperature of the lens and its surroundings is changed, there are two effects that can be modeled by OSLO to account for their influence upon optical performance:

Thermal expansion - When temperature increases, all lengths in the optical system (radii of curvature, axial thicknesses, spacer thicknesses, aspheric and diffractive surface coefficients, and aperture radii) increase (approximately) proportionately, according to the value of the thermal expansion coefficient of each material. Thermal expansion coefficient values are provided in the Schott, Ohara, and Corning glass catalogs, and may be specified for individual glass or air spaces in lenses using the TCE command, as described above.

Thermal variation of refractive index - The refractive indices of optical materials (i.e. glasses) and of air vary with temperature; the index of air (and thus the relative indices of glasses) also varies with atmospheric pressure. Coefficients for the index vs. temperature relation are provided in the Schott glass catalog and can be specified for glasses added to the Private and Shared catalogs.

The temperature of the lens and its surroundings is set by using the **tem** command or by changing the Temperature value in the General Operating Conditions spreadsheet. The syntax of the **tem** command is: tem(*temperature, apply_thermal_expansion)*, where *temperature* is the temperature in degrees Celsius (default = 20 degrees, or room temperature), and *apply_thermal_expansion* is “Yes” to expand all lengths in the lens or “No” to leave the lengths unchanged. If the temperature is changed through the General Operating Conditions spreadsheet, thermal expansion is always applied. Refractive indices are always recomputed when the temperature is changed.

OSLO applies thermal expansion to lenses as follows. First, the radius of curvature, aspheric and diffractive coefficients, and aperture radius of each surface is expanded according to the expansion coefficient of the glass (i.e., non-air) side of the surface; cemented surfaces are expanded according to the average of the two expansion coefficients. Second, the (axial) thickness of each glass (i.e, non-air space) is expanded, also according to the expansion coefficient of the glass. Finally, air spaces are expanded by calculating the change in spacer thickness (which is taken to be the same as the edge thickness) and adding this change to the axial thickness; the expansion coefficient used is that of the spacer material (aluminum, by default).

To see the effect of temperature changes on the laser doublet, open the “public\len\demo\lt\lasrdblt.len” file and print the lens data. The last line of the OPERATING CONDITIONS: GENERAL section of the output shows the temperature in degrees Celsius; the default is 20 degrees, or room temperature. The REFRACTIVE INDICES section lists the glass (medium) for each surface and the value of the thermal expansion coefficient (TCE). Note that the

refractive index of AIR is always given as 1.0, and that the TCE values of the air spaces is that of aluminum (236.0×10^{-7}).

```
*OPERATING CONDITIONS: GENERAL
   Temperature:              20.000000    Pressure:                  1.000000

*REFRACTIVE INDICES
 SRF     GLASS            RN1        TCE
  0      AIR           1.000000      --
  1      LASF35        2.014931   74.000000
  2      AIR           1.000000  236.000000
  3      LASF35        2.014931   74.000000
  4      AIR           1.000000  236.000000
  5      IMAGE SURFACE
```

Perform a paraxial analysis and note the Effective focal length and Image numerical aperture. Trace a spot diagram from the on-axis field point and note the Strehl ratio, which characterizes the performance of the system.

```
*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:     15.000000    Image axial ray slope:    -0.250000
   Object num. aperture:    1.5000e-19    F-number:                  2.000002
   Image num. aperture:       0.250000    Working F-number:          2.000002
.........
 OTHER DATA
   Entrance pupil radius:    15.000000    Srf 1 to entrance pup.:       --
   Exit pupil radius:        12.089964    Srf 4 to exit pupil:     -15.956128
   Lagrange invariant:     -1.5000e-05    Petzval radius:         -205.107372
   Effective focal length:   60.000049

*TRACE REFERENCE RAY
         FBY          FBX          FBZ
          --           --           --
        FYRF         FXRF          FY           FX
         --           --           --           --
         YC           XC          YFS          XFS          OPL    REF SPH RAD
         --           --     0.005766     0.005766    66.447315    48.354128


*SPOT DIAGRAM: MONOCHROMATIC
 APDIV    17.030000
 WAVELENGTH 1
 WAV WEIGHTS:
       WW1
    1.000000
 NUMBER OF RAYS TRACED:
       WV1
       232
 PER CENT WEIGHTED RAY TRANSMISSION:   100.000000

*SPOT SIZES
   GEO RMS Y   GEO RMS X   GEO RMS R  DIFFR LIMIT     CENTY        CENTX
    0.000490    0.000490    0.000692    0.001593      --           --

*WAVEFRONT RS
 WAVELENGTH 1
   PKVAL OPD     RMS OPD  STREHL RATIO    RSY          RSX          RSZ
    0.033080    0.011037    0.995412      --           --           --
```

Now apply change the temperature to 40 degrees and apply thermal expansion by issuing the command tem(40, yes) or by setting the Temperature to 40 in the General Operating Conditions spreadsheet. Print the lens data and compare the surface data and refractive indices to the original values. Perform a paraxial analysis and compare the new focal length and numerical aperture to the original values. Trace a spot diagram from the on-axis field point and compare the Strehl ratio to the original value.

```
*OPERATING CONDITIONS: GENERAL
   Temperature:              40.000000    Pressure:                  1.000000

*LENS DATA
He-ne f/2 doublet focusing lens
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    1.0000e+14            AIR

  1     41.046074      5.000740     15.002220 A        LASF35 C
  2   -542.755316     13.906183     15.002220            AIR

  3    -40.701023      5.000740      9.001332          LASF35 P
  4   -124.330000     32.413446      9.000000            AIR
```

```
  5        --            --          0.002000

*REFRACTIVE INDICES
 SRF     GLASS            RN1         TCE
  1      LASF35        2.015019   74.000000
  2      AIR           1.000000  236.000000
  3      LASF35        2.015019   74.000000
……

*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:     15.000000    Image axial ray slope:    -0.250029
   Object num. aperture:    1.5000e-19    F-number:                  1.999768
   Image num. aperture:       0.250029    Working F-number:          1.999768
……
 OTHER DATA
   Entrance pupil radius:    15.000000    Srf 1 to entrance pup.:       --
   Exit pupil radius:        12.089353    Srf 4 to exit pupil:      -15.961051
   Lagrange invariant:     -1.5000e-05    Petzval radius:          -205.103693
   Effective focal length:   59.993027

*TRACE REFERENCE RAY
         FBY          FBX          FBZ
         --           --           --
        FYRF         FXRF          FY           FX
         --           --           --           --
         YC           XC           YFS          XFS          OPL    REF SPH RAD
         --           --        -0.022704    -0.022704    66.472804   48.374497

*SPOT DIAGRAM: MONOCHROMATIC
 APDIV    17.030000
 WAVELENGTH 1
 WAV WEIGHTS:
       WW1
    1.000000
 NUMBER OF RAYS TRACED:
       WV1
       232
 PER CENT WEIGHTED RAY TRANSMISSION:   100.000000

*SPOT SIZES
   GEO RMS Y   GEO RMS X   GEO RMS R  DIFFR LIMIT     CENTY       CENTX
    0.004227    0.004227    0.005978    0.001593      --          --

*WAVEFRONT RS
 WAVELENGTH 1
   PKVAL OPD     RMS OPD  STREHL RATIO    RSY          RSX          RSZ
    1.460707    0.434566    0.040493      --           --           --
```

In most cases, the effects of temperature change are limited to first-order; that is, changes in focal position and magnification. If the lens has a focusing mechanism, it can be used to counteract the temperature change (the focal shift is said to be a *compensator* for the temperature change). This can be seen here by using Autofocus and then re-tracing the on-axis spot diagram; the Strehl ratio should indicate that the system is once again well-corrected.

```
*AUTOFOCUS
Optimal focus shift =    -0.032475

*TRACE REFERENCE RAY
         FBY          FBX          FBZ
         --           --           --
        FYRF         FXRF          FY           FX
         --           --           --           --
         YC           XC           YFS          XFS          OPL    REF SPH RAD
         --           --         0.009771    0.009771    66.440329   48.342022

*SPOT DIAGRAM: MONOCHROMATIC
 APDIV    17.030000
 WAVELENGTH 1
 WAV WEIGHTS:
       WW1
    1.000000
 NUMBER OF RAYS TRACED:
       WV1
       232
 PER CENT WEIGHTED RAY TRANSMISSION:   100.000000

*SPOT SIZES
   GEO RMS Y   GEO RMS X   GEO RMS R  DIFFR LIMIT     CENTY       CENTX
    0.000594    0.000594    0.000841    0.001592      --          --

*WAVEFRONT RS
```

```
WAVELENGTH 1
   PKVAL OPD     RMS OPD  STREHL RATIO    RSY         RSX         RSZ
    0.086983    0.019991    0.984163      --          --          --
```

In cases where no focusing mechanism is available, athermalization is considerably more difficult. It is necessary to choose materials (glass as well as mounts and spacers) carefully so that the effects of temperature change on one part of the lens are canceled by the effects on other parts of the lens.

Zoom telescope

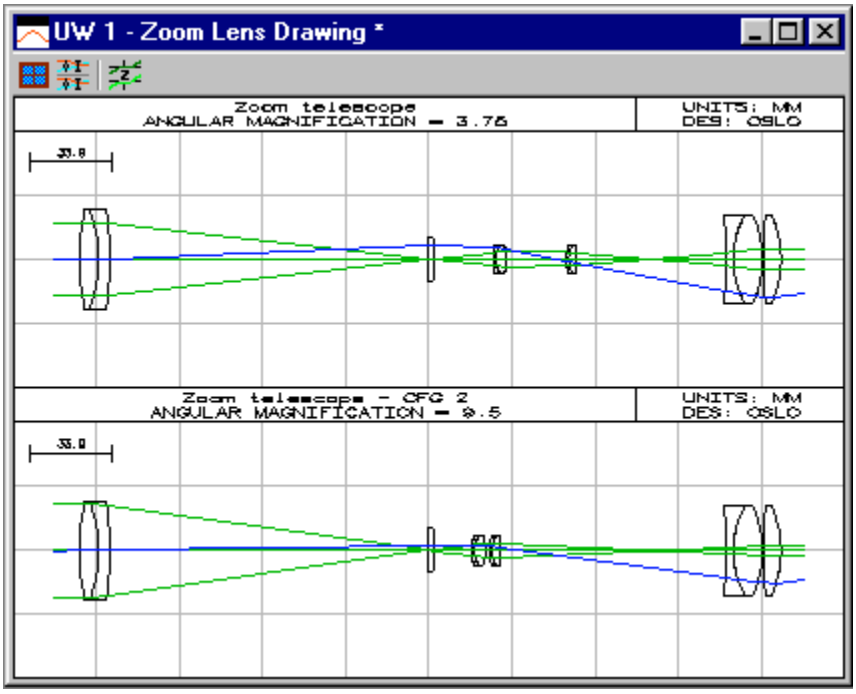
A rifle scope is an inverting telescope designed to be mounted on a rifle and used as a telescopic sight. The scope consists of four parts: an objective, an erecting system, a reticle, and an eyepiece. In use, the objective and erecting system for an image of an object at or near infinity on the reticle (or vice versa). The erector system in a real system contains tilt and decentering adjustments that provide alignment capability as well as compensation for windage and bullet drop, but the design included here does not include such adjustments. The overall system is afocal, and must be designed with generous eye relief to prevent injury when the rifle is fired.

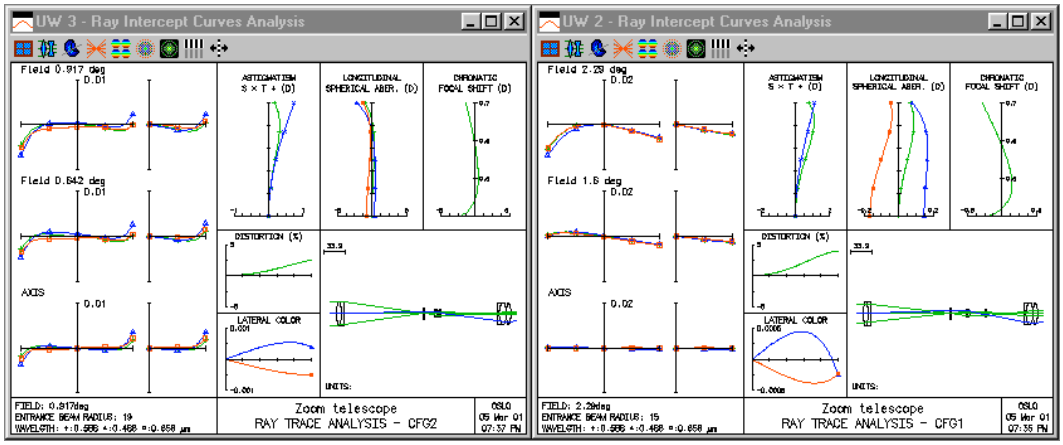
In fact, this system is one position of a zoom system. Surface 11, the last surface of the second erector doublet, is specified by a the command **pk lnm 4 11 90.7025** command, sometimes called a *zoom pickup*, because it holds the total distance between surface 4 and 12 at 90.7025mm, no matter what value is given to any intervening thicknesses. The system is zoomed by changing thicknesses 5 and 8. To design the system, you can choose some value of th[8], then optimize the image quality by varying th[5]. This will produce the proper location (or locations - there may be two) of components and the magnification. By repeating this procedure for several values of th[8], you can construct a *cam curve* that shows how the elements must track to change the magnification.

The lenses themselves can be designed with the system set at some particular magnification, or possibly at two different magnifications, to see how the optimum changes vs. magnification. Finally, you select one or the other (or some compromise) and make a final cam curve to complete the design. It is not necessary to use actual zoom optimization for a simple system such as this.

Ray displacements in afocal mode are actually direction tangents, not angles in radians, but are ordinarily so small that there is negligible difference.

The ray analyses shown on the next page show the performance of the scope at its normal magnification (3.75X), and also at a higher power (9.5X), which is achieved by changing th[5] to 15.208348, th[8] to 1.596579, the entrance beam radius to 18, and the field angle to .9 degrees. Note that the system has the afocal general operating condition set, so the ray displacements automatically are shown in radians.





OSLO has a number of routines to simplify working with zoom systems. The aberrations toolbar in the text output window contains several buttons dedicated to zoom systems, permitting analysis of a system in several positions with single commands, for example:

```
*GROUP THICKNESSES AND AIR SPACES FOR ZOOMING SYSTEMS

Group 1 consists of surf  2 to  6   Thickness =  144.810000
Group 2 consists of surf  7 to  9   Thickness =    5.480000
Group 3 consists of surf 10 to 12   Thickness =    4.250000
Group 4 consists of surf 13 to 17   Thickness =   22.610000

CFG  OBJ<->GRP1  GRP1<->GRP2  GRP2<->GRP3  GRP3<->GRP4   GRP4<->IMS
1     1.000e+20      23.7700      24.2900      60.6288     120.5415
2     1.000e+20      15.2083       1.5966      91.8839     120.5415

*ZOOM LENS DATA
MAGNIFICATION      CFG1       CFG2
        GRP1 -1.381e-18 -1.381e-18
        GRP2   -13.1573     4.0236
        GRP3     0.0936    -0.7755
        GRP4 -2.375e+05 -3.235e+04

          POWER        EFL        FNP        SNP         FF         BF
GRP1     0.0072   138.0769   243.1908  -139.9642   105.1138    -1.8872
GRP2     0.0379    26.3813     2.7291    -0.9525   -23.6522    25.4288
GRP3     0.0278    35.9617     0.0191    -2.6531   -35.9427    33.3087
GRP4     0.0220    45.3525    14.6663     1.2961   -30.6862    46.6486

                  IMAGE  EFFECTIVE   INFINITY      IMAGE      FIELD
CFG      EFL   DISTANCE        f/#        f/#      ANGLE      ANGLE        MAG
1  4.039e+07   120.5415  1.346e+06  1.346e+06     2.4137     2.2906     3.7496
2  1.394e+07   120.5415  3.668e+05  3.668e+05     5.0959     0.9167     9.5000

*VARIATION OF THE 3rd ORDER SEIDEL COEFFICIENTS BY ZOOMING
                  SA3        CMA3        AST3        PTZ3        DIS3
CFG1
    GRP 1   -0.000494    0.000194   -0.000269   -0.000597   -0.000454
    GRP 2   -0.000272    0.000406   -0.002056   -0.001250    0.009511
    GRP 3   -0.000210    0.000187   -0.001053   -0.000863   -0.000986
    GRP 4   -0.000013   -0.000234    0.001114   -0.000682    0.001474
    SUM     -0.000989    0.000553   -0.002263   -0.003391    0.009545
CFG2
    GRP 1   -0.002543    0.000315   -0.000138   -0.000306   -0.000074
    GRP 2   -0.000501   -0.000363    0.000086   -0.000642    0.001106
    GRP 3   -0.000645   -0.000003   -0.000359   -0.000443    0.001075
    GRP 4   -0.000002   -0.000061    0.000412   -0.000350    0.003030
    SUM     -0.003691   -0.000113    0.000002   -0.001741    0.005137
```

Wide-angle triplet - ASA

This section shows how Adaptive Simulated Annealing can be combined with normal damped least squares in a design study. The task was to design a 35 mm efl triplet lens with a speed of *f*/3.5 covering a half-field of 31.5 degrees. The wavelength range was the visible spectrum. No specification was imposed for distortion or vignetting.

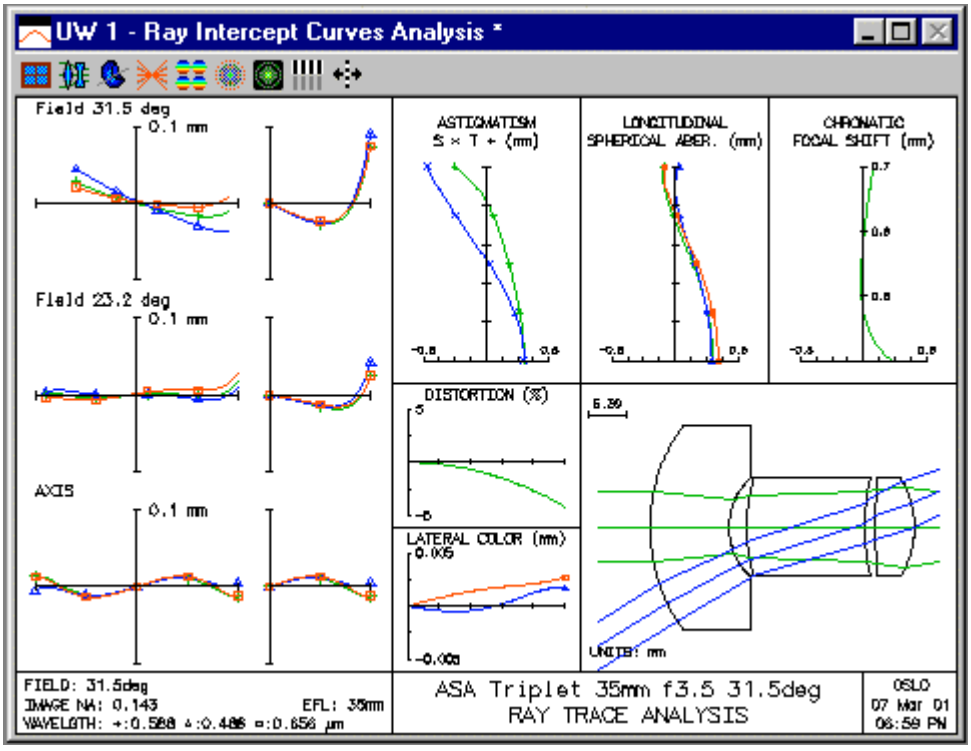
The starting design consisted of three zero-thickness pieces of model glass. (ASA does not require a starting design; if one is provided, it is discarded during the first pass.) The curvatures were allowed to vary between ±0.15, the glass thicknesses between 2 and 10, and the air spaces between 2 and 20. The refractive indices were allowed to vary between 1.5 and 1.9, and the dispersion factors between 0 and 1. The annealing rate was set to 0.05, as was the termination level. The error function included 50 terms based on Lobatto quadrature with 3 field points and 9 rays. Chromatic aberration was controlled using *D–d*, and the focal length was included in the error function rather than constrained as a solve. The starting system is shown below.

```
*LENS DATA
ASA Triplet 35mm f3.5 31.5deg
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    6.1713e+19            AIR

  1        --            --     V    5.000000 AS          AIR

  2        --     V      --     V    5.000000 S        GLASS1 V
  3        --     V      --     V    5.000000 S           AIR

  4        --     V      --     V    5.000000 S        GLASS2 V
  5        --     V      --     V    5.000000 S           AIR

  6        --     V      --     V    5.000000 S        GLASS3 V
  7        --     V      --     V    5.000000 S           AIR

  8        --            --          5.000000 S


*OPERANDS
 OP    DEFINITION                    MODE     WGT     NAME          VALUE   %CNTRB
O 1    "AVE"                           M      --     _YAVE1        3.310721   0.00
O 7    "RMS"                           M      --     YRMS1         2.500011   0.00
O 8    "AVE"                           M      --     _XAVE2        1.125000   0.00
O 13   "RMS"                           M      --     XRMS2         2.250000   0.00
O 14   "AVE"                           M      --     _YAVE2          --       0.00
O 19   "RMS"                           M      --     YRMS2         2.000000   0.00
O 20   "AVE"                           M      --     _XAVE3        0.875000   0.00
O 25   "RMS"                           M      --     XRMS3         1.750000   0.00
O 26   "AVE"                           M      --     _YAVE3          --       0.00
O 31   "RMS"                           M      --     YRMS3         1.500000   0.00
O 32   "AVE"                           M      --     _CHRAVE1        --       0.00
O 38   "RMS"                           M      --     CHRRMS1         --       0.00
O 39   "AVE"                           M      --     _CHRAVE2        --       0.00
O 44   "RMS"                           M      --     CHRRMS2         --       0.00
O 45   "AVE"                           M      --     _CHRAVE3        --       0.00
O 50   "RMS"                           M      --     CHRRMS3         --       0.00
O 51   "PU+0.142857"                   M    1.000000                0.142857   0.01
O 52   "PYC(1,5)"                      M    0.010000                  --       0.00
** BOUND VIOL: V1 V2 V3 V4 V5 V6 V7
MIN ERROR:     6.823322
```

ASA was set up to generate 20 solutions, which took about 3 hours on a Sparc2 workstation. The lenses produced during the ASA portion of the run were then modified to remove the glass variables, and the ordinary DLS optimization was used to drive each ASA solution to a local minimum, varying only the curvatures and thicknesses. The results are summarized in the following table and in the figures that follow (The final error function is saved in System Note #1 by ASA).

ASA solutions
solution #1: merit .0276
solution #2: merit .0275
solution #3: merit .0422
solution #4: merit .0488
solution #5: merit .0044
solution #6: merit .0213
solution #7: merit .0284
solution #8: merit .0337
solution #9: merit .0312
solution #10: merit .0274
solution #11: merit .0409
solution #12: merit .0288
solution #13: merit .0253
solution #14: merit .0232
solution #15: merit .0324
solution #16: merit .0364
solution #17: merit .0251
solution #18: merit .0281
solution #19: merit .0372
solution #20: merit .0225
average .0296

Solution #5 has the best error function of all the solutions found by ASA, so it was used as a starting point for additional (local) optimization. The local optimization involved constraining the stop to the front surface of the center element, adding more rays to the error function, changing the PU requirement on the back surface from a minimize to a constraint mode operand, and experimenting with glass combinations. Next, the design was turned around. This led to the final design:




```
*LENS DATA
```

```
Final design
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    6.0086e+19             AIR

  1     25.682267 V   11.257104 V   14.000000             SF59 C
  2      9.668970 V    2.500000      7.000000             AIR

  3     25.150557 V   16.844313 V    7.000000             SF59 C
  4     31.457684 V    1.548261 V    7.000000 A           AIR

  5    110.580228 V    5.567810 V    7.000000             LAK8 C
  6    -13.566281 V   37.963035 V    7.000000             AIR

  7        --            --         20.919998 S
```

The final design has somewhat marginal performance at this focal length. Like most designs of this type, it is limited by oblique spherical aberration that cannot be corrected. On the other hand, the lens does have good field coverage and ample back focus, and would be quite satisfactory at a reduced focal length.

The design study shows how ASA can be combined with normal lens design to produce a new solution. In this mode of application, no attempt is made for global optimization. ASA is used instead as a *multiple solution generator*. ASA has advantages over competing methods in this mode of operation, because it uniformly searches the entire solution space.

Wide-angle triplet - GE

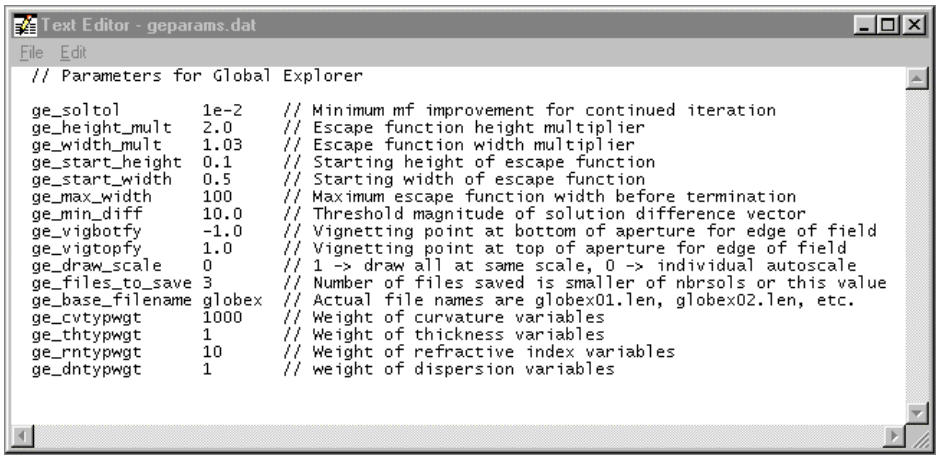
Global Explorer is a CCL command based on the Escape Function method developed by Isshiki, and is included with OSLO Premium. This example presents a demo procedure to illustrate the steps in using Global Explorer. You should first run the procedure as described below, to understand its use. Then you can experiment with different lenses and different values for the parameters.

Before running Global Explorer, you must have a lens in your computer that is ready for optimization, that is, it must have variables and operands that define a suitable error function. The escape function that is added to the error function is generated automatically by Global Explorer, so your error function should not include it. This demo uses a starting design for a wide-angle triplet having the same paraxial specifications as the design example used for ASA.

1. Open two graphic windows in addition to the text window, and arrange your display so the windows are similarly sized and located to the figures below.

2. Click File >> Open on the main OSLO menu. Click Public in the dialog box, and open the file demo/six/gedemo.len.

3. First choose the Global Explorer's Set Parameters option by executing it from the Optimization >> Global Explorer >> Set Parameters menu, or by typing the command ge (If you type the command, it will display an options box). This will display the parameters used by Global Explorer in the OSLO Editor. Don't change these parameters for the demo, just exit from the editor by clicking File >> Exit in the text editor (not on the main menu). Later, you can repeat this step to edit different parameters (i.e. change their values) and save the file before exiting.



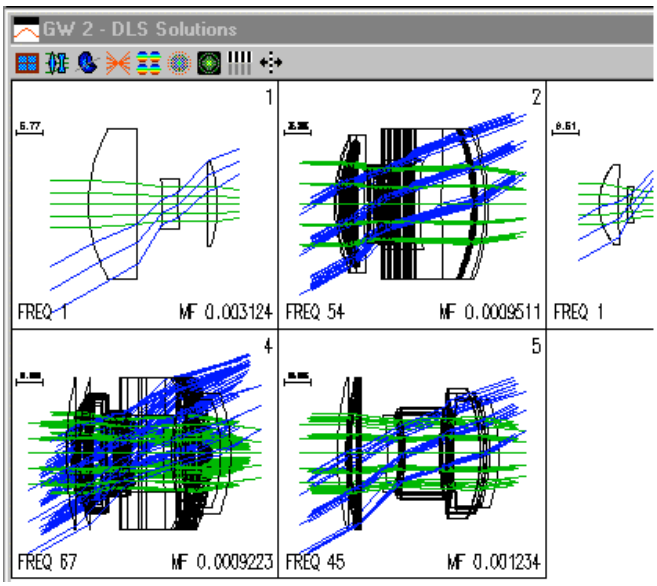
Most of the parameters for Global Explorer are described in chapter 7. There are a few additional ones that have been provided in the CCL program. For example, the ge_vigbotfy and ge_vigtopfy parameters allow you to specify the vignetting that sets the apertures of the solutions. In the demo, the range of fractional aperture vignetting goes from –1.0 to 1.0. Note that this is not the same as the vignetting defined in the optimization field point set, which has a range from –0.8 to 0.8. Often it is advantageous to have the optimization vignetting different from the usage vignetting.

Another of the parameters is called ge_draw_scale. If this is set to 1 all the solutions will be drawn to the same scale. If ge_draw_scale is set to 0, each solution will be scaled automatically to fit its viewport.

The parameter ge_soltol sets the value of the corresponding OSLO operating condition opst (optimization solution tolerance). If you have a small display, you may want to use the zoom capability of OSLO graphics windows (provided by the toolbar buttons) to enlarge a particular viewport of interest so you can see it better.

Global Explorer will save as many lens files as are indicated by the ge_save_files parameter. Each file will contain the name set by ge_base_filename. For the demo, there will be three files, globex01.len, globex02.len, and globex03.len

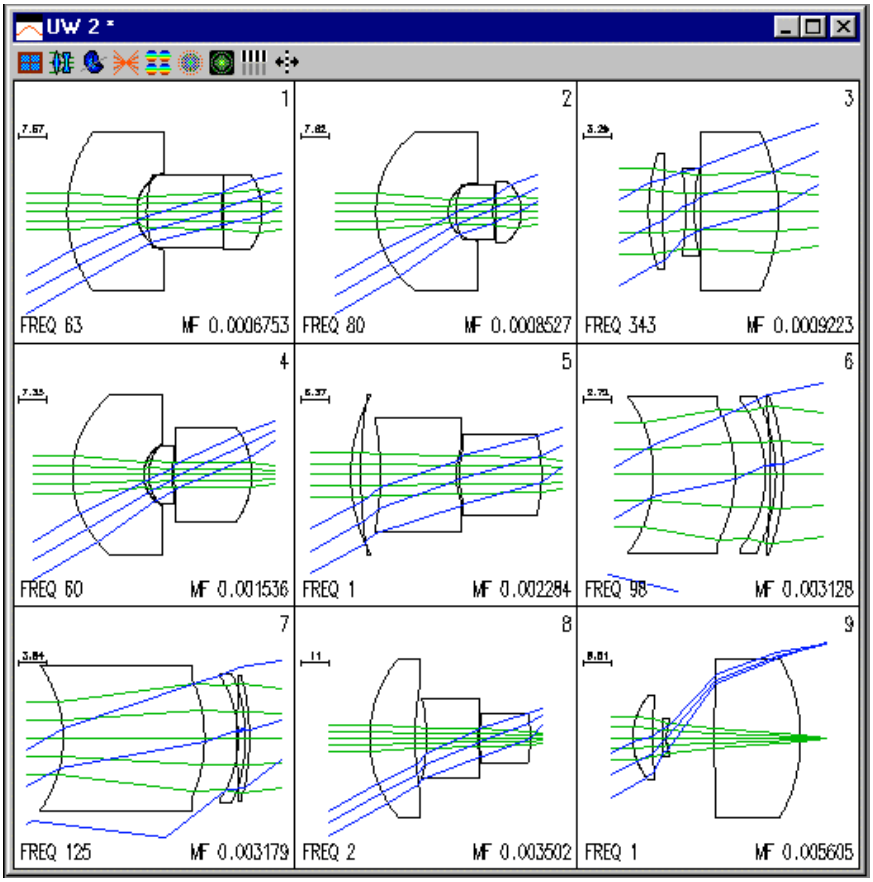
You can open these and evaluate or modify them as desired. Note that the solutions not saved as files will be not be available for evaluation or modification. It is quite usual for most of the Global Explorer solutions to be of only casual interest, and discarding them prevents excessive disk clutter.



4. Run Global Explorer again, and this time choose the Global Explorer option. The program will display an options box asking for the desired number of solutions. (Note that the values are the squares of integers, which is required by the window formatting that Global Explorer uses.) Double click "16" for this demo. The program will immediately begin to search for solutions. As it progresses, it will display the current starting point in graphics window GW1, and the solutions in graphics window GW2. There will be several solutions that are similar to previous solutions, and the appropriate graphics viewport will be overwritten if the new solution is better (i.e., lower error function) than the old. In each viewport, you will see the solution number in the upper right, the FREQ (i.e., number of times that solution has been obtained) in the lower left, and the minimum error function MF in the lower right. After the search is nearly complete, the window will appear as follows. Note that the Text Window displays the values of the parameters used for the current search.

You will see that it is not possible to tell much about the details of the solutions displayed in each viewport. There are two reasons for this. The first is that the drawing is automatically scaled to fit in the viewport, so different solutions may be drawn at different scales. The second is that each solution drawn in a viewport is compared for similarity only to the solution that currently exists there, not to all previous solutions. This means that after several replacements, the solution may be quite different from the one originally shown in the viewport. However, because solutions are only replaced if the new one is better than the old, the method guarantees that the final one will be the best of a series.

After the search is completed, the solutions will be sorted so the minimum error function appears first, and combined so that all solutions that fall within the minimum difference (as defined by the ge_min_diff parameter described in the Beginner's Guide) are merged. The solutions window is then redrawn, with only the best one in each group being displayed. For this reason, the number of final solutions may be smaller than the original number of solutions (the FREQ of each final solution is adjusted to indicate the total merged number of overlaps). For the demo, the final display will appear as shown below.



After you have successfully run the demo, you can re-run the program on your own lenses. Global Explorer is a multiple solution generator whose purpose is to provide several solutions that may be interesting for additional design work. You will find that adjusting the parameters used for the program will have major effects on the ability of the algorithm to generate multiple solutions and the quality of the solutions. You should not hesitate to experiment with different values.

One of the interesting aspects of Global Explorer is that it displays chaotic behavior, in contrast to Adaptive Simulated Annealing (ASA), which displays stochastic behavior. That is, if you start Global Explorer with exactly the same input conditions, you will get the same final results in several trials, but if you change the conditions even a tiny amount between trials, you may get results that are grossly different. This is not the case with ASA, which uses random numbers, and hence automatically produces a different execution path for each run.

In setting up a starting point to use with Global Explorer, it is important to use a trusted and stable error function, since the program may run for extended lengths of time without user intervention. It is usually worthwhile to include edge and center thickness controls in the error function, of course in addition to control of paraxial properties.

While you are using Global Explorer, you may wish to abort the optimization if you see that the process is not proceeding according to your expectations. To do this, press the ESCAPE key on your keyboard. It may take a few moments to respond, but the program should terminate gracefully, saving the work previously completed.

For additional information about Global Explorer, please see the "Beginner's Guide to Global Explorer" posted on the web site www.sinopt.com under the Design Note on "Multiple Solution Generators". This document is also supplied as an HTML file in the OSLO help directory. Also, see the paper "Global optimization with escape function" by Masaki Isshiki in the 1998 International Optical Design Conference Proceedings, SPIE Vol. 3482, pp. 104–109 (1998).

Eikonal design

OSLO Premium contains several sample eikonal functions and lenses that use these functions. These functions and the material in this section were contributed by Adriaan Walther of Worcester Polytechnic Institute. The eikonals themselves are contained in the eikonal DLL; the source code may by found in \bin\dll\eikonal.c. The lenses are in \public\len\demo\premium\dll.

The eikonal "ang35" is the angle eikonal of a lens system between its front nodal plane and back nodal plane. The paraxial part of the eikonal is $f(a - b + c)$, in which f is the focal length, which is stored in eikonal coefficient 0. Coefficients 1 through 16 go with the third order and fifth order terms of the series development in the following order:

aa, ab, ac, bb, bc, cc;

aaa, aab, aac, abb, abc, acc, bbb, bbc, bcc, ccc.

Here "aac" means "a squared times c", etc. Writing the terms in this fashion shows more clearly the way in which the terms are ordered. If you look at the C-code you'll see a rather peculiar looking expression for this eikonal function, but not to worry: it is simply a Horner expansion of the polynomial, which saves many multiplications compared to straightforward computation.

"Angthrd" is exactly the same, but with the fifth order terms left out. In this case there are only seven coefficients: the focal length and the six third order coefficients.

The macro lens

This lens is stored in file "macro04.len". It is used at six different magnifications: 0, –0.1, –0.2, –0.3, –0.4, and –0.5. The aperture is f/2.8 for an object at infinity. The aperture stop is fixed; it is located in the front nodal plane. The image height is kept at 20 mm for all six configurations, close to the corner of a 24x36 negative.

The default OSLO error function was generated for the six magnifications jointly, and the lens was optimized by first varying the third order coefficients (6 variables), then varying the third and fifth order coefficients (16 variables), and finally adding the six image distances as variables (22 variables in total). This procedure was tried a few times with different starting points, but it was found that the aberration curves came out pretty much the same. File "macro04.len" shows the result of one of these optimization runs.

This case was done as an exercise. It is essentially the problem that was solved in J. Opt. Soc. Am. A 6, 415–422 (1989). At the time this paper was written, an entire computer program was written to solve this problem; with OSLO it can be solved painlessly in no more than a couple of hours. It is an interesting example of the use of eikonal ray tracing, because it shows the best performance that can be expected from a non-floating element macro lens, no matter how complicated we make it.

The two-group zoom lens

It is well known that a regular camera lens cannot be corrected perfectly for more than one magnification. The question may be asked whether there are similar theorems for zoom lenses. For instance: can a zoom lens, used with the object at infinity, be corrected perfectly for its entire range of zoom settings? It can be shown, algebraically, that a two-group zoom lens cannot be perfect in this sense. One case of numerical optimization illustrating this point is stored in file "0714c.len".

Here are some data on this lens: it is a reversed telephoto system with a front group with a focal length of –50 mm and a rear group with a focal length of +40 mm. For a system focal length f, the spacing (measured between nodal planes) is 2000/f – 10, and the back focal length is 0.8f + 40. In the file, the nominal system has a focal length of 30 mm, and configurations 2, 3, and 4 have system focal lengths of 45 mm, 60 mm, and 75 mm. The pupil is put in the front nodal plane of the second group. For each configuration its diameter is adjusted to yield an f/2.8 aperture. The field angle is adjusted to give a maximum image height of 20 mm.

The eikonal function for this lens is "angthrd", which uses the third order coefficients only. The error function is the OSLO standard merit function for the four configurations jointly. The

variables are the six third order eikonal coefficients for both the front and the rear part of the lens, and, in the last stages, also the image distances.

The algebra shows that the third order aberrations cannot all be corrected over a continuous range of focal lengths. When you look at the aberration graphs keep in mind that there are also fifth and higher order aberrations that have not been corrected at all; if the fifth order coefficients were included in the optimization the result would be a lens better than shown in this file.

As it is, the results are not so very bad. If the aperture were reduced to f/4 a very good two-group zoom lens would be possible. Designers interested in this type of system might want to do more work on it to find the ultimate limits of performance for this type of system.

The Donders afocal system

This is a "+–+" afocal system that is symmetric at unit magnification. (One could also consider "–+–" systems.) With computer algebra it can be shown that, with both the object and the image at infinity, all the third and fifth order aberrations for the entire range of possible magnifications can be corrected by giving the eikonal coefficients suitable values. Note that this holds for an extended field of view, not merely on axis. All seventh order aberrations can be corrected as well; the conjecture that all the aberrations to all orders can be corrected has not, as of yet, been proven.

If this telescope is put in front of a fixed focal length perfect camera lens, we end up with a zoom lens without any third, fifth, (and seventh) order aberrations at any zoom setting. The distance between the third group of the telescope and the fixed focal length lens can be kept constant, so these two groups can be consolidated into one group. The conclusion is that, in principle, a three-group zoom lens used with the object at infinity can form a perfect image at all zoom settings.

This is merely an existence proof, not a recipe how to design zoom lenses. Nevertheless these results may have some practical use, because the theory tells us what the eikonal function should be for each of the three groups. These groups can then be designed one at a time. Of course, we all know from bitter experience that the resulting lenses will not match the desired eikonal functions perfectly. So at the end of the design a full scale optimization for the entire zoom lens is still necessary. Nevertheless it is quite possible that by using the theory we might arrive at new and better zoom lens designs.

Of course, proper imaging is only part of the problem. Weight and size play an important role, the groups should not bump into one another, etc. Then there is the problem of focussing. The guess is that it takes two more variable spacings (four variable spacings in all) to make a zoom lens that is perfect at all zoom settings as well as for all object distances.

The lens stored as "0703a.len" is a Donders telescope followed by a long focal length perfect lens so that we do not have to deal with images at infinity. It was determined algebraically what the eikonal coefficients should be. Note that the symmetry of the system restricts the values of the eikonal coefficients. If F(a,b,c) is the eikonal for the first group, then the eikonal of the third group must be F(c,b,a). So, for instance, the 8th coefficient for the first group (the aab term) must be equal to the 15th coefficient for the third group (the bcc term). The middle lens group must be symmetric by itself, so if its eikonal is G(a,b,c) we must have G(a,b,c) = G(c,b,a). So its 8th coefficient must be equal to its 15th coefficient, etc.

As the lens file is set up, the transverse magnification of the telescope is –0.50. For other magnifications, G say, the first and second spacings should be changed to 50 (3 – G) and 50 (3 – 1/G) respectively. As all the third and fifth order aberrations are zero the location of the entrance pupil is not really important; it is put in the front nodal plane of the first lens for convenience.

Try to ray trace this lens; you'll find that the aberrations are not zero! The reason is that the third and fifth order aberrations are corrected, but the seventh and higher order aberrations are not. To verify that the residual aberrations are indeed mostly seventh order, multiply both the aperture size and the field angle by the seventh root of 1/10, i.e. by 0.7197. This would reduce the seventh order aberrations by a factor of 0.7197 to the seventh power, which is 0.100. The third order aberrations would change by a factor 0.7197 to the third power, which is 0.372. The fifth order aberrations would change similarly by a factor 0.193. A ray trace with the reduced aperture and field shows

that the ray intercept curves look exactly the same as before, but with values 10 times as small. This demonstrates that all third and fifth order aberrations are really zero.

As an exercise, you can change the magnification by merely changing two thicknesses (see above) and then ray trace again with the full and the reduced aperture and field. You'll find that the third and fifth order aberrations are always corrected, no matter what value you choose for the magnification.

Other eikonals

"Axperf" is the angle eikonal for a lens corrected for spherical aberration at all conjugates. Note that the last term of formula (31.49) in the book is wrong; it should be (Pa + Qb + Rc)(4ac – b^2), in which P, Q, and R are three arbitrary constants. This error is corrected in "axperf".

"Concent" is the angle eikonal for an arbitrary concentric lens. The high degree of symmetry of a concentric system makes that each aberration order is completely specified by a single coefficient. Terms out to the eleventh order have been included in the eikonal. (See book, Eq. (29.7)).

"Concapl" is the angle eikonal for a concentric system that is perfectly aplanatic for a specified magnification. The inputs are the focal length and the magnification. See book, Eq. (29.24). Remember that with a flat rather than concentric object plane there will be fifth and higher order off-axis aberrations, because the distance to the center of the system varies with the field angle when the object plane is flat.

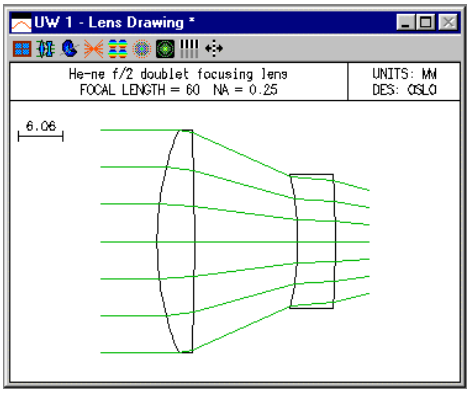
Tolerancing

User-defined tolerancing

User-defined tolerancing is a term used in OSLO to describe the process of settting tolerances when optical performance is measured by a user-defined error function. This may be the same function used to design the system, or a different function intended to represent as-used performance. In user-defined tolerancing, compensators are adjusted using the design optimization routines. This is slower than MTF/wavefront tolerancing by a wide margin, but it provides great flexibility. There are two levels of user-defined tolerancing in OSLO, depending on whether the tolerance operands are implemented as CCL or built-in operands. Of course, built-in operands provide increased speed.

Using CCL tolerance operands

As an example of the use of CCL operands, we consider tolerancing the air-spaced doublet (lasrdblt.len) supplied in the demo/lt directory. The air space in this design can be expected to be critical, since it controls the ray height on the overcorrecting surface. To get an idea of the tolerance on thickness 2, you can use the User-defined tolerance routine in OSLO.



To run a tolerance analysis on a lens, use a copy of the lens, not the original file. OSLO uses the same data structure for optimization and tolerancing, so tolerancing data will overwrite optimization data. After you have opened a saeparate copy of the lens, you should follow the steps below.

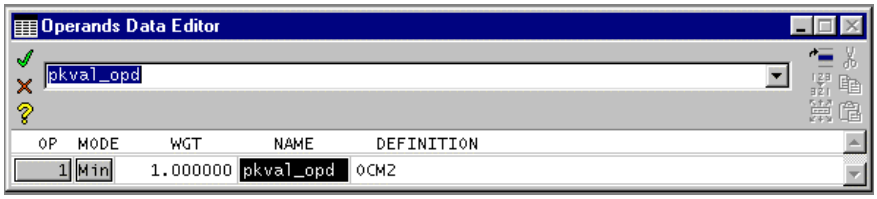
1) Remove any variables, then make the thickness of the image surface a variable, which will be used as a compensator during tolerancing.

2) Click Tolerance>>Update Tolerance Data>>Surface to open the tolerance data spreadsheet. The only tolerance of present interest is the second thickness. The ISO 10110 default value for this thickness is 0.2 mm, which is much too loose for the present system. As a general rule, it is best to start with tolerances that are too small and then increase them as you become familiar with their effect on performance. Change the tolerance to 0.05 mm, and close the spreadsheet.

3) Enter the star command *opsettol. The star command will set up the proper callback function, and the operands spreadsheet will open. The various options for tolerance operands, labeled Ocm*x*, will be shown in the text output window, to help you in choosing the desired one for your application.

```
*opsettol (operand numbers in parentheses)
GEO_RMS_R(1) PKVAL_OPD(2) RMS_OPD(3)   STREHL(4)
   0.000692    0.033080    0.011037    0.995416
```

4) In the spreadsheet, enter **ocm2** for the tolerance operand, and enter the name **pkval_opd** to provide a mnemonic description of the operand.



5) Now, to compute the effect of a .05 mm tolerance on the peak-to-valley opd, click Tolerance>>User-Defined Tolerancing. Choose *Sensitivity* from the dialog box, and select Air space from the options list, then click OK.

What happens is that OSLO computes the present value of the tolerance error function with the nominal system, then changes the tolerance of the second thickness by 0.05, and re-optimizes the system to restore the original error function value (n.b. it does not minimize the function). Both positive and negative perturbations are evaluated. After a short time, the text window should contain an analysis similar to the following.

```
*TOLERANCE SENSITIVITY ANALYSIS
 ERROR FUNCTION FOR NOMINAL SYSTEM:    0.033080

 AIR SPACE TOLERANCE
                      ERROR FUNCTION CHANGE           COMPENSATED CHANGE
 SRF    TOLERANCE    PLUS PERT    MINUS PERT       PLUS PERT    MINUS PERT
  2      0.05         6.953074      6.973735        0.259757      0.294596

 STATISTICAL SUMMARY
                        UNCOMPENSATED    COMPENSATED
 WORST CASE CHANGE         6.973735        0.294596
 STANDARD DEVIATION
     RSS                   6.973735        0.294596
     UNIFORM               4.026288        0.170085
     GAUSSIAN              3.067138        0.129567

 COMPENSATOR STATISTICS
 COMP           MEAN         STD DEV        MAX          RSS
 TH     5     0.000122     0.149552     0.149674     0.149674
```

The analysis shows the effects of a .05 air space tolerance on the error function, which represents the peak-to-valley opd. The error function changes are shown for both positive and negative perturbations, and for both compensated (i.e. adjustment of the image distance) and uncompensated (fixed image distance) conditions. Then comes a statistical analysis of the probable effects of this tolerance specification on a large number of systems, assuming various probability distributions.

The above analysis shows the change for a given tolerance. You may instead be interested in the tolerance that can be allowed to produce a given change, say 0.15, which would bring the system to (approximately) the diffraction limit. Click Tolerance>>User-Defined Tolerancing again, but this time select *Inverse Sensitivity*. Select Air Space from the list, then enter 0.15 as the allowed change in the error function. The text window will then show that the allowed tolerance is about 7 microns if the focus is not adjusted, or 35 microns if the focus is used to compensate for a spacing error.

```
*INVERSE SENSITIVITY ANALYSIS
 ERROR FUNCTION FOR NOMINAL SYSTEM:    0.033080
 ALLOWED CHANGE IN ERROR FUNCTION:     0.150000

 AIR SPACE TOLERANCE
                 ALLOWED TOLERANCE
 SRF      UNCOMPENSATED    COMPENSATED
  2          0.007301        0.035245
```

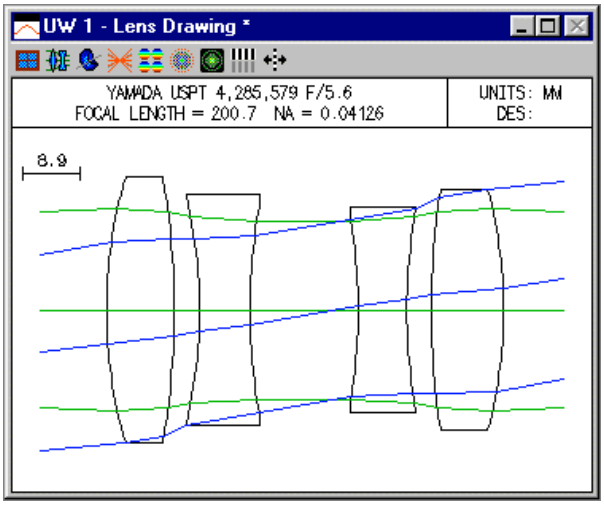
The above analysis, while simple, shows the essential steps in tolerancing.

Using built-in operands

As an example of using built-in operands with the user-defined tolerancing routines, we will compute power (spherical) error tolerances for a four-element copy lens from the OSLO lens library. Open the lens "public\len\lib\toolbox\yama001.len." Before saving the lens to a temporary file, move the thickness of the next to last surface to the image surface and then delete the surface, so that your results will correspond to the ones below. In tolerancing, particularly in user-defined tolerancing, you should not expect exact duplication of results. This is because tolerancing involves optimization routines that may be set up slightly differently, or use different parameters, from the ones use to make the example output.

1) Note the lens is designed for a magnification of –1. As a first step in tolerancing (after saving the file under a new name), we will assign apertures to the elements so that there is no vignetting. Close the lens spreadsheet if it is open, then use Optimize>>Support Routines>>Vignetting>>Set Apertures, and accept the default values shown in the dialog box. The resulting lens is

```
*LENS DATA
YAMADA USPT 4,285,579 F/5.6
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --        406.280280     56.340000            AIR

 AST        --        -36.854492     17.000000 AK         AIR

  2      71.953519     10.404504     20.300000          LAKN13 C
  3    -112.429158      3.825293     20.300000            AIR

  4     -78.402563      7.660029     17.600000            LF5 C
  5      84.822117     16.593731     15.600000 K          AIR

  6     -80.540361      7.411215     13.900000            LF5 C
  7      82.495107      3.759799     15.600000            AIR

  8     121.677279     11.054088     18.400000          LAKN13 C
  9     -70.206359    373.312870     18.400000            AIR

 10        --          0.001155     57.000000 K          AIR

 IMS       --            --         56.852074 S

*PARAXIAL CONSTANTS
   Effective focal length:  200.690160    Lateral magnification:    -1.009090
   Numerical aperture:        0.041257    Gaussian image height:    56.852145
   Working F-number:         12.119077    Petzval radius:         -2.0564e+03
   Lagrange invariant:       -2.347600
```



2) For tolerance operands, we will use the field-averaged RMS spot size, computed in three wavelengths. Since this is a rotationally-symmetric lens and we are only perturbing the curvatures, we can use the OSLO error function generator with the default field and pupil sampling options. In the dialog box for the Optimize>>Generate Error Function>>OSLO Spot Size/Wavefront dialog box, accept all the defaults except for the

color correction method, which should be set to *Use All Wavelengths*. After generating the error function, you can compute the operands using the Ope button in the text output window, which should produce the following.

```
*OPERANDS
OP   MODE    WGT      NAME          VALUE    %CNTRB DEFINITION
O 20   M    0.750000 Yrms1         0.015309  12.71 RMS
O 61   M    1.500000 Xrms2         0.012533  17.03 RMS
O 102  M    1.500000 Yrms2         0.017178  31.99 RMS
O 143  M    0.375000 Xrms3         0.012813   4.45 RMS
O 184  M    0.375000 Yrms3         0.035323  33.82 RMS
MIN RMS ERROR:     0.017534
```

3) We will assume that we are going to allow a focus adjustment in the final lens assembly, so next you should designate that the back focus (the thickness of surface 10) is a variable to be used as a compensator.

```
*VARIABLES
 VB   SN  CF  TYP       MIN         MAX        DAMPING      INCR        VALUE
V 1   10   -  TH        --          --        1.000000    0.001693     0.001155
```

4) Now use Lens>>Show Tolerance data>>Surface to display the default (ISO110) tolerances for the lens in the text output window:

```
*SURFACE TOLERANCES
YAMADA USPT 4,285,579 F/5.6
      RADIUS  RD TOL    FRINGES THICKNESS TH TOL            RN TOL  DECEN  TILT
 SRF CON CNST CC TOL   PWR   IRR  TLC TOL DZ TOL      GLASS  V TOL   Y/X    A/B
  1     --      --     --    --  -36.8545 0.4000        AIR   --     --     --
        --      --                   --      --              --     --     --
 ------------------------------------------------------------------------------
  2   71.9535   --   10.00* 2.00* 10.4045 0.4000     LAKN13 0.0010   --   0.1667
        --      --                   --      --             0.8000   --   0.1667

  3 -112.4292   --   10.00* 2.00*  3.8253 0.4000        AIR   --     --   0.1667
        --      --                   --      --              --     --   0.1667
 ------------------------------------------------------------------------------
  4  -78.4026   --   10.00* 2.00*  7.6600 0.4000        LF5 0.0010   --   0.1667
        --      --                   --      --             0.8000   --   0.1667

  5   84.8221   --   10.00* 2.00* 16.5937 0.4000        AIR   --     --   0.1667
        --      --                   --      --              --     --   0.1667
 ------------------------------------------------------------------------------
  6  -80.5404   --   10.00  2.00   7.4112 0.2000        LF5 0.0010   --   0.3333
        --      --                   --      --             0.8000   --   0.3333

  7   82.4951   --   10.00* 2.00*  3.7598 0.4000        AIR   --     --   0.1667
        --      --                   --      --              --     --   0.1667
 ------------------------------------------------------------------------------
  8  121.6773   --   10.00* 2.00* 11.0541 0.4000     LAKN13 0.0010   --   0.1667
        --      --                   --      --             0.8000   --   0.1667

  9  -70.2064   --   10.00* 2.00*373.3129 0.4000        AIR   --     --   0.1667
        --      --                   --      --              --     --   0.1667
 ------------------------------------------------------------------------------
 10     --      --     --    --    0.0012   --                      --     --
        --      --                   --                            --     --
 FRINGE WAVELENGTH:    0.546070
   Fringes measured over clear aperture of surface unless indicated.
 * Fringes measured over 30 mm diameter test area, per ISO 10110.
   Tilt tolerances are specified in degrees.
```

Examination of the above listing reveals that the spherical error tolerance is 10 fringes, measured over a 30 mm diameter test area for each surface. For this example, we will use the default of 10 fringes, but specify that the error is to be measured over the entire area of each surface.

5) To do this, we open the tolerance data spreadsheet and enter “10” for the spherical form error for surfaces 2 through 9. You can enter the values one by one in the spreadsheet, but it is easier to close the spreadsheet and use the Tolerance>>Update Tolerance Data>>Set Tolerance Value command, which can change all surfaces at once. After making the change, reopen the spreadsheet, which should appear as follows.

SRF	RADIUS	RD TOL	PWR FR	IRR FR	THICK	TH TOL	GLASS	RN TOL	DY TOL	TA TOL
1	0.0	0.0	0.0	0.0	-36.854	0.4000	AIR	0.0	0.0	0.0
2	71.954	0.0	10.00	2.00*	10.405	0.4000	LAKN13	0.0010	0.0	0.1667
3	-112.429	0.0	10.00	2.00*	3.825	0.4000	AIR	0.0	0.0	0.1667
4	-78.403	0.0	10.00	2.00*	7.660	0.4000	LF5	0.0010	0.0	0.1667
5	84.822	0.0	10.00	2.00*	16.594	0.4000	AIR	0.0	0.0	0.1667
6	-80.540	0.0	10.00	2.00	7.411	0.2000	LF5	0.0010	0.0	0.3333
7	82.495	0.0	10.00	2.00*	3.760	0.4000	AIR	0.0	0.0	0.1667
8	121.677	0.0	10.00	2.00*	11.054	0.4000	LAKN13	0.0010	0.0	0.1667
9	-70.206	0.0	10.00	2.00*	373.313	0.4000	AIR	0.0	0.0	0.1667
10	0.0	0.0	0.0	0.0	0.001	0.0			0.0	0.0

FRINGE WAVELENGTH: 0.546070

Fringes measured over clear aperture of surface unless indicated.

* Fringes measured over 30 mm diameter test area, per ISO 10110.

Tilt tolerances are specified in degrees.

Display all surface tolerances: ○ Yes ◉ No

6) Note that the lack of the asterisk next to the spherical fringe tolerance means that the fringes are measured over the clear aperture of the surface, as we want. Now we perform a sensitivity analysis for the spherical form error by selecting Tolerance>>User-defined Tolerancing >> Surface, choosing *Sensitivity* and selecting Power fringes from the options list.

```
*TOLERANCE SENSITIVITY ANALYSIS
 ERROR FUNCTION FOR NOMINAL SYSTEM:    0.017534

 POWER ERROR TOLERANCE
                    ERROR FUNCTION CHANGE          COMPENSATED CHANGE
 SRF    TOLERANCE    PLUS PERT     MINUS PERT      PLUS PERT     MINUS PERT
  2     10.0          0.015823      0.014216        0.000250     -0.000224
  3     10.0          0.016268      0.013579        0.001278     -0.001205
  4     10.0          0.014128      0.017034       -0.001387      0.001514
  5     10.0          0.016408      0.017957       -0.000143      0.000174
  6     10.0          0.023312      0.025297       -0.000605      0.000647
  7     10.0          0.020602      0.022830       -0.000778      0.000967
  8     10.0          0.020502      0.018393        0.000791     -0.000686
  9     10.0          0.021476      0.019538        0.000582     -0.000544

 STATISTICAL SUMMARY
                        UNCOMPENSATED    COMPENSATED
 WORST CASE CHANGE         0.157189        0.006203
 STANDARD DEVIATION
     RSS                   0.056306        0.002517
     UNIFORM               0.032509        0.001453
     GAUSSIAN              0.024764        0.001107

 COMPENSATOR STATISTICS
 COMP           MEAN         STD DEV         MAX          RSS
 TH    10     0.005922     1.536235     1.847792     4.347659
```

As expected, the change in performance is much less when we allow for refocusing. Assuming that the errors have a uniform distribution, the standard deviation of the change in the average spot size is reduced from 33 μm to 1.5 μm if back focus adjustment is allowed. However, to achieve this performance level, we need to allow for a (2σ) range in focus of ±3 mm. This provides the data we need to build a focusing mechanism.

For this lens, we will assume that the maximum allowed spot size, average over the field and chromatic range, is 20 μm. Since this value for the nominal system is 17.5 μm, this means the maximum allowed change is 2.5 μm. If we desire a probable success rate of 99%, the table presented above indicates that we need a ratio of maximum allowed change to standard deviation of 2.5, i.e., a standard deviation of 1.0 μm. We want to redistribute the tolerances to target this standard deviation and also to balance the contributions of the surfaces to this target. Using $\sigma_{\delta S}$ = 0.001, κ = 0.58 (uniform distribution) and n = 8 in Eq. **(9.57)** yields a target contribution of ΔS_{tar} = 0.0006. We use this value as the requested change in an *Inverse Sensitivity* analysis.

```
*INVERSE SENSITIVITY ANALYSIS
 ERROR FUNCTION FOR NOMINAL SYSTEM:    0.017534
```

```
 ALLOWED CHANGE IN ERROR FUNCTION:     0.000600

 POWER ERROR TOLERANCE
                ALLOWED TOLERANCE
 SRF     UNCOMPENSATED    COMPENSATED
  2         1.749036       22.487217
  3         1.604631        4.765315
  4         1.550645        4.064342
  5         1.656797       29.400028
  6         1.380191        9.298149
  7         1.425375        6.426525
  8         1.506102        7.705936
  9         1.490417       10.304978
```

Without back focus adjustment, we see that the allowed spherical error tolerance is about 1.5 fringes for surface 3, 4, 6, 7, 8, and 9 and 2 fringes for surfaces 2 and 5. We now set the tolerances to these values, and re-run the *Sensitivity* analysis.

```
*TOLERANCE SENSITIVITY ANALYSIS
 ERROR FUNCTION FOR NOMINAL SYSTEM:    0.017534

 POWER ERROR TOLERANCE
                     ERROR FUNCTION CHANGE           COMPENSATED CHANGE
 SRF   TOLERANCE    PLUS PERT    MINUS PERT       PLUS PERT    MINUS PERT
  2     2.0          0.001122      0.000552      4.8010e-05   -4.6960e-05
  3     1.5          0.000834      0.000105        0.000187     0.000105
  4     1.5         9.6535e-05     0.000896      9.6535e-05     0.000220
  5     2.0          0.000705      0.001284     -3.1057e-05    3.2300e-05
  6     1.5          0.000563      0.001238     -9.3445e-05    9.4393e-05
  7     1.5          0.000415      0.001128       -0.000129     0.000134
  8     1.5          0.000988      0.000343        0.000112    -0.000110
  9     1.5          0.001019      0.000411      8.4944e-05   -8.4107e-05

 STATISTICAL SUMMARY
                       UNCOMPENSATED    COMPENSATED
 WORST CASE CHANGE        0.008507        0.000912
 STANDARD DEVIATION
     RSS                  0.003037        0.000365
     UNIFORM              0.001753        0.000211
     GAUSSIAN             0.001336        0.000161

 COMPENSATOR STATISTICS
 COMP           MEAN        STD DEV        MAX          RSS
 TH    10     0.025265     0.236382     0.290346     0.673286
```

The standard deviation for the uncompensated case is now about 1.5 μm. This value is larger than the expected value of 1 μm because of deviations from the linear dependence of the spot size on the surface perturbations.

Based on the earlier inverse sensitivity analysis for the compensated case, we will assign tolerances of 5 fringes to surfaces 3, 4, 7, and 8, 10 fringes to surfaces 6 and 9, and 20 fringes to surfaces 2 and 5, and repeat the sensitivity analysis.

```
*TOLERANCE SENSITIVITY ANALYSIS
 ERROR FUNCTION FOR NOMINAL SYSTEM:    0.017534

 POWER ERROR TOLERANCE
                     ERROR FUNCTION CHANGE           COMPENSATED CHANGE
 SRF   TOLERANCE    PLUS PERT    MINUS PERT       PLUS PERT    MINUS PERT
  2    20.0          0.040977      0.039167        0.000527     -0.000422
  3     5.0          0.005663      0.003696        0.000631     -0.000613
  4     5.0          0.003848      0.005989       -0.000711      0.000743
  5    20.0          0.044052      0.045786       -0.000255      0.000379
  6    10.0          0.023312      0.025297       -0.000605      0.000647
  7     5.0          0.006392      0.008149       -0.000414      0.000461
  8     5.0          0.007201      0.005568        0.000383     -0.000357
  9    10.0          0.021476      0.019538        0.000582     -0.000544

 STATISTICAL SUMMARY
                       UNCOMPENSATED    COMPENSATED
 WORST CASE CHANGE        0.160539        0.004352
 STANDARD DEVIATION
     RSS                  0.071154        0.001577
     UNIFORM              0.041081        0.000910
     GAUSSIAN             0.031294        0.000694

 COMPENSATOR STATISTICS
```

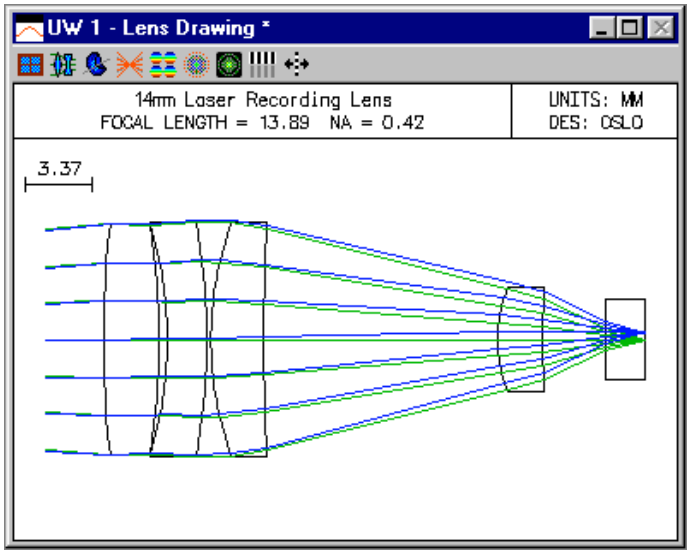
COMP		MEAN	STD DEV	MAX	RSS
TH	10	0.007251	1.723657	2.922255	4.879934

With these tolerances, the standard deviation of the performance changes is slightly less than our target of 1 μm. Also note that the individual contributions are more evenly distributed than in the previous analysis. The range of focus adjustment that is required has increased slightly to about ±3.5 mm.

Change table tolerancing

An example of the use of change tables is given in the paper by Smith(2). In this paper, Smith presents an analysis of the following laser recording lens, which is designed to work at a wavelength of 0.82 μm.

```
*LENS DATA
14mm Laser Recording Lens
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --         76.539000      1.949220            AIR

 AST     50.366000      2.800000      5.825000 A          SF11 C
  2     -39.045000      0.435265      5.810000            AIR

  3     -19.836000      2.000000      5.810000            SF11 C
  4     -34.360000      0.200000      5.950000            AIR

  5      17.420000      2.650000      5.905000            SF11 C
  6      79.150000     11.840000      5.610000            AIR

  7       7.080000      2.240000      2.620000            SF11 C
  8      15.665000      3.182000      2.065000            AIR

  9         --          2.032000      2.000000           ACRYL C
 10         --            --          2.000000            AIR

 IMS        --            --          0.350759 S

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.820000
        1.000000

*PARAXIAL CONSTANTS
   Effective focal length:   13.888494    Lateral magnification:    -0.179559
   Numerical aperture:        0.420000    Gaussian image height:     0.350000
   Working F-number:          1.190476    Petzval radius:          -15.782541
   Lagrange invariant:       -0.147420
```



The performance specification for this lens is that the Strehl ratio must be at least 0.75 over the entire field. Working from this requirement and the nominal design prescription, Smith shows that the tolerance budget must produce no more than 0.173 λ (peak-to-valley) of OPD.

Following the discussion in Section IV, we assign the following initial tolerances to the lens in order to compute the change tables. Note that the fringes are specified at a wavelength of 0.58929 μm (Sodium yellow). The sign of the spherical error fringes has been chosen to better match the change table in Table 2 of the paper. Using all positive values only changes the sign of the

2 W. J. Smith, “Fundamentals of establishing an optical tolerance budget,” Proc. SPIE Vol. 531, pp. 196-204 (1985).

corresponding change table entry and thus has no effect on the statistical sum (RSS) value. The tilt tolerance of 0.057 degrees is equal to 1 milliradian.

SRF	RADIUS	RD TOL	PWR FR	IRR FR	THICK	TH TOL	GLASS	RN TOL	DY TOL	TA TOL
1	50.366	0.0	-10.0	1.00	2.800	0.2000	SF11	0.0010	0.0	0.0570
2	-39.045	0.0	10.00	1.00	0.435	0.2000	AIR	0.0	0.0	0.0570
3	-19.836	0.0	10.00	1.00	2.000	0.2000	SF11	0.0010	0.0	0.0570
4	-34.360	0.0	10.00	1.00	0.200	0.2000	AIR	0.0	0.0	0.0570
5	17.420	0.0	-10.0	1.00	2.650	0.2000	SF11	0.0010	0.0	0.0570
6	79.150	0.0	-10.0	1.00	11.840	0.2000	AIR	0.0	0.0	0.0570
7	7.080	0.0	-10.0	1.00	2.240	0.2000	SF11	0.0010	0.0	0.0570
8	15.665	0.0	-10.0	1.00	3.182	0.0	AIR	0.0	0.0	0.0570
9	0.0	0.0	0.0	0.0	2.032	0.0	ACRYL	0.0010	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	AIR	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0			0.0	0.0

FRINGE WAVELENGTH: 0.589290

Fringes measured over clear aperture of surface unless indicated.

Tilt tolerances are specified in degrees.

Display all surface tolerances: ○ Yes ◉ No

Table 2 of Smith's paper is given in terms of peak-to-valley OPD, so when we compute the change tables, we enter a scale factor of 0.25 for the tolerance units. We need to compute change tables for power (spherical) error, element thickness, air spaces, index of refraction, and surface tilt to obtain the data presented in Table 2. (In the interest of clarity and since we are only interested in spherical aberration, coma, and astigmatism, only the first group of 9 tolerance aberrations is displayed in the output below. OSLO always displays a change table containing entries for all 18 aberrations and system quantities.)

```
*CURVATURE SENSITIVITY ANALYSIS
TOLERANCE UNITS
T (Trans.) =    0.007787   L (Long.) =    0.018488   W (Wvfr.) =    1.0
TOLERANCE THRESHOLD =      --

SRF TOL VAL  TR SPH  AX DMD    COMA FLD DMD    YFS     XFS   DEL BF  AX OPD FLD OPD
NOM            0.02     --     -0.05    --     -0.92   -0.72   -0.59    0.03    0.15
 1  -10.000    0.01     --      0.00    --     -2.69   -2.70    0.00    0.01    0.00
 2   10.000   -0.00     --     -0.01    --      2.68    2.69   -0.00   -0.00   -0.00
 3   10.000   -0.05     --      0.02    --     -2.59   -2.59    0.00   -0.00    0.00
 4   10.000    0.01     --     -0.01    --      2.60    2.61   -0.01    0.01    0.00
 5  -10.000   -0.03     --     -0.01    --     -2.58   -2.58    0.00   -0.01   -0.01
 6  -10.000    0.03     --      0.00    --      2.52    2.52   -0.00    0.03    0.01
 7  -10.000   -0.01     --      0.00    --     -2.53   -2.53    0.01   -0.01   -0.00
 8  -10.000    0.06     --      0.01    --      2.35    2.36   -0.01    0.06    0.03
RSS             0.09     --      0.03    --      7.27    7.28    0.02    0.07    0.04

*ELEMENT THICKNESS SENSITIVITY ANALYSIS
SRF TOL VAL  TR SPH  AX DMD    COMA FLD DMD    YFS     XFS   DEL BF  AX OPD FLD OPD
NOM            0.02     --     -0.05    --     -0.92   -0.72   -0.59    0.03    0.15
 1    0.200   -0.00     --      0.00    --     -0.01   -0.01    0.00   -0.00   -0.00
 3    0.200   -0.03     --     -0.00    --     -0.61   -0.62    0.01   -0.01   -0.01
 5    0.200    0.00     --     -0.04    --     -5.80   -5.81    0.01    0.00   -0.01
 7    0.200   -0.02     --     -0.04    --    -18.17  -18.18    0.02   -0.01   -0.01
RSS             0.04     --      0.06    --     19.09   19.10    0.02    0.01    0.02

*AIR SPACE SENSITIVITY ANALYSIS
SRF TOL VAL  TR SPH  AX DMD    COMA FLD DMD    YFS     XFS   DEL BF  AX OPD FLD OPD
NOM            0.02     --     -0.05    --     -0.92   -0.72   -0.59    0.03    0.15
 2    0.200   -0.02     --      0.02    --     -1.91   -1.92   -0.00   -0.01    0.00
 4    0.200    0.01     --     -0.02    --     -0.14   -0.14    0.00    0.01   -0.00
 6    0.200    0.01     --     -0.06    --     -6.91   -6.92    0.01    0.01   -0.01

*REFRACTIVE INDEX SENSITIVITY ANALYSIS
SRF TOL VAL  TR SPH  AX DMD    COMA FLD DMD    YFS     XFS   DEL BF  AX OPD FLD OPD
NOM            0.02     --     -0.05    --     -0.92   -0.72   -0.59    0.03    0.15
 1   0.0010    0.01     --      0.00    --     -0.92   -0.92    0.00    0.01    0.00
 3   0.0010    0.00     --     -0.00    --      0.40    0.41   -0.00    0.00    0.00
 5   0.0010   -0.01     --     -0.00    --     -0.91   -0.92    0.00   -0.00   -0.00
 7   0.0010   -0.00     --      0.00    --     -0.31   -0.31    0.00   -0.00   -0.00
 9   0.0010    0.00     --      0.00    --      0.10    0.10   -0.00    0.00    0.00
RSS             0.01     --      0.00    --      1.40    1.40    0.00    0.01    0.00

*SURFACE TILT SENSITIVITY ANALYSIS
```

SRF	TOL VAL	TR SPH	AX DMD	COMA	FLD DMD	YFS	XFS	DEL BF	AX OPD	FLD OPD
NOM		0.02	--	-0.05	--	-0.92	-0.72	-0.59	0.03	0.15
1	0.057	0.00	--	0.04	--	-0.01	-0.02	-0.01	0.02	0.01
2	0.057	-0.00	--	-0.07	--	0.05	0.03	0.04	0.04	-0.02
3	0.057	0.00	--	0.17	--	-0.06	-0.04	-0.05	0.13	0.11
4	0.057	0.00	--	-0.08	--	0.05	0.03	0.04	0.05	-0.01
5	0.057	-0.00	--	0.11	--	0.02	0.00	0.02	0.08	0.05
6	0.057	-0.00	--	-0.11	--	0.02	0.01	0.01	0.08	0.00
7	0.057	-0.00	--	0.02	--	-0.03	-0.00	-0.01	0.00	0.02
8	0.057	-0.00	--	-0.06	--	0.01	-0.01	0.00	0.04	-0.02
RSS		0.00	--	0.27	--	0.10	0.06	0.08	0.19	0.13

As mentioned in the paper, the aberrations of concern for this lens are spherical aberration, coma, and astigmatism (since the lens will be refocused for off-axis image points). We can convert the change table values for YFS and XFS to astigmatism by taking their difference. The resulting RSS astigmatism values for the five analyses are

Perturbation	Astigmatism RSS (λ)
Curvature	0.022
Thickness	0.012
Air space	0.015
Refractive index	0.003
Surface tilt	0.060

The RSS totals by aberration are

Aberration	RSS Total (λ)
Spherical	0.103
Coma	0.283
Astigmatism	0.067

The RSS totals by perturbation class are

Perturbation Class	RSS Total (λ)
Radius	0.099
Thickness/Air space	0.103
Refractive index	0.011
Surface tilt	0.273

Thus, the RSS = sqrt(.099**2+.103**2+.011**2+.273**2) = 0.308 λ, in excellent agreement with the analysis in Table 2 of Smith's paper. This value already exceeds the 0.173 λ that we have available in our budget. If we consider the additional affect of one fringe of irregularity, the total OPD variation becomes 0.84λ.

The art of tolerance budgeting comes in when we must reassign the tolerances in an attempt to reduce the total OPD to an acceptable level. The approach taken by Smith is given in Section VI, "Adjusting the Tolerance Budget", which is reproduced below.

> *The OPD of 0.84 wavelengths exceeds the value of 0.288 which we determined in Section III to be the maximum which we could allow in order to maintain the Strehl ratio of 0.75. Since it is too large by a factor of .84/.288 = 2.9X, we could simply reduce our trial budget by this factor across the board. This is not usually the best way.*

An inspection of Table 2 and its footnotes [the change table] indicates that the sensitivity of the tolerances varies widely, ranging from the total insensitivity of coma to the indicated index changes, to significant effects from the radius and thickness changes and very heavy contributions from the assumed surface tilts (or decentrations).

We have previously (in the last paragraph of Section II) noted that the RSS process indicates that the larger tolerance effects are much more significant than the smaller; the significance varies as the square of the size. Thus, a rational approach is to reduce the tolerances on those parameters which are the most sensitive. Conversely, one might also consider increasing the tolerances on those parameters which are relatively insensitive.

This is the technique which we shall apply here. However, there are practical considerations which should be observed. In most optical shops there is a fairly standard tolerance profile. For example, a shop may do most of its work to a five ring test glass fit, a thickness tolerance of ± 0.1 mm, and centering to a one minute deviation. If a larger tolerance is allowed, there will be a saving, but it will not be proportional to the increase in the tolerance. This is because the shop will still tend to produce to its customary profile. They may be able to relax their procedures a bit, and their usual percentage of rejections will drop, but the tendency will be very strong to produce the usual profile whether it has been specified or not. Thus, there is a limit on the increase in tolerance size which will produce a real savings. As another example, many optical glasses are routinely produced to an index tolerance of ± .001 or ± .0015. There is no saving in cost if the tolerance is increased beyond the standard commercial tolerance.

When tolerances are reduced <u>below</u> the "standard profile" however, the cost of fabrication begins to climb. This results from the additional care and effort necessary to hold the tighter tolerances and/or an increase in the rejection rate. In most shops there is effectively a practical limit to the smallness of a given class of tolerance, since the cost of fabrication rises asymptotically toward infinity as this limit is approached.

Thus, for most shops there is both a high limit on tolerances, beyond which there is no savings, and a low limit, which the shop is barely capable of meeting. Obviously, one should confine the tolerance specifications to this range (or find another shop whose capabilities encompass one's requirements).

If we take the RSS of the contributions of each parameter tolerance individually, as we have done in the last column of Table 2 [see the RSS totals by perturbation class table above], then we get a convenient measure of the sensitivity of each tolerance. Examination of the table indicates that the variations of radius, thickness, index and especially surface tilt are all significant contributors to the final RSS OPD. If there are a few very large contributors, a possible general technique would be to reduce any dominant tolerances by a factor approximating the factor by which the OPD of [the] trial budget exceeds the acceptable OPD. Another technique is to make the tolerance size inversely proportional to its sensitivity, so that each tolerance produces the same OPD; this is obviously subject to the limitations outlined above, as well as the necessity to weigh each class of tolerance in some way so as to take into account their different natures and costs.

Following this line, we get the following budget, for which the RSS OPD is 0.167 λ, just slightly better than the 0.173 λ required for our Strehl ratio specification of 75%.

```
*TOLERANCES
14mm Laser Recording Lens
               RADIUS    FRINGES          THICKNESS           INDEX DECEN TILT
 SRF    RADIUS   TOL    SPH   IRR THICKNESS  TOL       GLASS   TOL   TOL   TOL
  1   50.36600    0.0  1.00  0.20   2.80000 0.1000      SF11 0.0010   0.0 0.011
  2  -39.04500    0.0  1.00  0.20   0.43527 0.0200       AIR    0.0   0.0 0.011

  3  -19.83600    0.0  1.00  0.20   2.00000 0.0500      SF11 0.0010   0.0 0.011
  4  -34.36000    0.0  1.00  0.20   0.20000 0.0400       AIR    0.0   0.0 0.011

  5   17.42000    0.0  1.00  0.20   2.65000 0.0500      SF11 0.0010   0.0 0.011
  6   79.15000    0.0  1.00  0.20  11.84000 0.0300       AIR    0.0   0.0 0.011

  7    7.08000    0.0  1.00  0.20   2.24000 0.0700      SF11 0.0010   0.0 0.011
```

```
 8   15.66500    0.0  1.00  0.20   3.18200    0.0       AIR    0.0   0.0 0.011

 9        0.0    0.0   0.0   0.0   2.03200    0.0     ACRYL    0.0   0.0   0.0
10        0.0    0.0   0.0   0.0       0.0    0.0       AIR    0.0   0.0   0.0

11        0.0    0.0   0.0   0.0       0.0    0.0                    0.0   0.0
FRINGE WAVELENGTH:    0.589290
```

Fringes measured over clear aperture of surface unless indicated.

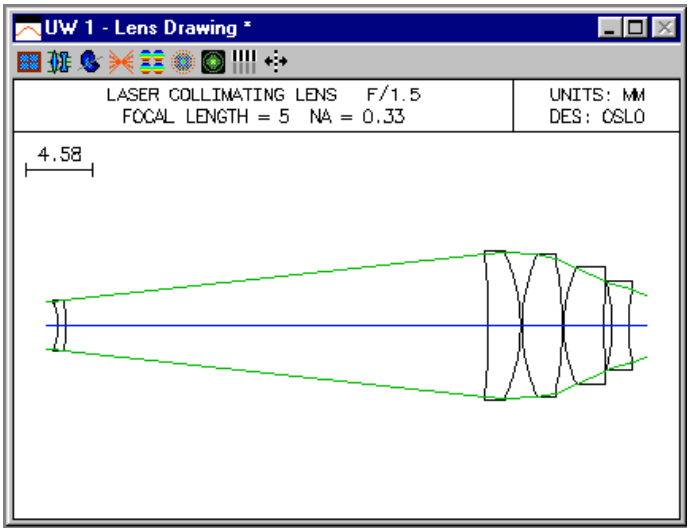
Wavefront/MTF tolerancing

As an example of using these tolerance routines, we will compute tolerance effects for a scaled version of a laser collimating lens from the OSLO library. The base lens we want to open is the public lens file "\len\lib\smithgen\ch22\ho5mmc.len". When opened, this lens has a focal length of 100 mm. The system notes tell us that the lens was designed for a focal length of 5 mm. We can restore this focal length by right-clicking in the surface data spreadsheet, selecting Scale Lens from the menu, and choosing "Scale lens to new focal length" from the fly-out menu, and entering "5" for the new focal length. This lens is designed for use with a Helium-Cadmium laser at a wavelength of 0.4416 μm, so only 1 wavelength is defined. Also, since we are working with a collimator, we will only consider the on-axis performance. Click the Field Points button in the surface data spreadsheet to open the Field Points spreadsheet, and delete field points 2 and 3. In the Setup spreadsheet, set the object distance to 1.0×10^{20} and the field angle to 5.7×10^{-5} degrees (n.b. if you enter 0.0 for the field angle, it will be set automatically to this value, since 0.0 field is not allowed). In order to compare the your results to the ones shown here, move the focus shift (0.014 mm) from surface 13 to the image surface, and delete surface 13. The specification for the modified lens is given below.

```
*LENS DATA
LASER COLLIMATING LENS   F/1....
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ       --        1.0000e+20    1.0000e+14            AIR

 AST       --           --          1.666655 AK          AIR

  2     -4.241940      0.552322      1.666655 K           BK7 C
  3     -8.586057     28.962576      1.769039 K           AIR

  4    -57.502645      2.209292      4.952020 K           BK7 C
  5    -12.502519      0.110464      5.088569 K           AIR

  6     12.244025      2.761618      4.915560 K           BK7 C
  7    -22.689595      0.110464      4.668527 K           AIR

  8      9.937532      2.761613      4.001831 K           BK7 C
  9    139.154768      0.546917      3.111206 K           AIR

 10    -11.530153      1.104643      3.026087 K           BK7 C
 11     11.290666        --          2.494580 K           AIR

 12        --          7.733278 S    2.578630 K           AIR

 IMS       --         -0.014000      0.004625 S

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.441600
        1.000000

*PARAXIAL CONSTANTS
   Effective focal length:    5.000000    Lateral magnification:  -5.0000e-20
   Numerical aperture:        0.329998    Gaussian image height:   5.0000e-06
   Working F-number:          1.515161    Petzval radius:          226.799242
   Lagrange invariant:     -1.6500e-06
```

From a spot diagram, we see that the on-axis RMS OPD for the nominal design is 0.05 waves.

```
*WAVEFRONT
 WAVELENGTH 1
   PKVAL OPD     RMS OPD  STREHL RATIO    YSHIFT      XSHIFT      RSZ
    0.214591    0.049599    0.908313      --          --          --
```



We will compute the sensitivity of the on-axis RMS OPD to decentration of the five components in the lens. We assign a trial decentration tolerance of 10 μm to surfaces 2, 4, 6, 8, and 10 (i.e., the front surfaces of the components). OSLO assumes that the component decentrations have a Gaussian distribution that is truncated at the $2\sigma = 10$ μm point.

	DECENTRATION		CLEAR APERTURE TILT		CENTER OF CURV TILT	
SRF	DCY	DCX	ALPHA	BETA	ALPHA	BETA
2	0.010000	0.010000	0.000000	0.000000	0.500000	0.500000
3			0.000000	0.000000	0.000000	0.000000
4	0.010000	0.010000	0.000000	0.000000	0.120000	0.120000
5			0.000000	0.000000	0.000000	0.000000
6	0.010000	0.010000	0.000000	0.000000	0.320000	0.320000
7			0.000000	0.000000	0.000000	0.000000
8	0.010000	0.010000	0.000000	0.000000	0.500000	0.500000
9			0.000000	0.000000	0.000000	0.000000
10	0.010000	0.010000	0.000000	0.000000	0.250000	0.250000
11			0.000000	0.000000	0.000000	0.000000

Tilt tolerances are specified in degrees.

From the Options menu, select MTF/Wvf Tolerancing. In the spreadsheet, select RMS wavefront tolerancing, sensitivity mode, and perturbation equation output. The tolerance item is component decentration and we want to compute the wavefront in wavelength 1. The resulting sensitivity output is shown below.

```
*RMS WAVEFRONT SENSITIVITY ANALYSIS - WAVELENGTH 1
 THRESHOLD CHANGE FOR INDIVIDUAL TOLERANCE DISPLAY:     0.010000
 TOLERANCE  SRF/ TOLERANCE              CHANGE IN RMS
    ITEM    GRP    VALUE  GRD CFG FPT    PLUS     MINUS         A           B
 CMP DEC Y   2    0.01     A   1   1    0.014     0.014      0.001572 -4.7537e-19
 CMP DEC Y   4    0.01     A   1   1    0.058     0.058      0.009040 -4.3880e-19
 CMP DEC Y   6    0.01     A   1   1    0.098     0.098      0.019341  9.5073e-19
 CMP DEC Y  10    0.01     A   1   1    0.131     0.131      0.030168 -1.1701e-18

 Note: Only tolerances that result in a performance change
       of at least 0.01 are displayed.
       No compensators have been used for this analysis.

 ------------------------------------------------------------------------------
 RMS WAVEFRONT ERROR
                                          NOMINAL  HIGH RMS    MEAN     STD DEV
   CFG FPT    FBY       FBX       FBZ       RMS     W/ TOLS   CHANGE    (SIGMA)
    1   1    --        --        --       0.050     0.222     0.054     0.060
 ------------------------------------------------------------------------------
```

From the sensitivity data, we note that all of the *B* coefficients are zero and that all of the *A* coefficients are positive. This means there is no linear term in the second-order expansion of RMS

OPD as a function of component decentration. These results are not surprising, since any decentration destroys the rotational symmetry of the system and we would expect that this would degrade the performance of the lens. This analysis is a good example of the inadequacy of a linear perturbation model for some tolerance criteria. As discussed above, the non-zero *A* coefficients result in a non-zero average change in performance, in this case, of about 0.12 λ.

After the display of the sensitivity data, a performance summary is shown. For each field point, four items are displayed. First is the nominal value of the RMS wavefront error. Second is the estimated high value of RMS wavefront error, with the tolerances applied. This value is taken to be the mean-plus-two-sigma value of the resulting distribution of systems. Finally, the mean change and standard deviation (sigma) of the performance measure are shown.

The level of performance degradation indicated above is probably not acceptable for this lens, which should operate at diffraction-limited or near diffraction-limited performance. We will attempt to rebudget the component decentration tolerances such that the upper limit of the RMS wavefront change corresponds to the Strehl tolerance limit of 0.8, or an RMS wavefront error of 0.07 λ. Since the nominal design has an RMS OPD of 0.049 λ, the maximum change is 0.021 λ.

Before carrying out this analysis, it is necessary, because of the short focal length of the lens, to reset the smallest allowed tolerance and the tolerance increment, which are nominally both 0.01. Use the Tolerance>>Update Tolerance Data>>Grades command to open the spreadsheet, and reset the minimum component decentration to 0.001, and the increment to 0.0001, as shown below.

CMP DECEN	0.0010	0.5000	0.0001	0.0300	0.2000	Gaussian
CMP TILT	0.0100	0.5000	0.0100	0.0500	0.3333	Gaussian

Now run the tolerancing analysis again, but this time in inverse sensitivity mode, with a requested change in RMS OPD of 0.01 λ.

```
*RMS WAVEFRONT INVERSE SENSITIVITY ANALYSIS - WAVELENGTH 1
 DIFFERENTIAL CHANGE FOR CALCULATION:    0.010000
 THRESHOLD CHANGE FOR INDIVIDUAL TOLERANCE DISPLAY:     0.001000
 TOLERANCE  SRF/ ALLOWED                CHANGE IN RMS
    ITEM    GRP TOLERANCE GRD CFG FPT    PLUS     MINUS         A           B
 CMP DEC Y   2    0.0083   A   1   1    0.010     0.010      0.001083 -3.9455e-19
 CMP DEC Y   4    0.0035   A   1   1    0.010     0.010      0.001107 -1.5358e-19
 CMP DEC Y   6    0.0024   A   1   1    0.010     0.010      0.001114  2.2818e-19
 CMP DEC Y   8    0.0141   A   1   1    0.010     0.010      0.001092  5.1559e-20
 CMP DEC Y  10    0.0019   A   1   1    0.010     0.010      0.001089 -2.2232e-19

 Note: Only tolerances that result in a performance change
       of at least 0.001 are displayed.
       No compensators have been used for this analysis.

 ---------------------------------------------------------------------------
 RMS WAVEFRONT ERROR
                                       NOMINAL  HIGH RMS    MEAN     STD DEV
   CFG FPT   FBY       FBX       FBZ     RMS     W/ TOLS   CHANGE    (SIGMA)
    1   1    --        --        --     0.050     0.082     0.008     0.012
 ---------------------------------------------------------------------------
```

From the above, we see that that the fourth component is the least sensitive to decentration, while the fifth component is most sensitive. The computed allowed tolerances yield a maximum (mean plus two standard deviations) change of 0.08 λ, mor than we want. Based on this, we try the following budget

	DECENTRATION		CLEAR APERTURE TILT		CENTER OF CURV TILT	
SRF	DCY	DCX	ALPHA	BETA	ALPHA	BETA
2	0.005000	0.005000	0.000000	0.000000	0.500000	0.500000
3			0.000000	0.000000	0.000000	0.000000
4	0.003000	0.003000	0.000000	0.000000	0.120000	0.120000
5			0.000000	0.000000	0.000000	0.000000
6	0.002000	0.002000	0.000000	0.000000	0.320000	0.320000
7			0.000000	0.000000	0.000000	0.000000
8	0.010000	0.010000	0.000000	0.000000	0.500000	0.500000
9			0.000000	0.000000	0.000000	0.000000
10	0.002000	0.002000	0.000000	0.000000	0.250000	0.250000
11			0.000000	0.000000	0.000000	0.000000

Tilt tolerances are specified in degrees.

The resulting sensitivity analysis is shown below.

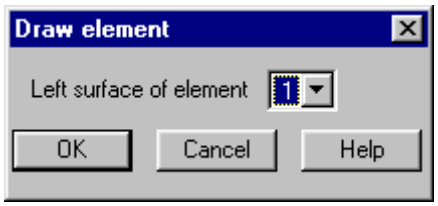
```
*RMS WAVEFRONT SENSITIVITY ANALYSIS - WAVELENGTH 1
 THRESHOLD CHANGE FOR INDIVIDUAL TOLERANCE DISPLAY:     0.001000
 TOLERANCE  SRF/ TOLERANCE                CHANGE IN RMS
    ITEM    GRP    VALUE  GRD CFG FPT    PLUS     MINUS         A           B
 CMP DEC Y   2    0.005    A   1   1    0.004     0.004      0.000393 -2.3768e-19
 CMP DEC Y   4    0.003    A   1   1    0.008     0.008      0.000814 -1.3164e-19
 CMP DEC Y   6    0.002    A   1   1    0.007     0.007      0.000774  1.9015e-19
 CMP DEC Y   8    0.01     A   1   1    0.005     0.005      0.000550  3.6567e-20
 CMP DEC Y  10    0.002    A   1   1    0.011     0.011      0.001207 -2.3403e-19

 Note: Only tolerances that result in a performance change
       of at least 0.001 are displayed.
       No compensators have been used for this analysis.

 ------------------------------------------------------------------------------
 RMS WAVEFRONT ERROR
                                          NOMINAL  HIGH RMS    MEAN     STD DEV
   CFG FPT   FBY       FBX       FBZ       RMS     W/ TOLS   CHANGE    (SIGMA)
    1   1    --        --        --       0.050     0.073     0.006     0.009
 ------------------------------------------------------------------------------
```

The upper end of the range of RMS wavefront change with this budget is 0.028 λ, slightly larger than our target of 0.021 λ. We have not allowed for any compensating parameters during this analysis, so the resulting tolerances are extremely small. This example has been presented to illustrate the types of calculations and analysis that can be performed, not as an example of a complete analysis of a manufacturable lens.

ISO 10110 Element Drawing

ISO 10110 is an international standard titled “Preparation of drawings for optical elements and systems”. It prescribes not only how optical drawings should appear, but also how constructional data and tolerances should be specified. OSLO uses ISO 10110 recommendations for aspheric surface forms, default tolerances, and element drawings. Currently, the element drawing routines are limited to rotationally symmetric lenses with spherical surfaces, specified according to Part 10 of the standard, “Table representing data of a lens element”.

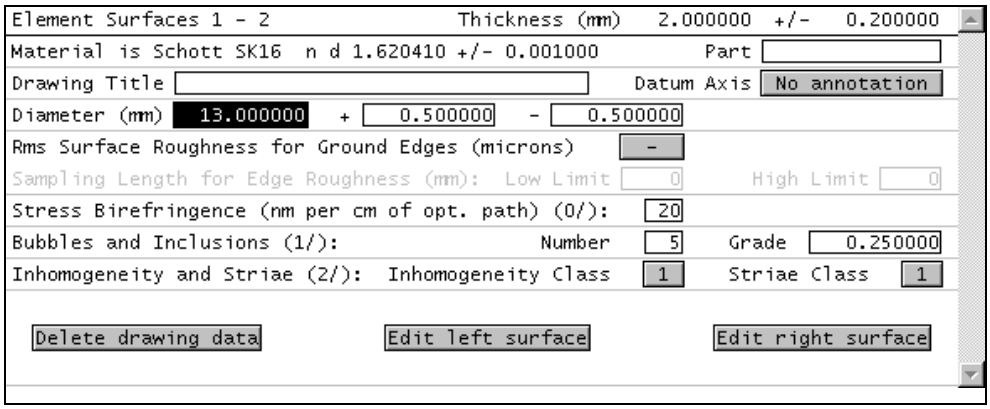
Element drawings are prepared using the Lens >> Lens Drawing >> Element. This command will bring up a dialog box with a drop-down list showing the first surfaces of all the elements in the current lens. If the system contains tilted or reflecting surfaces, the list may not be accurate.



After you select a surface, the program shows a spreadsheet that contains the items that need to be specified for the element as a whole. Most of the fields are set to default values obtained from the lens data, ISO 10110, or the tolerance data. In addition to these sources of data, the address preferences (**ad1**-**3**) are used to fill in the title block on element drawings.

Each field contains an English description of the data to be provided, and where appropriate, a numerical code (e.g. 1/) according to the standard. Some fields have lists that include values recommended in the standard, and these are indicated by buttons.

At the bottom of the initial dialog box are three buttons. The first deletes all element data from a surface (such data is normally stored with the lens). The second and third buttons bring up a dialog box for editing the left and right surfaces of the lens, as shown on the next page. After you have completed any necessary data entry, close the spreadsheet with the Green check, and the drawing will be produced in the current graphics window. For hard copy, you can right-click in the graphics window and print it in the normal way.



Surface 2 Element dimensions are in mm.

Radius 158.650000 CX +/- 0.000000 Centring Tol. (minutes) (4/) 20.000000

Optically Eff. Diameter 13.000000 Prot. Chamfer 0.200000 --> 0.400000

Surface Polishing Grade P Rms Surface Roughness - Rq (nm) -

Sampling Length for Rq (microns): Low Limit 0 High Limit 0

Surface Form Deviations - 3/A(B/C):

A = Sagitta error, B = Irregularity, C = Rotationally symmetric irregularity

A (fringes) 10.000000 B (fringes) 2.000000 C (fringes) 0.000000

RMS Residual Surface Deviations (3/):

RMSt = Total rms deviation, RMSi = Rms irregularity, RMSa = Rms asymmetry

RMSt < 0.000000 RMSi < 0.000000 RMSa < 0.000000

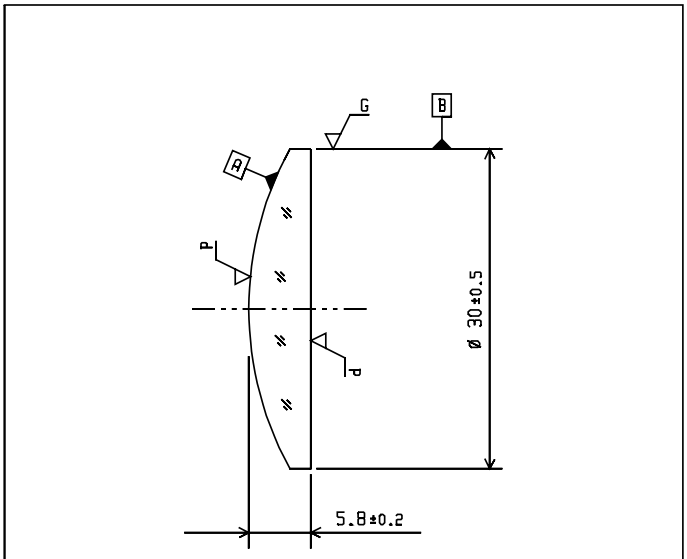
Surface Imperfection Tolerances (5/):

Surface Imperfections	Number	5	Grade	0.250000
Coating Blemishes	Number	0	Grade	0.000000
Long Scratches	Number	0	Max Width (mm)	0.000000
Edge Chips	Maximum Extent From Edge (mm)			0.000000

Surface Treatment/Coating Specification

Laser Irradiation Damage Threshold (6/)

This dialog box is similar to the first, in that default values are obtained from the lens data where possible (if these data are to be changed, they must be changed in the appropriate source, e.g. tolerance data must be changed in the tolerance spreadsheet). The following is a drawing of the single element used for this example.



Left Surface:	Material Specification:	Right Surface:
R 31.123 CX øe 30.0 Prot. Cham. 0.2 - 0.4 Ⓛ - 3/ 10(2) 4/ 0' 5/ 5x0.25 6/ -	Schott BK7 n d 1.5168±0.001 v - 0/ 20 1/ 5x0.25 2/ 1:1	R ∞ øe 30.0 Prot. Cham. 0.2 - 0.4 Ⓛ - 3/ 10(2) 4/ 20' 5/ 5x0.25 6/ -

Element Drawing Example		Center for Optics Manufacturing University of Rochester Rochester, New York 14623
A-1996.0306A	Ind. acc. ISO 10110	

The table below gives a brief summary of the data meanings. For additional information, please consult the standard or the OSA ISO 10110 User's Guide.

In the U.S., copies of standards are available from

American National Standards Institute
11 West 42nd Street
New York, NY 10036
tel (212) 642-4900 fax (212) 302-1286

The "OSA User's Guide for ISO 10110" is available from

Optical Society of America
2010 Massachusetts Avenue, NW
Washington, DC 20036
tel (202) 223-8130
fax (202) 223-1096

Summary of ISO 10110 drawing codes	
Property and code form	**Data**
Stress birefringence 0/A	A = Maximum optical path difference (nm/cm)
Bubbles & inclusions 1/NxA	N = Number of bubbles A = Bubble grade number
Inhomogeneity and striae 2/AB	A = Homogeneity class B = Striae class
Surface form tolerance 3/A(B/C) or 3/A(B/C) RMSx < D or 3/- RMSx < D x is either t, i, or a	A = Maximum sagitta error B = Peak-to-valley irregularity C = Non-spherical, rotationally symmetric error D = Maximum rms tolerance t = total rms deviation from nominal surface i = rms irregularity a = rms asymmetry after removal of spherical and rotationally symmetric irregularity A,B,C,D in fringes (default λ= 0.5461μm)
Centering tolerance 4/σ	σ = Surface tilt angle (minutes or seconds)
Surface imperfection tolerance 5/N x A; CN′ x A′; LN″ x A″; EA‴	N = Number of allowed scratches A = Defect grade number (mm) N′ = Number of coating blemishes A′ = Coating blemish grade number (mm) N″ = Number of long scratches (> 2mm) A″ = Maximum width of scratch A‴ = Maximum extent of edge chips (mm)
Laser damage threshold 6/H_{th}; λ; pdg; f_p; n_{ts} x n_p (pulsed) or 6/E_{th}; λ; n_{ts} (cw)	H_{th} = Energy density threshold (J/cm^2) E_{th} = Power density threshold (W/cm^2) λ = Laser wavelength pdg = Pulse duration group f_p = Pulse repetition frequency n_{ts} = Number of test sites n_p = Number of pulses per site

Reflecting systems

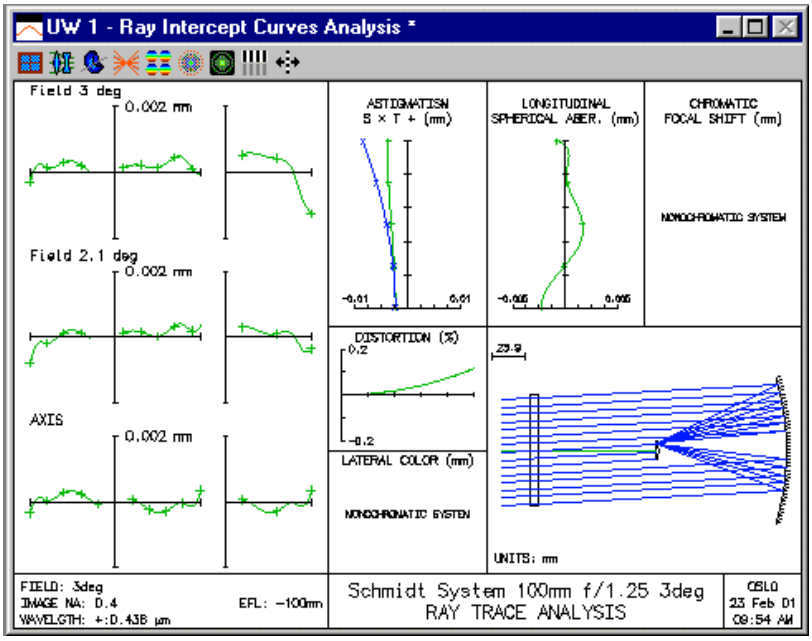
Schmidt camera

A Schmidt camera consists of a spherical mirror with an aspheric corrector plate at or near the center of curvature of the mirror. The purpose of the aspheric plate is to correct the spherical aberration of the mirror. In the design included here, an additional lens is placed near the image plane to correct field aberrations on a flat image plane (a normal Schmidt camera has a curved image surface). The system data is shown below.

```
*LENS DATA
Schmidt System 100mm f/1.25 3deg
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    5.2408e+18            AIR

  1    2.6134e+03      6.500000     42.000000         FSILICA   *
  2        --         88.660000     42.000000 AP          AIR

  3        --         98.860000     42.000000 X           AIR   *

  4   -201.029000    -96.860000 P   55.000000      REFL_HATCH

  5    -40.500000     -2.000000      7.000000         FSILICA P
  6    234.033000     -0.416671      7.000000 P           AIR P

  7        --            --          5.300000

*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF        CC          AD          AE          AF          AG
  1      -1.000000 -5.8678e-08 -4.2209e-12  1.2779e-15 -3.0697e-19

*PICKUPS
  2     AP     1
  4     THM    3        2.000000
  5     GLA    1
  6     GLA    3      AP     5

*APERTURES
 SRF   TYPE APERTURE RADIUS
  0     SPC   5.2408e+18
  1     SPC    42.000000
  2     PKP    42.000000
  3     SPC    42.000000
     Special Aperture Group 0:
     A  ATP      Ellipse  AAC     Obstruct  AAN       --
        AX1    -7.000000  AX2     7.000000  AY1    -7.000000  AY2     7.000000
  4     SPC    55.000000
  5     SPC     7.000000
  6     PKP     7.000000
  7     SPC     5.300000
```

Note that an obstruction must be placed on surface 3 to block the rays that hit the image surface before hitting the primary mirror. The obstruction is specified as a special aperture as shown, and is marked not drawable, so that it does not appear in the lens drawing.

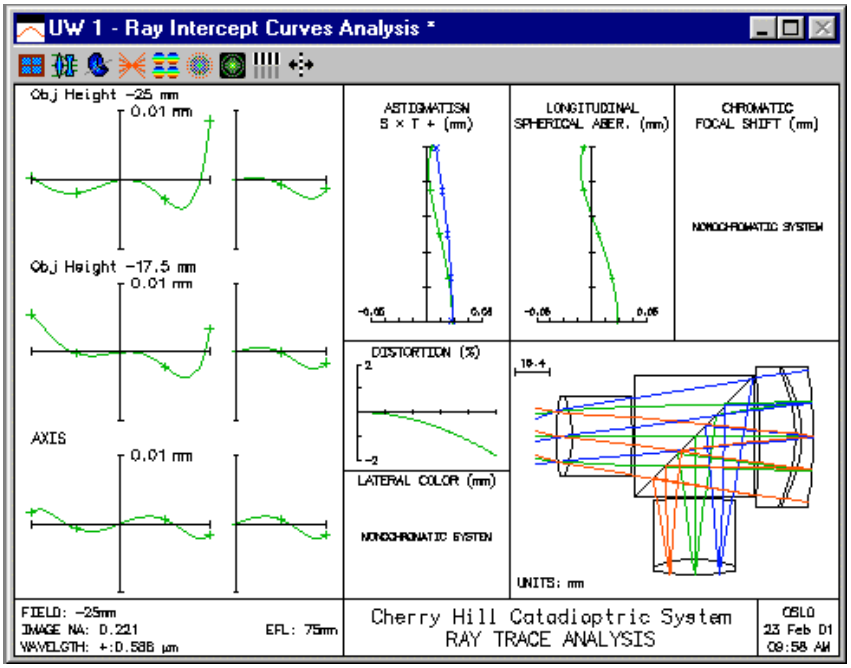
The performance of the system is indicated in the ray analysis below. Note that the ray-intercept curves are not smooth. This is characteristic of designs that use high-order aspherics, as this one does. It is likely that the curves could be smoothed out by careful weighting of the variables during optimization.


UW 1 - Ray Intercept Curves Analysis *
Field 3 deg
0.002 mm
Field 2.1 deg
0.002 mm
AXIS
0.002 mm
ASTIGMATISM
S x T + (mm)
-0.01
0.01
LONGITUDINAL
SPHERICAL ABER. (mm)
-0.005
0.005
CHROMATIC
FOCAL SHIFT (mm)
MONOCHROMATIC SYSTEM
DISTORTION (%)
0.2
-0.2
LATERAL COLOR (mm)
MONOCHROMATIC SYSTEM
23.9
UNITS: mm
FIELD: 3deg
IMAGE NA: 0.4
WAVELGTH: +:0.438 µm
EFL: -100mm
Schmidt System 100mm f/1.25 3deg
RAY TRACE ANALYSIS
CSLO
23 Feb 01
09:54 AM

Reversible catadioptric lens

This lens is named after the 1985 International Lens Design Conference in Cherry Hill, NJ, where it had the best performance of all designs submitted to the reversible lens design contest. The rules of that contest specified that the design had to meet certain paraxial specifications, and had to have the best performance under the condition that it be completely reversible, in the sense that it had to have identical performance whether it was used forwards or backwards. Most of the submitted solutions were conventional lens forms, but this lens and a few others like it were catadioptric designs that achieved very high performance by using a mirror to accomplish most of the focusing power.

The design is included in the demo examples because it shows the use of pickups to impose constraints required for designing catadioptric systems, particularly ones like this one that have additional reversibility conditions. Please open the file to see the detailed data; there is not enough room for a listing here.

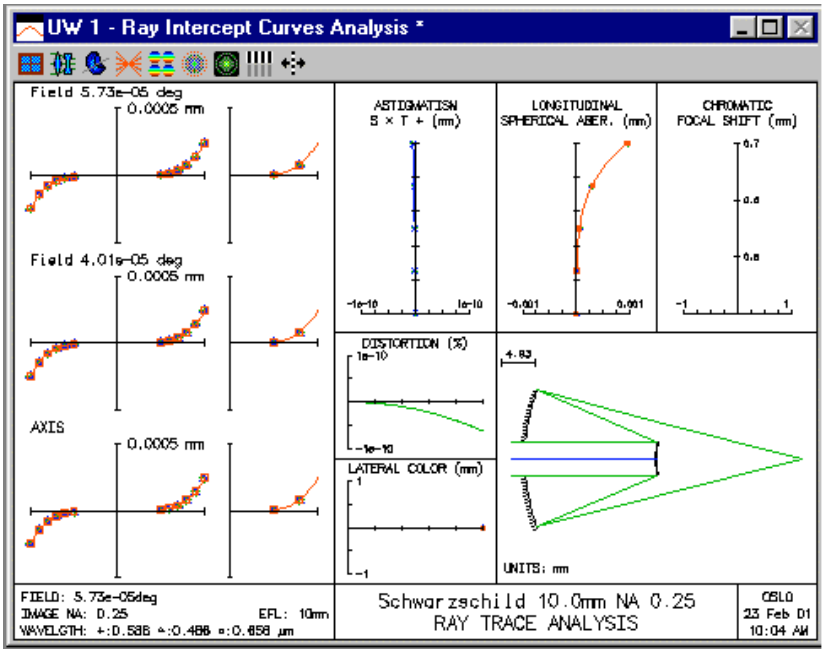


The design is also a good one to study to learn the sign conventions used in reflecting systems. Finally, note that an elliptical aperture is required on the beam-splitter surface to accommodate the tilted surface in a cylindrical geometry.

Schwarzschild objective

The Schwarzschild objective is a convex-concave mirror system in which the mirrors are approximately concentric, with the long conjugate on the convex mirror side. Originally intended as a microscope objective, the design is widely used in a variety of two-mirror relay systems. The archetype Schwarzschild system can be designed analytically (there is a star command *schwarz that does this), but for practical use, variations are usually required.

The design included here is a basic system designed using the *schwarz star command. The ray analysis below shows that the third-order spherical and coma aberrations are corrected, but there is high-order over-corrected spherical. This can be corrected by perturbing the design. Also, the obscuration can be decreased, usually at the expense of increased coma.



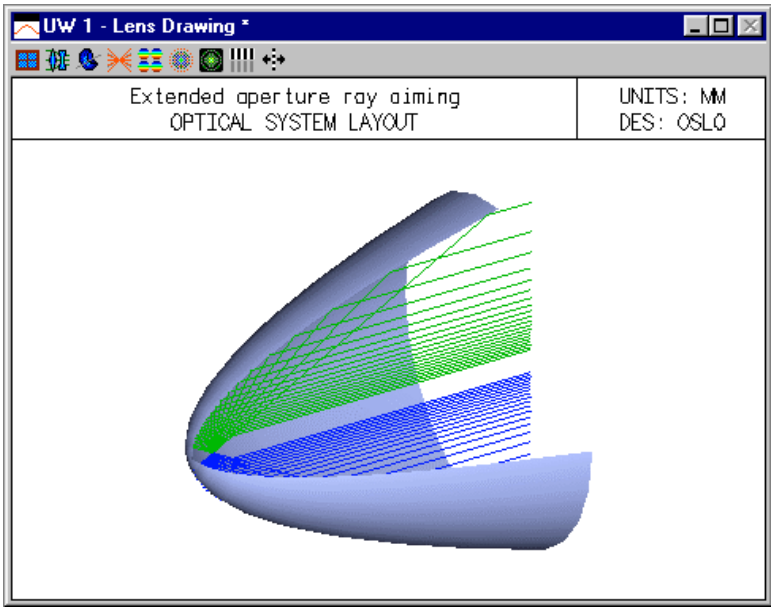
Extended aperture parabola

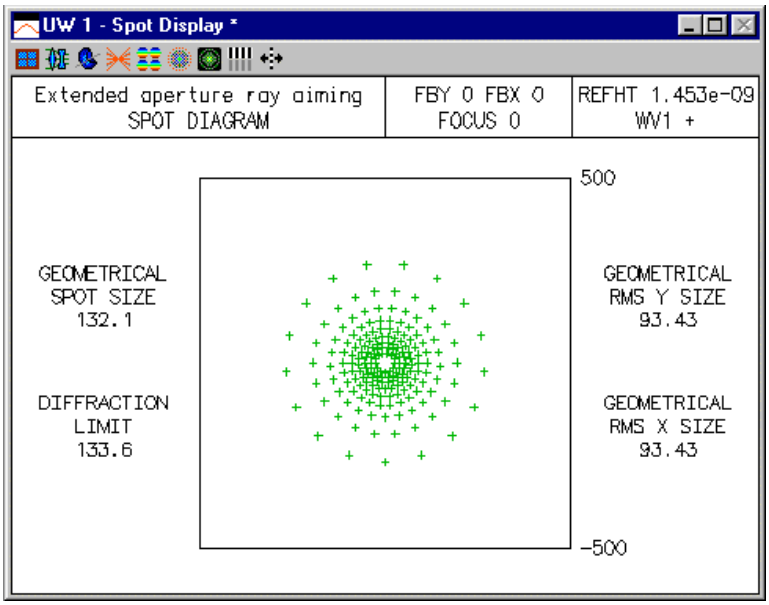
An increasing number of optical systems require ray tracing beyond 90 degrees. The nikofish.len example shows how OSLO accommodates optical systems that have a field of view greater than 90 degrees. This example shows how OSLO accommodates optical systems that have an aperture greater than 90 degrees. Often, such systems are found in illumination applications where high-quality image formation is not required, such as the design of luminaires or light collectors.

The general operating condition **xarm** (extended aperture ray aiming mode) sets up a system of fractional aperture coordinates based on a maximum beam angle (**xaba**), measured relative to the positive *z*-axis at the object surface. When **xarm** is on, fractional coordinates are measured similarly to the tilt angles tla and tlb. A fractional coordinate of +1.0 corresponds to a ray angle equal to xaba above the *z*-axis, and a fractional coordinate of -1.0 corresponds to a ray angle equal to xaba below the *z*-axis.

Extended aperture ray aiming is limited to tracing single rays and computing spot diagrams. There is no reference ray, so analyses such as ray fans have no meaning. The figure below shows a parabolic mirror used in conjunction with extended aperture ray tracing. Since the mirror is to the left of the object, it must be an alternate intersection surface. Note that the mirror has a central obstruction to block rays.

```
*LENS DATA
Extended aperture ray aiming
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        -50.000000    1.0000e-06             AIR

  1    100.000000    2.0000e+03    380.000000 AKX  REFL_HATCH   *

  2        --            --        380.000000

*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF        CC          AD          AE          AF          AG
  1      -1.000000      --          --          --          --

*SURFACE TAG DATA
  1     ASI   1

*APERTURES
 SRF   TYPE APERTURE RADIUS
  0     SPC   1.0000e-06
  1     SPC   380.000000  CHK
     Special Aperture Group 0:
     A  ATP      Ellipse  AAC     Obstruct  AAN       --
        AX1   -24.000000  AX2    24.000000  AY1   -24.000000  AY2    24.000000
  2     SPC   380.000000

*OPERATING CONDITIONS: GENERAL
   Image surface:                    2    Aperture stop:                    1
.
   Telecentric entrance pupil:     Off    Wide-angle ray aiming mode:     Off
   Aper check all GRIN ray segs:   Off    Extended-aper ray aiming mode:   On
   Plot ray-intercepts as H-tan U: Off    XARM beam angle:         150.000000
   Source astigmatic dist:          --    Ray aiming mode:          Aplanatic
   Temperature:             20.000000    Pressure:                  1.000000
```


UW 1 - Lens Drawing *
Extended aperture ray aiming
OPTICAL SYSTEM LAYOUT
UNITS: MM
DES: OSLO

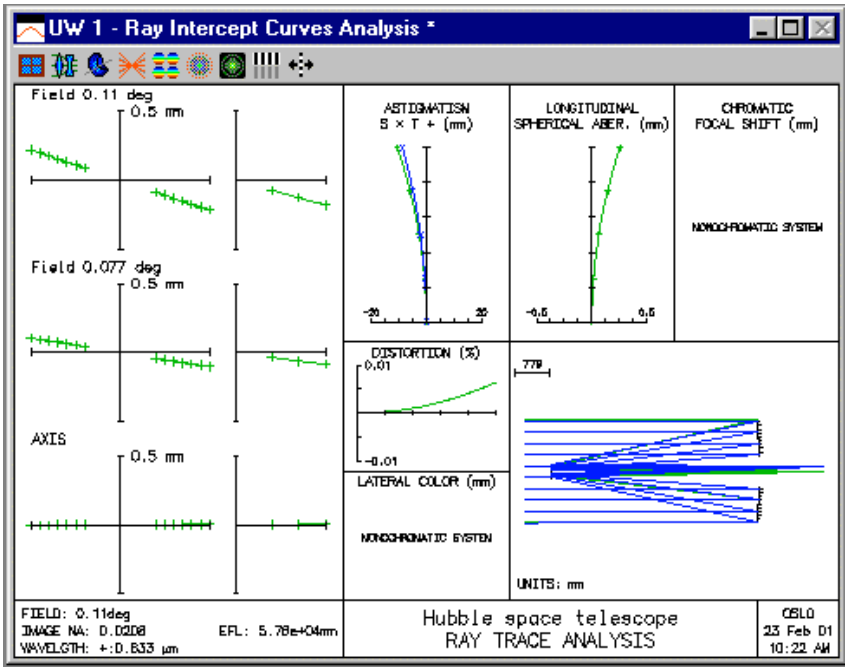

UW 1 - Spot Display *
Extended aperture ray aiming
SPOT DIAGRAM
FBY 0 FBX 0
FOCUS 0
REFHT 1.453e-09
WV1 +
500
GEOMETRICAL
SPOT SIZE
132.1
GEOMETRICAL
RMS Y SIZE
93.43
DIFFRACTION
LIMIT
133.6
GEOMETRICAL
RMS X SIZE
93.43
-500

Hubble space telescope

The Hubble telescope has become famous for many reasons, not the least of which is the well-known fabrication error that has now been compensated. The version included here is the original Ritchey-Chrétien design including two hyperbolic mirrors. The design itself is ideal, but it was fabricated in a way that produced about 6 waves of spherical aberration due to an improper conic constant on the primary mirror. You may wish to experiment to find the deviation in the conic constant that would cause such aberration, as well as ways to compensate for it.

In setting up the system, a wavelength of 0.6328 was used, because that is the wavelength of the light used to test it. Note that surface 1 is needed to serve as a central obstruction.

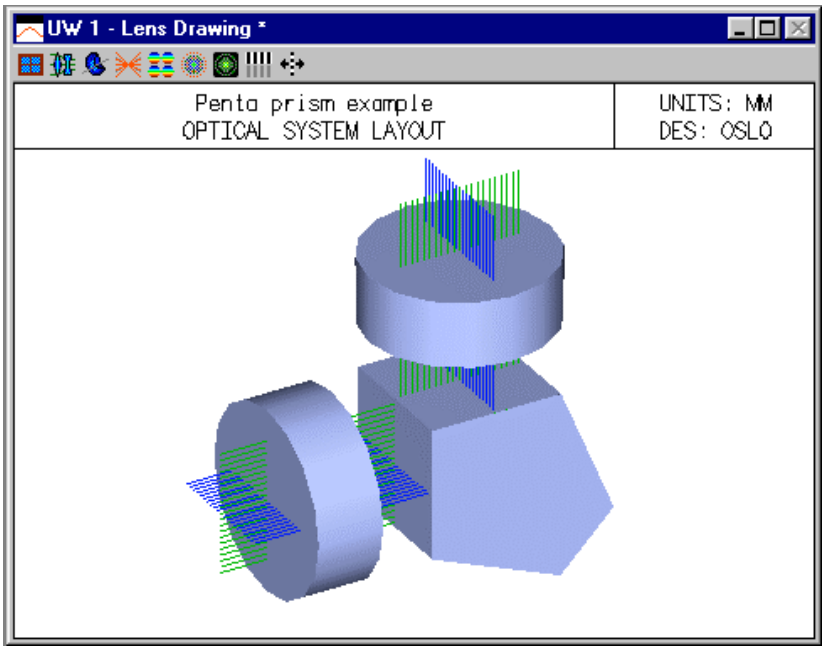
```
*LENS DATA
Hubble space telescope
 SRF      RADIUS      THICKNESS    APERTURE RADIUS       GLASS SPE   NOTE
  0         --        1.0000e+20    1.9199e+17             AIR

  1         --        4.9061e+03    1.2094e+03 SX          AIR    *

  2   -1.1040e+04   -4.9061e+03 P   1.2000e+03 AX    REFL_HATCH    *

  3   -1.3580e+03    6.4059e+03 S   150.000000       REFL_HATCH    *

  4         --            --        110.578590 S

*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF        CC           AD           AE           AF           AG
  2      -1.002299       --           --           --           --
  3      -1.496000       --           --           --           --
```



Pentaprism

A pentaprism changes the direction of a beam by 90 degrees. It is straightforward to enter such a system using the bend command in OSLO. However, to draw a picture of the system requires special work because of the nature of the prism. The system shown here uses OSLO's boundary data information (**bdi**) to draw the prism, as shown in the figure below.

The pentaprism system included here is intended as a base system into which you can insert your own optics as required. Blank pieces of glass are placed in front and behind the prism for demonstration. The data are shown in the following list.




```
*LENS DATA
Penta prism example
 SRF      RADIUS      THICKNESS    APERTURE RADIUS       GLASS SPE   NOTE
  0         --        1.0000e+20    1.0000e+14             AIR

  1         --          1.000000      1.414214 A           BK7 C *
  2         --          1.000000      1.414214             AIR     Your lens

  3         --          2.414214      1.000000 X          BAK1 C *
  4         --         -2.000000      1.082400 X       REFLECT   *
  5         --          2.414214      1.082400 X       REFLECT   *
  6         --          1.000000      1.000000 X           AIR   * Penta prism

  7         --          1.000000      1.414214             BK7 C *
  8         --            --          1.414214             AIR     Your lens

  9         --            --          1.414214
```

From an optical standpoint, the interesting surfaces are 3-6. The orientation of the surfaces is straightforward and easily handled by the **ben** command. The sides of the prism are rectangular, so each surface must have a rectangular special aperture, calculated according to the standard pentaprism geometry. Since the actual prism surfaces are to be represented by **bdi** information (in contrast to being enclosed in a bounding box, it is important
that the drawing information accurately represent the true situation.

```
*TILT/DECENTER DATA
  4     DT    1         DCX        --        DCY       --       DCZ       --
        BEN             TLA     22.500000    TLB       --       TLC       --
  5     DT    1         DCX        --        DCY       --       DCZ       --
        BEN             TLA     22.500000    TLB       --       TLC       --

*APERTURES
 SRF    TYPE APERTURE RADIUS
  0     SPC   1.0000e+14
  1     SPC     1.414214
```

```
  2     SPC     1.414214
  3     SPC     1.000000
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -1.000000  AX2     1.000000  AY1    -1.000000  AY2     1.000000
  4     SPC     1.082400
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -1.082400  AX2     1.082400  AY1    -1.082400  AY2     1.082400
  5     SPC     1.082400
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -1.082400  AX2     1.082400  AY1    -1.082400  AY2     1.082400
  6     SPC     1.000000
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -1.000000  AX2     1.000000  AY1    -1.000000  AY2     1.000000
  7     SPC     1.414214
  8     SPC     1.414214
  9     SPC     1.414214
```

For the drawing, surfaces 3-6 are marked “not drawn” in the special data surface control spreadsheet:

```
*SURFACE TAG DATA
  1     LMO ELE  (2 surfaces)
  3     LMO ELE  (4 surfaces)
  3     DRW OFF
  4     DRW OFF
  5     DRW OFF
  6     DRW OFF
  7     LMO ELE  (2 surfaces)
```

There is no spreadsheet for entering boundary data. You can use the normal lens editor in command mode, giving the commands

```
len upd
gto 3
bdi 16 9
vx 1 -1 -1 0 0
vx 2 -2 2 0 0
.
pf 1 1 2 3 4
pf 2  5 6 7 8
.
etc. according to the list below:
end
```

In connection with the input of bdi data, please note that the data must be preceded by a bdi command that states how many vertices and how many polygon faces are to be used.

The last number in each vertex record is the surface number relative to the current surface. In the output listing, this is converted into an absolute surface number reference.

```
*BOUNDARY DRAWING DATA
 SRF 3:
 VX NBR         X            Y            Z        COORD SURF
    1       -1.000000    -1.000000       --            3
    2       -1.000000     1.000000       --            3
    …etc.
 PF NBR        VX1          VX2          VX3          VX4
    1           1            2            3            4
    2           5            6            7            8
    …etc.
```

Thermal mirror

To carry out thermal analysis of systems containing mirrors, you may need to use an extra dummy surface in contact with the mirror to accommodate TCE data. If you insert a REFLECT surface in a system, the TCE of the surface will be used for both the mirror radius and spacer. If the two are made from different materials, you need to use an extra surface.

Consider the following 100 mm focal length single-mirror system. If you enter the system as shown and list the refractive indices, you see that the TCE of the mirror is 0, so nothing will change when the temperature is changed.

```
*LENS DATA
Thermal Mirror Example
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --        1.0000e+20    1.0000e+14            AIR
 AST  -200.000000   -100.000000     50.000000 AS      REFLECT
 IMS        --            --        1.0000e-04 S

*PARAXIAL CONSTANTS
   Effective focal length: -100.000000    Lateral magnification:  -1.0000e-18

*CONDITIONS: GENERAL
   Temperature:              20.000000    Pressure:                  1.000000
```

Now suppose that you have a mirror made from BK7 glass (TCE = 7.1e-6), mounted in an aluminum tube (TCE = 23.6e-6). The proper way to set this up is the following:

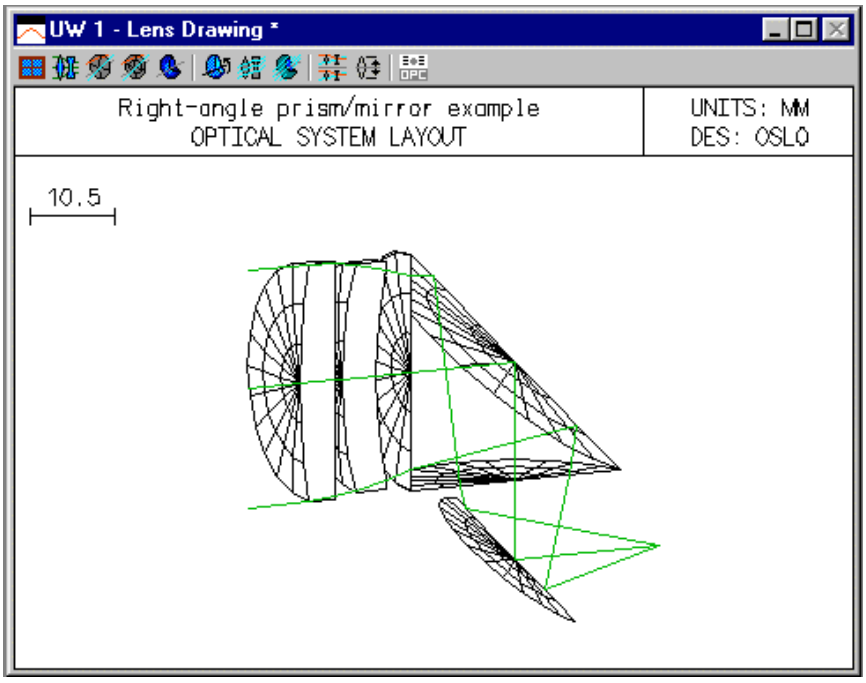
```
*LENS DATA
Thermal Mirror Example
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --        1.0000e+20    1.0000e+14            AIR
 AST  -200.000000        --          50.000000 AS      REFLECT
  2   -200.000000 P -100.000000     50.000000 S           AIR
 IMS        --            --        1.0000e-04 S

*PARAXIAL CONSTANTS
   Effective focal length: -100.000000    Lateral magnification:  -1.0000e-18

*CONDITIONS: GENERAL
   Temperature:              20.000000    Pressure:                  1.000000
```

Now if you change the temperature to 100, you will have the following system. Note that the dummy surface has shifted by .0002 microns, due to round-off error in the calculations.

```
*LENS DATA
Thermal Mirror Example
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --        1.0000e+20    1.0000e+14            AIR
 AST  -200.113600   -2.1149e-06     50.028400 AS      REFLECT
  2   -200.113600 P -100.176810     50.028399 S           AIR
 IMS        --            --         0.060106 S

*PARAXIAL CONSTANTS
   Effective focal length: -100.056800    Lateral magnification:  -1.0006e-18

*CONDITIONS: GENERAL
   Temperature:             100.000000    Pressure:                  1.000000
```

TIR prism/mirror system

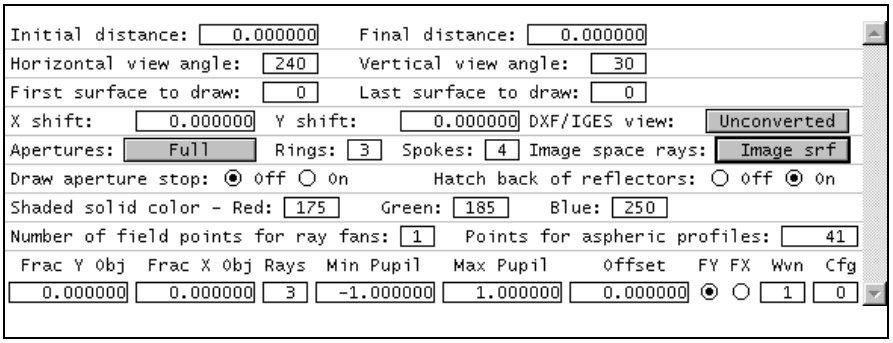
Many optical systems contain reflecting prisms and mirrors for redirecting beams or changing the orientation of images. Plane mirrors generally have no effect on image quality, and reflecting prisms are usually equivalent to blocks of glass. Nevertheless, for mechanical and tolerancing reasons it is often necessary to include the effects of tilted and decentered surfaces in the design data. This section gives an example of how to add a right-angle prism and turning mirror behind the lasrdblt.len lens. The example is primarily tutorial, not optical.



The prism will have an aperture of 10 (note that apertures in OSLO are specified by their radii, not diameters), and will be located 1 mm to the right of the last lens surface.

1) Open the lasrdblt.len file, change the lens identifier to “Right-angle prism/mirror example” and save it in your private directory under a new name.

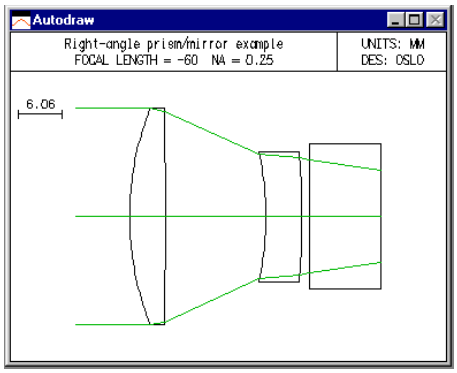
2) Before entering any surface data, click Lens >> Lens drawing conditions, and modify the spreadsheet so that it looks like the following. The items to be changed are Apertures, Image space rays and Number of field points for ray fans. These changes will make the Autodraw window more useful.



3. Close the Lens drawing operating conditions spreadsheet and return to the surface data spreadsheet. Drag the bottom of the window frame down until there is room for four more rows, then move the cursor to the row button for surface 5. Now press SHIFT+SPACE four times to insert the additional rows (this is a shortcut to using the mouse), or click row button 5 and click the Insert Before toolbar icon 4 times. Drag the bottom of the spreadsheet window down to see all the rows. Enter 1 for the thickness of surface 4.

4. Next you will insert the prism. Enter SF15 for the glass on surface 5. Click on the Special button for surface 6, select Surface Control >> General, and turn on TIR only. This means that rays will be reflected if they undergo total internal reflection and will fail otherwise. Note that for a TIR surface, the refractive index of the surface is the one for the medium into which the ray goes if it *fails* the TIR condition. The program knows that if the ray is totally internally reflected, the refractive index is the same as that in the incident medium.

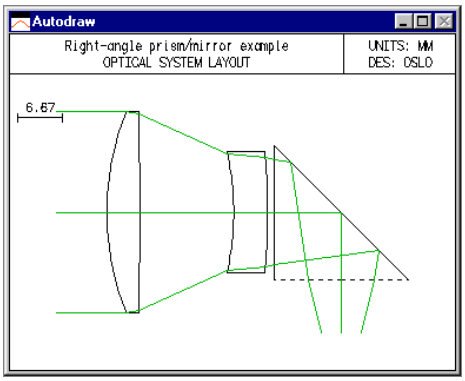
5. Set the apertures of surfaces 5 - 7 to 10. Set the thickness of surface 5 to 10. The Autodraw window should now look as follows:



5. Surface 6 will be the reflecting face of the prism, tilted at 45 degrees. Click the Special button and select the Coordinates item on the pop-up list. In the spreadsheet, enter 45 for TLA, and change the Tilt and bend button to Yes.

6. Enter -10 for the thickness of surface 6. Thicknesses are negative after an odd number of reflections, as discussed previously. Enter 10*sqrt(2) for the aperture.

7. The center of the turning mirror is to be 8mm below the prism. Set th 7 to -8.

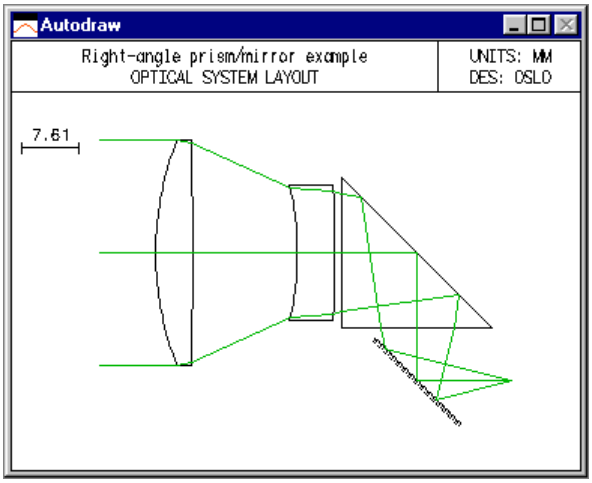


8. Click the Spe button in the text window. The following should appear in the Text window.

```
*TILT/DECENTER DATA
  6     DT    1         DCX       --        DCY       --        DCZ       --
        BEN             TLA    45.000000    TLB       --        TLC       --

*SURFACE_TAG DATA
  6     TIR   1
```

The next step will be to add a turning mirror that reflects the beam so that it continues from left to right.

9. Enter data for the turning mirror as shown below. Use an axial ray height solve to locate the final image surface, in the same manner as the laser doublet example. The mirror is entered using the Reflect (Hatch) item on the Glass Options list. This has no optical significance, but causes the back side of the mirror to be hatched on the drawing, as shown. Set the aperture to 8mm. Select Special, Coordinates to set the tilt to -45 degrees, and set the bend flag.



Gen | Setup | Wavelength | Field Points | Variables | Draw On | Surfs | Notes

Lens:Right-angle prism/mirror example Zoom 1 of 1 Efl 60.000049

Ent beam radius 15.000000 Field angle 5.7296e-05 Primary wavln 0.632800

SRF	RADIUS	THICKNESS	APERTURE RADIUS	GLASS	SPECIAL
OBJ	0.000000	1.0000e+20	1.0000e+14	AIR	
AST	41.040000	5.000000	15.000000 A	LASF35 C	
2	-542.675000	13.900000	15.000000	AIR	
3	-40.695000	5.000000	9.000000	LASF35 P	
4	-124.330000	1.000000	9.000000	AIR	
5	0.000000	10.000000	10.000000	SF15 C	
6	0.000000	-10.000000	14.142136	AIR	FC
7	0.000000	-7.000000	10.000000	AIR	
8	0.000000	12.599165 S	8.000000	REFL_HATCH	C
IMS	0.000000	0.000000	0.002000		

```
*TILT/DECENTER DATA
  6     DT    1          DCX        --        DCY        --        DCZ        --
        BEN              TLA    45.000000   TLB        --        TLC        --
  8     DT    1          DCX        --        DCY        --        DCZ        --
        BEN              TLA   -45.000000   TLB        --        TLC        --

*SURFACE TAG DATA
  6     TIR   1
```

Note that an axial ray height solve is used on surface 8, even though the system has tilted elements. In the case where only tilts of plane surfaces are used, and the bend flag is used to propagate the optical axis, paraxial optics remain valid.

Diffractive optics

Hybrid achromatic doublet

One attractive feature of using a diffractive surface is that it provides a way to add power to a system with only a marginal change in bulk. Also, since the manufacturing processes for diffractive surfaces do not restrict the phase function to just a power (r^2) term, it is possible to add a diffractive surface to a refractive singlet to perform both chromatic and monochromatic aberration correction. As a simple example, we will examine the design of a refractive/diffractive achromatic doublet. This type of element, combining both refractive and diffractive optics, is sometimes called a *hybrid* element.

We can use the thin lens achromatic doublet design equations for an initial design. Let ϕ denote the total power of the doublet and ϕ_{ref} and ϕ_{diff} denote the powers of the refractive and diffractive components, respectively. Thus, since this is a thin lens system, $\phi = \phi_{ref} + \phi_{diff}$. Also, let ν_{ref} and ν_{diff} be the Abbe values for the two components. Then, the component powers for the achromat are

$$\phi_{ref} = \frac{\nu_{ref}}{\nu_{ref} - \nu_{diff}}\phi \qquad (10.70)$$

$$\phi_{diff} = \frac{\nu_{diff}}{\nu_{diff} - \nu_{ref}}\phi$$

For refractive materials, ν_{ref} is positive, and, as we have seen, ν_{diff} is negative. Thus, we see from Eq. (10.70) that both components have the same sign of power as the total power of the doublet. This is in contrast to the familiar two-glass achromatic which consists of a positive crown glass element and a negative flint glass element. Since both components of the hybrid doublet are of the same sign of power, this generally results in weaker surface curvatures for the refractive lens. We also see from Eq. (10.70) that most of the power of the doublet is provided by the refractive component. This is not surprising, given that the diffractive lens is far more dispersive. For example, using the usual d, F, and C lines and BK7 glass ($\nu_{ref} = 64.2$, $\nu_{diff} = -3.45$), we see that approximately 95% of the total power of the doublet is provided by the refractive component.

Given the very strong wavelength dependence for a diffractive element, one very often finds that chromatic variation of *something* becomes the limiting aberration for a system containing diffractive optics. For this achromatic doublet, secondary spectrum may be a problem. The partial dispersion for a diffractive lens is given by

$$P_{diff} = \frac{\lambda_{short} - \lambda_0}{\lambda_{short} - \lambda_{long}} \qquad (10.71)$$

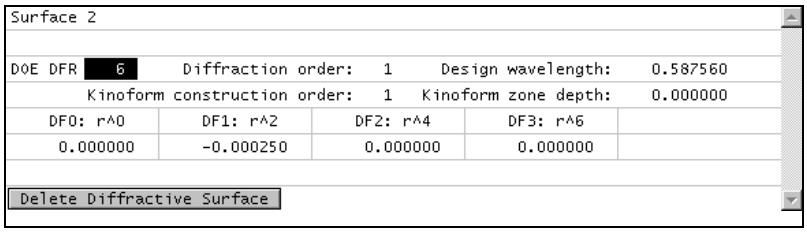
For the visible spectrum (d, F, C) this value is $P_{diff} = 0.596$. The longitudinal secondary color for the achromat is

$$\Delta l_{SS} = \left(\frac{-1}{\phi}\right)\frac{P_{ref} - P_{diff}}{\nu_{ref} - \nu_{diff}} \qquad (10.72)$$

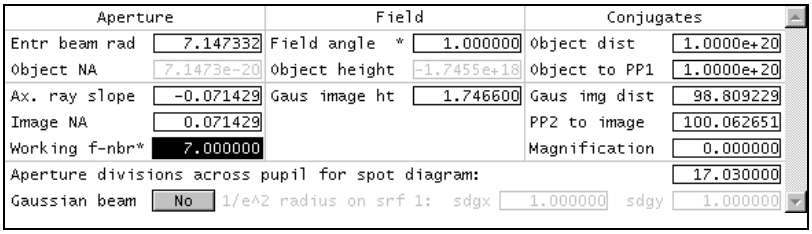
For a BK7/diffractive achromat, the secondary spectrum is about −0.0014 of the focal length; compare this to a two-glass achromat, which has a secondary spectrum of about 1/2200 = 0.00045 of the focal length. Another common problem in hybrid systems, particularly fast systems, is spherochromatism. Of course, if the first order properties of a diffractive lens are strong functions of wavelength, it should not be surprising that the higher order properties also have large chromatic variations.

As a design example, we will choose a 100 mm focal length, F/7, BK7/diffractive hybrid doublet. The field of view is ± 1°. At this speed and aperture, it seems reasonable to assume that the main aberrations will be of third and fifth-order, so we will use a sixth-order expansion for the diffractive surface. We start with a convex-plano lens with an approximate power split of 95% to 5% between the refractive and diffractive components.

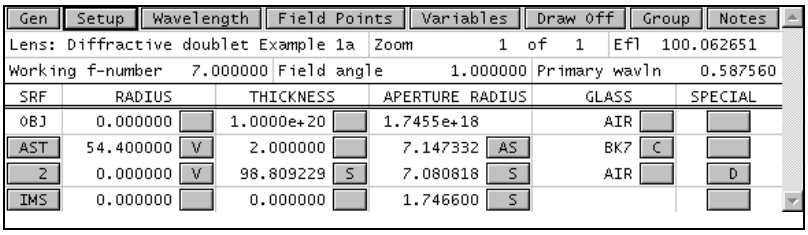
Open a new lens with 2 surfaces. On surface 1 set the radius to 54.4, the thickness to 2, and the glass to BK7. Put an axial ray height solve to 0 for the thickness of surface 2 (you can type "py 0" in the cell, or use theoptions button). Click the Special options button for surface 2. Select Diffractive Surface >> Symmetric CGH (even orders). In the spreadsheet, set the DOE DFR field to 6, and set the DF1 coefficient to -0.00025, as follows:



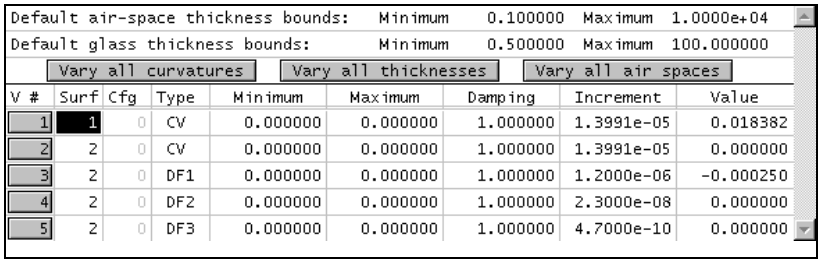
Close the Special spreadsheet and click the Setup button in the lens spreadsheet. Enter 7.0 for the working f-number and 1.0 for the field angle, as follows:



Close the Setup spreadsheet, and enter a Lens ID for the lens as shown below. Click the options buttons for the radius of surfaces 1 and 2 and make them variables. Save the lens as diffdblt1.len. The spreadsheet should look as follows.



Click the Variables button, and add DF1, DF2, and DF3 on surface 2 as variables. In order to enter subscripted variables like these, you must type the entries into the cells; you cannot click the menu items in the options list. The variables spreadhseet should look like the following.



On the Optimize menu, click Generate Error Function >> Ray operands. Accept the initial dialog box defaults (i.e. no vignetting). After closing the dialog, delete all the operands from the spreadsheet that pops up (use SHIFT+Drag with the row buttons), except for the following. Modify the first operand as shown.

OP	MODE	WGT	NAME	DEFINITION
1	Min	1.000000	EFL100	OCM10-100.0
2	Min	1.000000	AXIS_EDY	OCM16
3	Min	1.000000	AXIS_EOPD	OCM17
4	Min	1.000000	AXIS_EDMD	OCM18
5	Min	1.000000	OFAX_COMA	OCM31

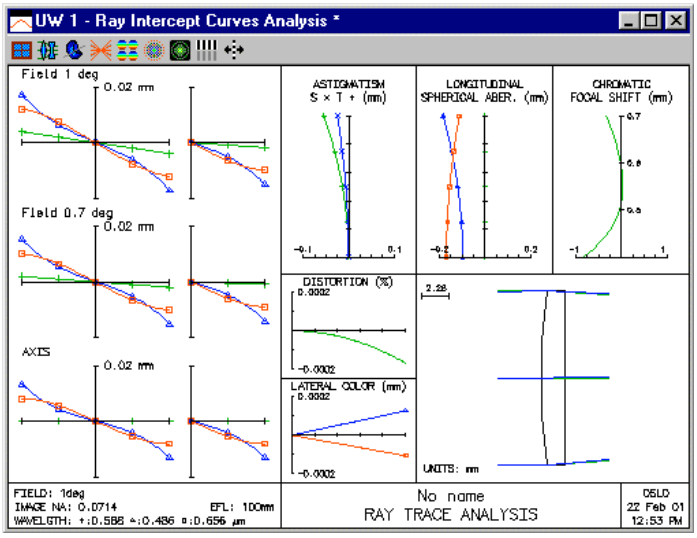
The names of the operands describe the meaning of the particular OCM operands set up for this error function. The actual function that evaluates them is called opcb_rays(), which can be found in the public CCL file optim_callbacks.ccl. Operands 2-4 are the DY, OPD, and Conrady DMD for the on-axis marginal ray. Operand 5 is a user-defined coma operand, defined (according to the CCL code) as shown below. You see that it forces the DY for the upper and lower rim rays from the edge of the field to be anti-symmetrical, meaning that the aberration must be astigmatic, as described in Chapter 5.

```
...
Ocm[23] = ssb(9,5);  // off-axis upper FY dy
…
Ocm[26] = ssb(9,8);  // off-axis lower FY dy
// insert user defs here
Ocm[31] = 0.5*(Ocm[23] + Ocm[26]); // tangential coma
...
```

After you hace completed the lens data entry, the variables entry, and the operands entry, save the lens (diffdblt1.len). Close the lens spreadsheet and re-open it to establish a base system you can return to if the optimization does not go the way you expect. The click the Ite button (twice) in the text window. After the first series, the error function should be in the 1e-9 range, and after the second it should be in the 1e-15 range.

```
*ITERATE FULL  10
NBR     DAMPING    MIN ERROR    CON ERROR  PERCENT CHG.
 0  1.0000e-08    0.032620       --
 1  1.0000e-08    0.001508       --        95.376152
…
10  1.0000e-08  2.3639e-09       --        78.468650

*ITERATE FULL  10
NBR     DAMPING    MIN ERROR    CON ERROR  PERCENT CHG.
…
 9  3.7921e-08  6.4692e-15       --        49.278539
 10  3.7921e-08  6.3801e-15       --          1.377601
```

Now click the ray analysis report graphics button in the current graphics window, which should produce a plot similar to the following.



The plot shows that the spherical aberration, coma, and axial color are under good control, with the dominant residual aberration being secondary spectrum, as expected from the above discussion. The correction produced by the DMD operand is typical. In order to study the chromatic correction in more detail, we will modify the error function to make a non-zero target for the DMD.operand. Since we don't know what target to use, it would be helpful to try several values, and we will set up a graphic slider to do this.

First, save the preliminary design as diffdblt2.len.

In command mode, operands are set up using the "o" command, for example the AXIS_EDMD operand that we want is entered by the command

```
O 4 "OCM18-<target>" "AXIS_EDMD" //cmd defn and name are in reversed order from ss
```

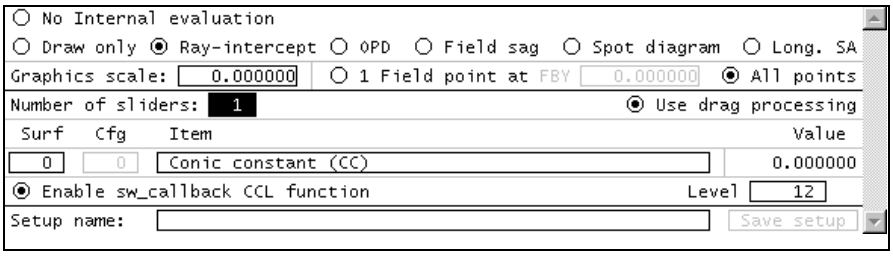
What we need to do is to attach the target value to a slider and execute this command every time the slider is moved. A straightforward way to do this is to write a CCL program, but this is not needed for this simple task. Instead, we can adapt the built-in slider-wheel spreadsheet to perform the task. The slider-wheel spreadsheet is set up to execute the CCL callback function sw_callback. There is a default function in the public CCL directory (in asyst_callbacks.ccl), but we need a customized version. It should be placed in the private CCL directory; commands placed there replace the public ones at run-time.

Using a text editor, create a file my_callbacks.ccl in your private CCL directory, and enter the following CCL code in the file:

```
/* My slider-wheel callback for diffdblt example */
cmd Sw_callback(int cblevel, int item, int srf)
{
        if (cblevel > 10)
        {
                sprintf(str1,"ocm18%+f \n",-cc[0]);
                o 4 str1;
                ite cblevel-10;
        }
        else if (cblevel > 0)
                ite cblevel;
}
```

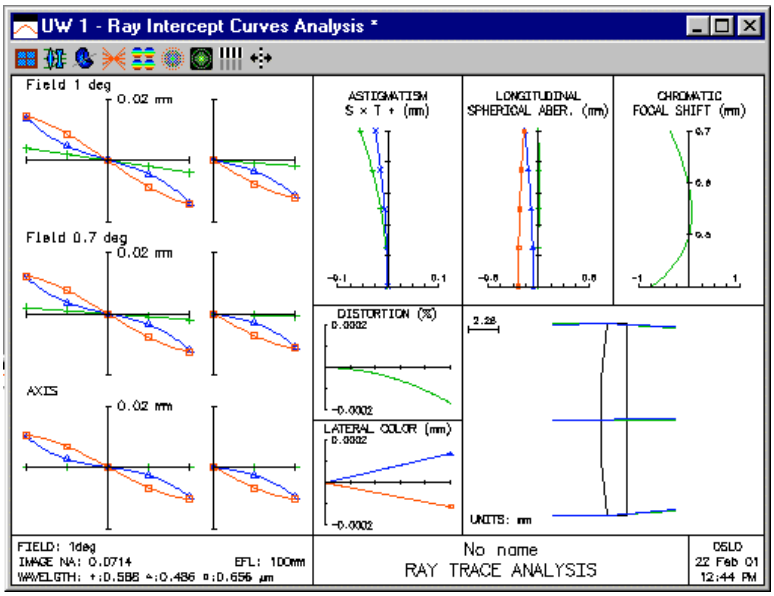
After you have entered the command, you must re-compile your private CCL. If you are using the built-in OSLO editor, it will be compiled automatically when you exit from the editor. Otherwise, you can use the button on the main toolbar to compile the command. Assuming that you get a "No error detected" compilation message, you are ready to use the command. We will discuss how it works later.

To use the slider-wheel to adjust the DMD target, open the sliderwheel setup spreadsheet and set it up as shown below. Note that the parameter being varied is the conic constant of the object surface. When the curvature of the object surface is zero, the conic constant has no effect on anything. Our sw_callback function makes use of this, and uses the conic constant of surface 0 to hold the target value for the DMD operand. By setting the callback level to 12, we trigger the path through the function that sets up the DMD operand, as you can see from looking at the above callback command.



After you close the spreadsheet, use the Tile Window command to arrange the window on your screen. If you click on one of the buttons at the end of the slider bar, you will see that changing the

target by .001 is too big a step, so use the step size buttons at the right-hand end of the slider to reduce the step to .0001. You can then experiment with using the mouse wheel, or dragging the slider, to see the effects. If you set the target (i.e. cc[0]) to -0.0002, then update the ray analysis window, you should see a plot like the following.



Here, we adjusted the target for DMD so that the ray-intecepts is red and blue are the same for the marginal ray. This was done mainly as an exercise; the original chromatic balance targeting DMD to 0 is actually slightly better. As a tool for studying parameters during optical design, sliders can be very helpful. For future reference, the two most important techniques used here are:

- Use a dummy parameter (i.e. one that has no optical effect on the system) to store non-optical data. This allows you to use the slider-wheel spreadsheet for a broadened class of applications.
- Use the CCL "sprintf" command to build commands at run-time, i.e. commands that are built dynamically as you drag a slider.

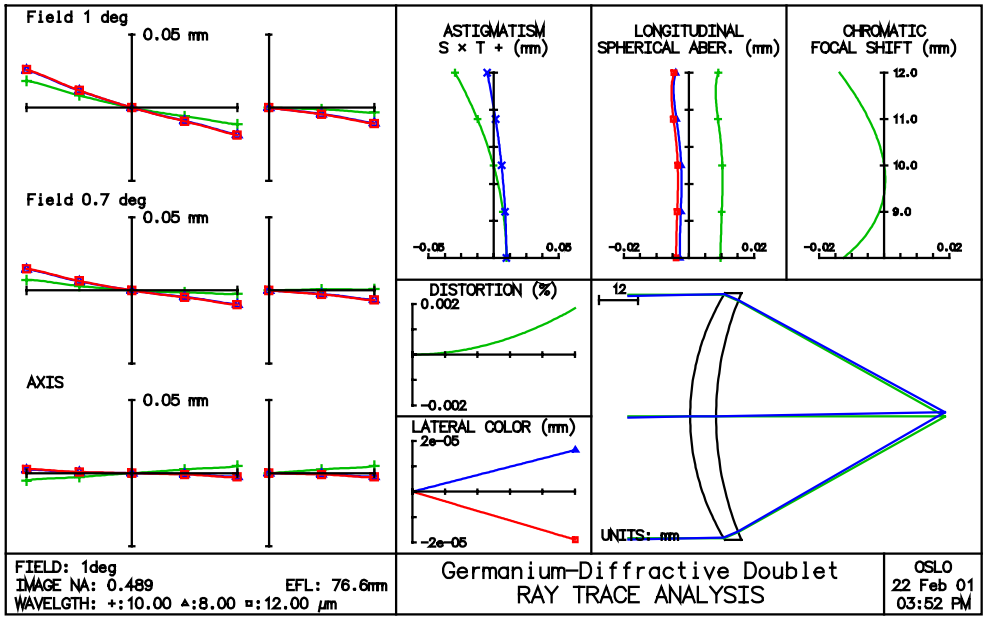
Infrared diffractive doublet

One of the first practical applications of diamond-turned computer-generated holograms was a single element achromat for the infrared. This lens, designed by Reidl and McCann, combines a conventional aspheric on the front surface with a diffractive surface on the back of a Germanium singlet to achieve good performance at f/1.

There are several reasons why diffractive optics may be more favorably applied to infrared systems. Most obviously, the longer wavelength means that the diffracting zones are wider and easier to fabricate. Many IR materials may be diamond turned, allowing for fabrication of both diffractive surfaces and aspheric refracting surfaces. By using an aspheric surface for aberration correction, the diffractive surface is only needed to correct chromatic aberration. Usually, this results in a very weak diffractive surface, with very low higher order aberration contributions. Also, the low diffractive power means that the λ/L ratios are small and high diffraction efficiency can be expected. Finally, the chromatic properties of some IR materials provide a good match to the diffractive surface. The paper by Reidl and McCann3 suggests the consideration of zinc selenide and zinc sulfide in the 3-5 μm band, and germanium and Amtir 3 in the 8-12 μm band.

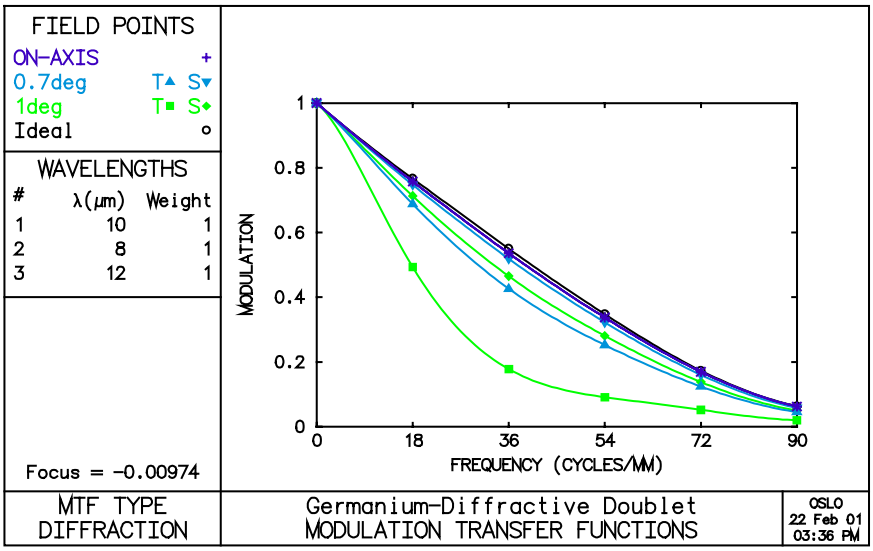
3 M. J. Reidl and J. T. McCann, “Analysis and performance limits of diamond turned diffractive lenses for the 3-5 and 8-12 micrometer regions,” Proc. SPIE Vol. CR38, 153-163 (1991).

Reidl and McCann give the following design for an F/1 germanium-diffractive doublet. The front surface of the lens is aspheric and the rear surface is spherical, with an r^2 diffractive phase profile. Note that the system, while achromatic, has secondary spectrum that cannot be corrected using this scheme.




```
*LENS DATA
Germanium-Diffractive Doublet
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0         --        1.0000e+20    1.7455e+18            AIR

  1         --            --         37.500000 AS          AIR

  2     72.560520      8.000000     38.000000            GERMA C *
  3     97.060800     70.304121 S   38.000000              AIR   *

  4         --         -0.009739      1.342401 S

*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF        CC          AD          AE          AF          AG
  2      -0.080670  8.4761e-10  2.6551e-13      --          --

*DIFFRACTIVE SURFACE DATA
  3   DOE DFR   2 - SYMMETRIC DIFFRACTIVE SRF       DOR   1 DWV     10.000000
                                                    KCO   1 KDP        --
      DF0         --      DF1   -2.2143e-05

*WAVELENGTHS
CURRENT  WV1/WW1     WV2/WW2     WV3/WW3
   1   10.000000    8.000000   12.000000
        1.000000    1.000000    1.000000

*REFRACTIVE INDICES
 SRF     GLASS           RN1         RN2         RN3         VNBR        TCE
  0      AIR           1.000000    1.000000    1.000000      --          --
  1      AIR           1.000000    1.000000    1.000000      --       236.000000
  2      GERMA         4.003227    4.005480    4.002024  869.131714      --
  3      AIR           1.000000    1.000000    1.000000      --       236.000000
  4      IMAGE SURFACE

*PARAXIAL CONSTANTS
   Effective focal length:   76.643439    Lateral magnification:  -7.6643e-19
   Numerical aperture:        0.489279    Gaussian image height:    1.337816
   Working F-number:          1.021913    Petzval radius:        -383.173311
   Lagrange invariant:       -0.654565
```

On-axis, the lens is essentially diffraction limited over the 8-12 µm wavelength region.

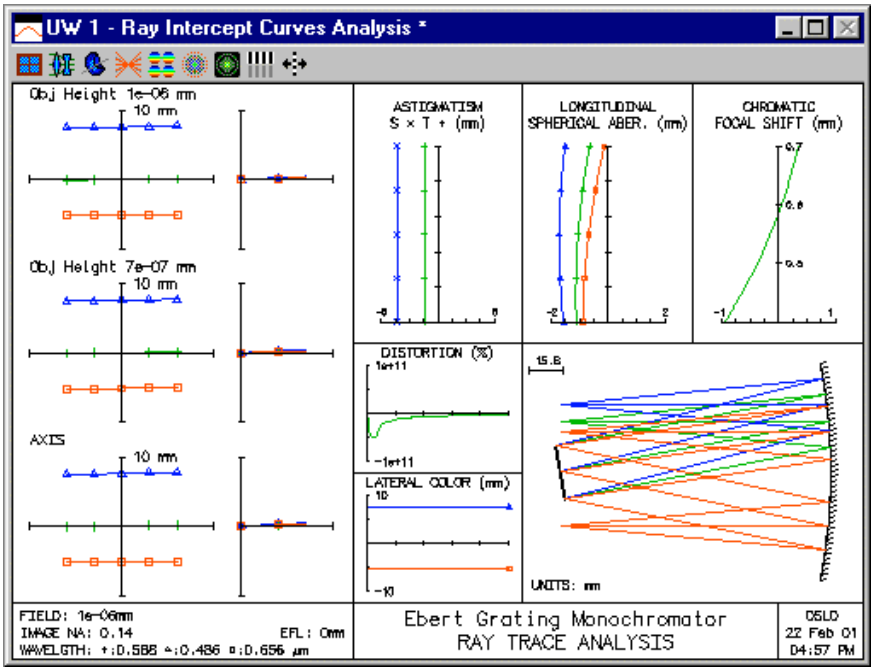


Because of the large difference in dispersions between germanium and the diffractive surface, almost all of the power of the doublet is contained in the refractive component. The diffractive surfaces has only three full diffracting zones.

```
*DIFFRACTIVE SURFACE ZONE RADII
 SURFACE 3        UNITS: mm
 PHASE INCREMENT PER ZONE =     1.000000 x 2 PI
 MINIMUM APERTURE RADIUS      --        MAXIMUM APERTURE RADIUS   38.000000
 ZONE NUMBER   ZONE RADIUS
       0           --
       1        21.251117
       2        30.053618
       3        36.808015
 MINIMUM ZONE WIDTH WITHIN LIMITS =    6.754397 (ZONE 3)
```

Simple grating monochromator

The Ebert monochromator, also known as the Fastie-Ebert monochromator, since it was developed for practical use by Fastie, is a very simple design that uses a plane grating at or near the focal point of a spherical mirror. The design included in OSLO puts the grating exactly at the focal point, which makes the system telecentric on both object and image sides. The focal points of the overall system are also at infinity, so the system is afocal. Notwithstanding this, the object and image are both at finite distance, and the system is set up as a focal system.



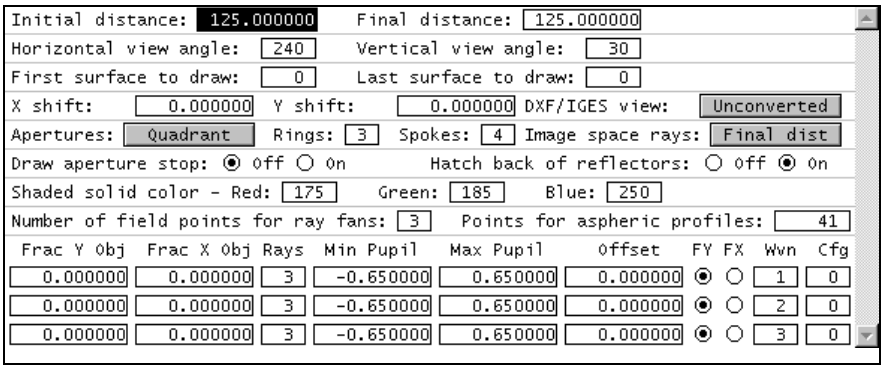
The grating is the aperture stop. This means that the chief ray enters the system parallel to the axis. To accommodate this, the **tele** general operating condition in OSLO must be turned on. An additional concern caused by the grating being the stop comes from the fact that it is square (the grating used is 25mm square with 600 grooves/mm). This means that the input numerical aperture must be large enough so that the grating is filled, which means that the reported paraxial data does not report the actual performance (see the discussion of apertures in chapter 4).

```
*LENS DATA
Ebert Grating Monochromator
 SRF      RADIUS      THICKNESS    APERTURE RADIUS       GLASS SPE   NOTE
  0         --        125.000000    1.0000e-06             AIR   *

  1   -250.000000   -125.000000     50.000000       REFL_HATCH

  2         --        125.000000 P   12.500000 AX    REFL_HATCH    *

  3   -250.000000 P -125.000000     50.000000 P        REFLECT

  4         --            --        25.000000
```

The system setup is somewhat unorthodox. The field of view is very narrow, and the object surface is decentered to move the entrance slit off the axis of the spherical mirror. The mirror is tilted using a return_coordinates (rco) command in which the coordinates are returned to the current surface. This has the effect of providing a local coordinate system in which the tilt of the current surface is removed prior to moving to the next surface.

```
*TILT/DECENTER DATA
  0     DT    1          DCX        --        DCY    -25.000000   DCZ        --
                         TLA        --        TLB        --       TLC        --
  2     RCO   2
        DT    1          DCX        --        DCY        --       DCZ        --
                         TLA    10.000000     TLB        --       TLC        --

*SURFACE TAG DATA
  2     GOR  -1           GSP      0.001667

*APERTURES
 SRF    TYPE APERTURE RADIUS
  0     SPC    1.0000e-06
  1     SPC     50.000000
  2     SPC     12.500000

     Special Aperture Group 0:
     A  ATP     Rectangle  AAC      Transmit  AAN        --
        AX1    -12.500000  AX2     12.500000  AY1    -12.500000  AY2     12.500000

  3     PKP     50.000000
  4     SPC     25.000000
```

Another interesting aspect of the Ebert monochromator setup is the default drawing rays, which must be set to fractional coordinates that account for the oversized pupil, and to wavelengths that show the grating dispersion.



f-θ scan lens

A laser line scan lens is essentially a monochromatic system and only needs to have good performance at a single wavelength, so it is a suitable candidate for diffractive optics. The lens should have a uniform spot size across the field and satisfy the *f-θ* condition so that it can translate a constant angular velocity into a constant linear velocity.

If we consider the design from the point of view of the third-order aberrations, we can identify three degrees of freedom:

i) the fourth order term (*DF2*) in the phase polynomial (the second-order term *DF1* is determined by the desired focal length),

ii) the position of the aperture stop, and

iii) the bending of the substrate for the diffractive lens.

We know that the Petzval curvature is equal to zero for a diffractive lens. If we can eliminate both coma and astigmatism, then we have eliminated all of the field-dependent aberrations that affect the spot size. There is a general solution to this problem, using the fourth-order term and stop position to set coma and astigmatism to zero. The remaining variable, the bending, can be used to control one of the remaining Seidel aberrations: spherical aberration or distortion. For the case of a scan lens, we need a prescribed, non-zero amount of distortion in order to achieve *f-θ* correction. (Recall that a distortion-free lens has an image height that is proportional to the tangent of the field angle.)

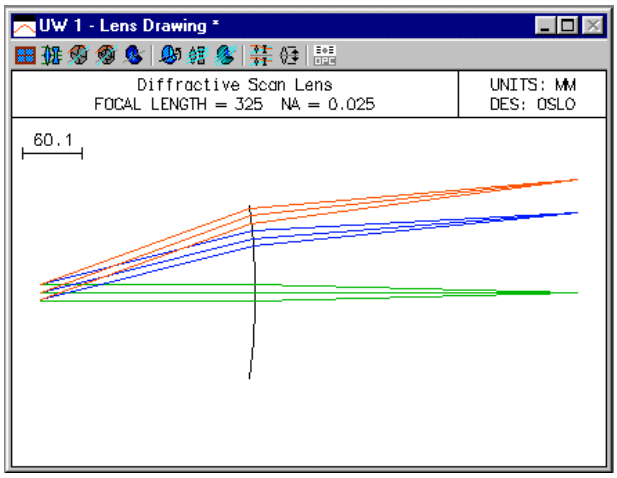
Scan lenses of this type have been considered by Buralli and Morris(4) The third-order solution shown in this example is a diffractive scan lens with a substrate radius of curvature that is equal to −2 times the focal length and an aperture stop (i.e., the scanner mechanism) located 2/3 of the focal length in front of the lens. The lens has the following parameters: focal length = 325 mm; F/20; λ_0 = 0.6328 μm; scan angle = ±20°. With these construction parameters, this lens is capable of scanning the width of an 8.5 by 11 inch piece of paper at a resolution of approximately 1600 dots per inch.

The lens data below illustrates the use of the Sweatt model for a diffractive lens. Note that the two radii of curvature differ slightly from the diffractive lens substrate radius of $-2f = -650$ mm. In the limit of $n \to \infty$, the Sweatt model is exact; since we must use a finite value for the index in OSLO (n = 10,001, in this case) these is a slight error in the resulting calculations using this lens. By converting the lens to a **DFR** surface (use the ***swet2dfr** SCP command), however, you can see that the error introduced by the finite value of n is negligible.

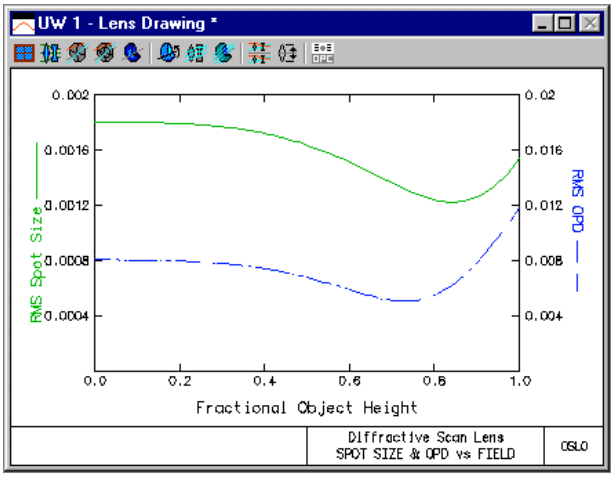
```
*LENS DATA
Diffractive Scan Lens
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE   NOTE
  0        --        1.0000e+20    3.6397e+19            AIR

  1        --        216.666667      8.125000 AS         AIR

  2   -650.065007        --         87.000000      SWEATT_MOD
  3   -649.935006    324.926455     87.000000            AIR

  4        --            --        118.283242 S

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.632800
        1.000000

*REFRACTIVE INDICES
 SRF     GLASS           RN1        TCE
  0      AIR           1.000000     --
  1      AIR           1.000000  236.000000
  2      SWEATT_MOD  1.0001e+04  236.000000
```

4 D. A. Buralli and G. M. Morris, “Design of diffractive singlets for monochromatic imaging,” Appl. Opt. **30**, 2151-2158 (1991).

```
  3      AIR           1.000000  236.000000
  4      IMAGE SURFACE

*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:      8.125000    Image axial ray slope:    -0.025000
   Object num. aperture:    8.1250e-20    F-number:                  20.000000
   Image num. aperture:       0.025000    Working F-number:          20.000000
 FIELD
   Field angle:              20.000000    Object height:          -3.6397e+19
   Gaussian image height:   118.290326    Chief ray ims height:    118.281403
 CONJUGATES
   Object distance:         1.0000e+20    Srf 1 to prin. pt. 1:    216.666667
   Gaussian image dist.:    325.000000    Srf 3 to prin. pt. 2:          --
   Overall lens length:     216.666667    Total track length:      1.0000e+20
   Paraxial magnification: -3.2500e-18    Srf 3 to image srf:      324.926455
 OTHER DATA
   Entrance pupil radius:     8.125000    Srf 1 to entrance pup.:        --
   Exit pupil radius:        24.375000    Srf 3 to exit pupil:    -650.000000
   Lagrange invariant:       -2.957258    Petzval radius:         -3.2503e+06
   Effective focal length:  325.000000
```



We can use the **Evaluate>>Spot Diagram>>Spot Size and OPD vs field** command from the menu to evaluate the RMS spot size and RMS OPD across the field.



Note the excellent state of correction of the field aberrations, even though this design was done using only third-order considerations. The spot size even gets slightly smaller with field angle, mainly due to a small amount of fifth-order Petzval curvature that is of opposite sign to the (uncorrected) third-order spherical aberration. One feature of using the Sweatt model is access to the aberration coefficients, since the lens appears as a “normal” refractive lens to the program.

```
*SEIDEL ABERRATIONS
 SRF       SA3          CMA3         AST3         PTZ3         DIS3
 SUM    -0.005714 -2.4643e-06  1.1956e-05 -5.3812e-05   -5.223535
```

```
*FIFTH-ORDER ABERRATIONS
 SRF      SA5         CMA5        AST5        PTZ5        DIS5        SA7
 SUM -4.6871e-06  1.7327e-05 -1.0561e-06    0.003965    0.403654 -4.6909e-09
```

The state of correction for the *f-θ* condition can be assessed using the ***ftheta** SCP command.

```
*F-THETA ANALYSIS
   FRACTIONAL  FIELD ANGLE      IMAGE     REFERENCE   SCAN ERROR  LOCAL ERROR
  FIELD ANGLE   (DEGREES)      HEIGHT    IMG HEIGHT   (PER CENT)   (PER CENT)
      --          --           --          --          --           --
    0.050000     1.000000     5.671892     5.671697     0.003443     0.003443
    0.100000     2.000000    11.343785    11.343394     0.003445     0.003447
    0.150000     3.000000    17.015677    17.015091     0.003446     0.003448
    0.200000     4.000000    22.687569    22.686788     0.003443     0.003435
    0.250000     5.000000    28.359458    28.358485     0.003433     0.003390
    0.300000     6.000000    34.031342    34.030182     0.003408     0.003287
    0.350000     7.000000    39.703214    39.701879     0.003364     0.003097
    0.400000     8.000000    45.375069    45.373576     0.003291     0.002785
    0.450000     9.000000    51.046897    51.045273     0.003182     0.002310
    0.500000    10.000000    56.718686    56.716970     0.003027     0.001626
    0.550000    11.000000    62.390422    62.388667     0.002814     0.000684
    0.600000    12.000000    68.062087    68.060364     0.002532    -0.000571
    0.650000    13.000000    73.733659    73.732061     0.002168    -0.002197
    0.700000    14.000000    79.405115    79.403758     0.001709    -0.004257
    0.750000    15.000000    85.076425    85.075455     0.001141    -0.006815
    0.800000    16.000000    90.747558    90.747152     0.000448    -0.009939
    0.850000    17.000000    96.418478    96.418849    -0.000384    -0.013704
    0.900000    18.000000   102.089144   102.090546    -0.001373    -0.018179
    0.950000    19.000000   107.759511   107.762243    -0.002534    -0.023443
    1.000000    20.000000   113.429531   113.433940    -0.003887    -0.029577
 CALIBRATED FOCAL LENGTH =  324.964300
```

Here again we see that the performance is quite good, just with a third-order design.

The Sweatt model can be converted to a phase model surface by using the ***swet2dfr** command and answering "y" (for "yes") when prompted for whether you wish to change the lens data.

```
*LENS DATA
Diffractive Scan Lens
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    3.6397e+19             AIR

  1        --        216.666667      8.125000 AS          AIR

  2   -650.000000    324.926455     87.000000             AIR   *

  3        --            --        118.283242 S

*DIFFRACTIVE SURFACE DATA
  2   DOE DFR  10 - SYMMETRIC DIFFRACTIVE SRF       DOR   1 DWV      0.632800
                                                    KCO   1 KDP        --
      DF0        --     DF1     -0.001538 DF2   -9.1033e-10 DF3   -1.0773e-15
      DF4   -1.5937e-21 DF5   -2.6404e-27
```

Whether this is a practical design or not is, perhaps, a debatable question, since the Show >> Auxiliary Data >> Diffractive Surf Zones command reveals that this lens has about 18,000 diffracting zones! This example does, however, show some of the interesting performance capabilities of diffractive lenses.

Diffractive eyepiece

As mentioned in an earlier section, diffractive optics can be used to decrease the size and weight of an optical system. One application of this ability has been in the area of eyepiece design, as demonstrated by Missig and Morris.5 In this paper, Missig and Morris present two hybrid refractive/diffractive eyepieces, both with three refractive components, and each with performance that compares favorably to a five element Erfle eyepiece. The prescription for the hybrid eyepiece with a single diffractive surfaces is given below.

```
*LENS DATA
Hybrid Eyepiece #2
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    5.7735e+19             AIR

  1        --         15.811850      4.000000 AS          AIR

  2    -54.098070      3.929290     12.800000             BK7 C
```

5 M. D. Missig and G. M. Morris, "Diffractive optics applied to eyepiece design," Appl. Opt. **34**, 2452-2461 (1995).

```
  3    -19.425860      0.100000     12.800000            AIR

  4    157.285720      4.036940     13.000000            BK7 C
  5    -43.015550      0.244010     13.000000            AIR

  6     34.971150      5.382580     13.000000            BK7 C
  7        --         15.871516     13.000000            AIR   *

  8        --            --         11.560626 S

*DIFFRACTIVE SURFACE DATA
  7   DOE DFR   4 - SYMMETRIC DIFFRACTIVE SRF       DOR   1 DWV      0.587560
                                                   KCO   1 KDP        --
      DF0        --     DF1     -0.001945 DF2    4.1213e-06

*WAVELENGTHS
CURRENT  WV1/WW1     WV2/WW2     WV3/WW3
   1    0.587560    0.486130    0.656270
        1.000000    1.000000    1.000000

*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:      4.000000    Image axial ray slope:    -0.199764
   Object num. aperture:    4.0000e-20    F-number:                  2.502949
   Image num. aperture:       0.199764    Working F-number:          2.502949
 FIELD
   Field angle:              30.000000    Object height:          -5.7735e+19
   Gaussian image height:    11.560626    Chief ray ims height:     11.560626
 CONJUGATES
   Object distance:         1.0000e+20    Srf 1 to prin. pt. 1:     20.819636
   Gaussian image dist.:     15.871516    Srf 7 to prin. pt. 2:     -4.152074
   Overall lens length:      29.504670    Total track length:      1.0000e+20
   Paraxial magnification: -2.0024e-19    Srf 7 to image srf:       15.871516
 OTHER DATA
   Entrance pupil radius:     4.000000    Srf 1 to entrance pup.:       --
   Exit pupil radius:       100.615275    Srf 7 to exit pupil:     519.541289
   Lagrange invariant:       -2.309401    Petzval radius:          -32.184283
   Effective focal length:   20.023591
 Note: This optical system contains special surface data.
       Calculations based on a paraxial raytrace may be invalid.
```

This design weighs only about one-third of the weight of the Erfle eyepiece and uses only BK7 glass. The diffractive surface has about 360 diffracting zones and a minimum zone width of approximately 25 μm. Thus, for the design wavelength, the maximum value of λ/L is 0.58756/25 = 0.023, so we would expect that scalar diffraction theory should be applicable to the analysis of this lens.

Using the scalar theory developed in chatper 6, with $\lambda_0 = 0.58756$ μm and $m = 1$, we find that the diffraction efficiencies for the d, F, and C wavelengths are 1.0, 0.865, and 0.964, respectively. A useful, one number figure of merit for diffractive lenses is the integrated efficiency η_{int}, which is the pupil averaged value of the local diffraction efficiency η_{local}.

$$\eta_{\text{int}} = \frac{1}{A_{pupil}} \iint_{pupil} \eta_{local}(x,y)\,dxdy \tag{10.73}$$

where A_{pupil} is the area of the exit pupil. It has been shown that the integrated efficiency may be used to scale the *MTF* in order to account for the presence of non-design diffraction orders when assessing image quality. For more details on the use of the integrated efficiency, see Buralli and Morris(6).

In OSLO, the local diffraction efficiency may be found by including the diffraction efficiency in the output of a single ray trace. The effects of integrated efficiency may be included by enabling the *Use diffraction efficiency* option in the General operating conditions. All efficiencies are computed using the extended scalar theory described in an earlier section. For example, we can compute the local efficiency along the chief rays at full field by establishing a full field object point and then tracing single rays in each of the three wavelengths using the tre 7 command (the efficiency of all the other surfaces is 1.0).

```
*TRACE REFERENCE RAY
```

6 D. A. Buralli and G. M. Morris, “Effects of diffraction efficiency on the modulation transfer function of diffractive lenses,” Appl. Opt. **31**, 4389-4396 (1992).

```
        FBY         FBX         FBZ
       1.000000      --          --
        FYRF        FXRF         FY          FX
         --          --          --          --
         YC          XC          YFS         XFS         OPL    REF SPH RAD
      10.839311      --        2.652196   -0.469393   52.080996 -169.938905

*TRACE EFFICIENCY RAY - LOCAL COORDINATES
 SRF        Y           X           Z           YANG        XANG      DIFF EFF

  7      11.609880      --          --       -2.779553      --        0.996364
PUPIL      FY          FX                                              OPD
            --          --                                              --

*TRACE EFFICIENCY RAY - LOCAL COORDINATES (WAVELENGTH 2)
 SRF        Y           X           Z           YANG        XANG      DIFF EFF

  7      11.570568      --          --       -2.864184      --        0.862362
PUPIL      FY          FX                                              OPD
            --          --                                              --

*TRACE EFFICIENCY RAY - LOCAL COORDINATES (WAVELENGTH 3)
 SRF        Y           X           Z           YANG        XANG      DIFF EFF

  7      11.627421      --          --       -2.785666      --        0.961072
PUPIL      FY          FX                                              OPD
            --          --                                              --
```

Because the wavelength-to-grating period ratio is small, the local efficiencies are very close to the scalar theory predictions given above. Note that we have not entered a value for the depth of the kinoform surface, so the optimum depth has been used by default. The scalar theory depth value for a lens with the refractive index of BK7 is 0.58756 μm/0.5168 = 1.14 μm. We can evaluate the effects of changing the kinoform blaze height by entering a value for the kinoform zone depth. For example, with a depth of 1.5 μm, the efficiency along the full field chief ray drops to 70%.

```
*DIFFRACTIVE SURFACE DATA
  7   DOE DFR   4 - SYMMETRIC DIFFRACTIVE SRF       DOR   1 DWV      0.587560
                                                    KCO   1 KDP      0.001500
      DF0        --     DF1     -0.001945 DF2    4.1213e-06

*TRACE REFERENCE RAY
        FBY         FBX         FBZ
       1.000000      --          --
        FYRF        FXRF         FY          FX
         --          --          --          --
         YC          XC          YFS         XFS         OPL    REF SPH RAD
      10.839311      --        2.652196   -0.469393   52.080996 -169.938905

*TRACE EFFICIENCY RAY - LOCAL COORDINATES
 SRF        Y           X           Z           YANG        XANG      DIFF EFF

  7      11.609880      --          --       -2.779553      --        0.703233
PUPIL      FY          FX                                              OPD
            --          --                                              --
```

Note that changing the value of the kinoform depth does not change the propagation direction characteristics of the diffracted rays; only the diffraction efficiency is affected.

The integrated efficiency can be determined from the percent weighted ray transmission in spot diagrams. For example, using the optimum depth, the on-axis integrated efficiencies are 99.5%, 87.0%, and 95.6%. Again, these values are close to the scalar theory values because of the low λ/L ratios. These efficiency values are used to scale the *MTF*, if calculated.

```
*SPOT DIAGRAM: MONOCHROMATIC
 APDIV    50.000000
 WAVELENGTH 1
 WAV WEIGHTS:
       WW1         WW2         WW3
    1.000000    1.000000    1.000000
 NUMBER OF RAYS TRACED:
       WV1         WV2         WV3
      1976           0           0
 PER CENT WEIGHTED RAY TRANSMISSION:    99.516699

*SPOT DIAGRAM: MONOCHROMATIC
 APDIV    50.000000
 WAVELENGTH 2
 WAV WEIGHTS:
       WW1         WW2         WW3
```

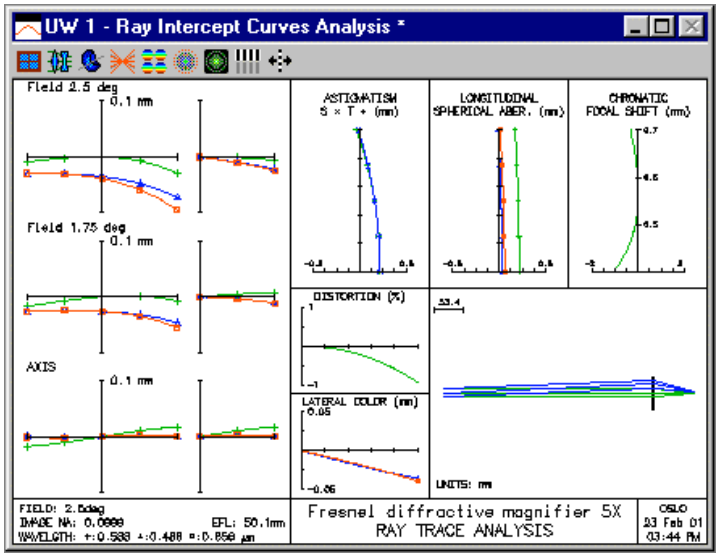
```
    1.000000    1.000000    1.000000
 NUMBER OF RAYS TRACED:
       WV1         WV2         WV3
         0        1976           0
 PER CENT WEIGHTED RAY TRANSMISSION:    87.037691

*SPOT DIAGRAM: MONOCHROMATIC
 APDIV    50.000000
 WAVELENGTH 3
 WAV WEIGHTS:
       WW1         WW2         WW3
    1.000000    1.000000    1.000000
 NUMBER OF RAYS TRACED:
       WV1         WV2         WV3
         0           0        1976
 PER CENT WEIGHTED RAY TRANSMISSION:    95.634114
```

Fresnel diffractive magnifier

The Fresnel diffractive magnifier is a plastic element with an aspheric Fresnel surface on the front and a diffractive surface on the back. It is an interesting system for both its tutorial value and its optical performance. The optical performance is much better than that of a simple lens, as shown by the ray analysis below.

A Fresnel surface in OSLO is one that has the power of a curved surface, but is actually placed on a substrate having a different curvature (usually flat). This is accomplished by dividing the surface into prismatic zones, so that the surface normal approximates that of the curved refracting surface at any point. As the number of zones increases, the approximation becomes better. Fresnel surfaces were originally used in searchlights, where a substantial reduction in weight was accomplished. More recently, Fresnel surfaces have been used in a variety of consumer optics products made from embossed plastic.

A diffractive surface is outwardly similar to a Fresnel surface, but there is an important difference between the two: With a Fresnel surface, light from different zones combines incoherently, while with a diffractive surface, light from different zones combines coherently. The former follows the laws of refraction, while the latter follows the laws of diffraction. Curiously, the chromatic aberration of the one has the opposite sign of the other, which is used in the magfrenl.len magnifier. The lens data is shown below.




```
*LENS DATA
Fresnel diffractive magnifier 5X
 SRF      RADIUS      THICKNESS    APERTURE RADIUS       GLASS SPE   NOTE
  0         --        1.0000e+20    4.3661e+18             AIR

  1         --        250.000000      5.000000 A           AIR

  2     32.626000       2.000000     20.000000            CARBO C *
  3         --         48.700000     20.000000              AIR   *

  4         --             --          5.000000

*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF         CC           AD           AE           AF           AG
  2      -0.969340 -1.5465e-07        --           --           --

*DIFFRACTIVE SURFACE DATA
  3   DOE DFR   2 - SYMMETRIC DIFFRACTIVE SRF       DOR   1 DWV      0.587560
      DF0         --      DF1      -0.001041

*SURFACE TAG DATA
  2     FRN   1   FCV        --        FCC        --
```

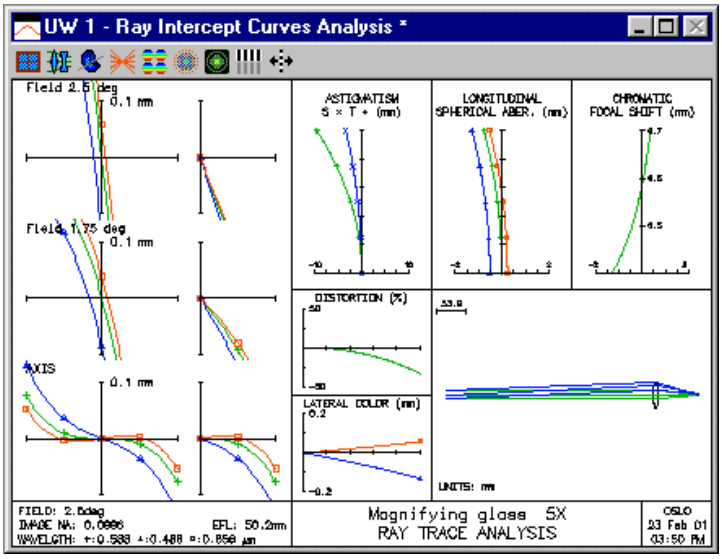
Note that surface 2, the Fresnel surface, is made aspheric to correct higher order aberrations. Surface 3, the diffractive surface, contains only a quadratic phase term to add some focusing

power and correct the chromatic aberration. The ray analysis on the previous page shows that the system is well-corrected for primary but not secondary color.

Conventional 5x magnifier

This lens is an ordinary 50mm focal length singlet, 31mm diameter, set up as it would be for use as a visual magnifier. The eye relief has been assumed to be 250mm, and the pupil diameter has been taken as 10mm to provide freedom of eye movement.

This lens has been included in the demo library mainly for comparison to the Fresnel diffractive magnifier (magfrenl.len). A great deal of caution needs to be exercised in interpreting the ray analysis for a visual instrument such as a magnifier, because the eye is neither in a fixed position, nor at a fixed focus. The eye (particularly a single eye) can tolerate rather poor imagery, but tends to be intolerant of imagery that changes rapidly with eye position.



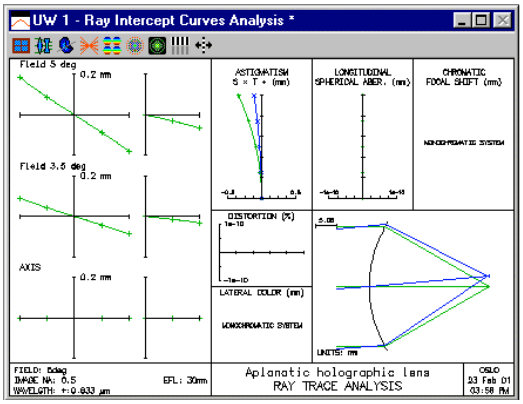
Optical holographic element

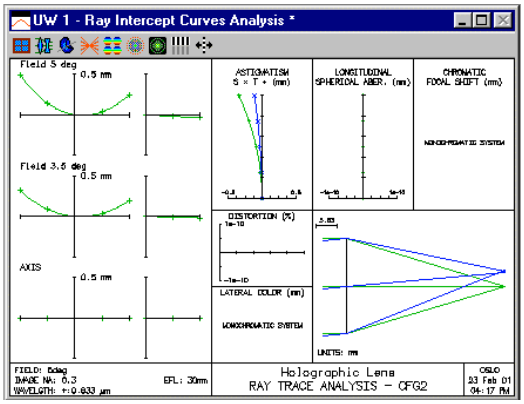
This is a hologram made from two point sources. An example holographic lens may be found in the public lens file "len\demo\lt\hologram.len." This is an on-axis holographic optical element, recorded with a helium-neon laser (0.6328 μm), with a focal length of 30 mm. The prescription for the HOE is shown below.

```
*HOLOGRAPHIC OPTICAL ELEMENT DATA
  2     HOR  -1          HWV      0.632800
        HV1   0          HX1        --        HY1        --        HZ1   -1.0000e+20
        HV2   1          HX2        --        HY2        --        HZ2     30.000000
```

The locations of the two point sources used to record the holograms are specified by their *x*, *y*, and *z* coordinates in the local coordinate system of the HOE surface. This is an on-axis lens, so the *x* and *y* coordinates of both point source 1 (HX1, HY1) and point source 2 (HX2, HY2) are zero. Point source 1 is actually a plane wave, incident upon the hologram from the left (the negative *z* direction), so HZ1 is set to -1.0×10^{20}. Since this is a real wave (the light travels from the source to the HOE surface), source 1 is real. This status is indicated by the "virtual factor" HV1; the virtual factor is true (1) for a virtual source and false (0) for a real source. Since point source 1 is real, HV1 = 0. Point source 2 is located at the focal point, which is 30 mm to the right of the HOE surface, so HZ2 is equal to 30. This wave is recorded such that the light is propagating from the HOE to the source, so this is a virtual source and HV2 = 1. As mentioned above, a helium-neon laser was used to record the hologram, so the construction wavelength (HWV) is 0.6328. The construction wavelength is always specified in micrometers, just like all wavelengths in OSLO. The desired reconstruction diffraction order (HOR) is –1; in this order, an on-axis plane wave reconstruction beam will be diffracted into a spherical wave converging to the focal point. This follows from elementary holography theory: if one of the construction beams is used to reilluminate the hologram, the other construction beam will be produced by the diffraction.

This lens has two configurations; in the first configuration, the hologram substrate has a radius of curvature equal to the focal length while in the second configuration, the substrate is planar. For both configurations, the on-axis image is perfect, since the reconstruction geometry is the same as the recording geometry. The off-axis performance, however, is quite different. Curving the hologram substrate is analogous to the "bending" of a thin lens. Curving the HOE around the focal point eliminates the coma and results in an aplanatic element (i.e., the Abbe sine condition is satisfied), as shown by Welford.7 The elimination of coma is quite evident from the ray intercept curves.





7 W. T. Welford, "Aplanatic hologram lenses on spherical surfaces," Opt. Commun. **9**, 268-269 (1973).

The Rowland circle mount

An important advance in the development of grating spectrographs was the concave grating, introduced by Henry A. Rowland in the early 1880's. The grating lines are ruled on a concave reflecting surface, so that this single surface both separates the light of different wavelengths (by diffraction) and focuses the resulting spectra. The grating lines are equally spaced along a chord of the surface and are, thus, unequally spaced on the reflecting surface itself.

Rowland found that if the source (entrance slit) is placed on a circle whose diameter is equal to the radius of curvature of the grating surface, the resulting diffracted spectra are focused on this same circle, provided that the surface of the grating is tangent to this circle and the grating lines are orthogonal to the plane of the circle. This circle is called the *Rowland circle*. For this example, we use a grating with 600 lines/mm, ruled on a surface with a radius of curvature of 100 mm. The radius of the Rowland circle is half the radius of the grating surface, or 50 mm.

The performance of a concave grating with object and image on the Rowland circle is illustrated in this example. You can set up the system by entering the following data

Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

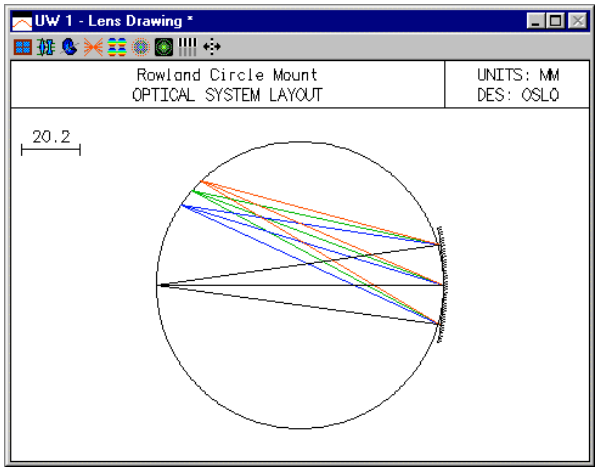
Lens: Rowland Circle Mount | Zoom 1 of 1 | Efl -50.000000

Ent beam radius 20.000000 | Image height 1.0000e-06 | Primary wavln 0.587560

SRF	RADIUS		THICKNESS		APERTURE RADIUS		GLASS		SPECIAL
OBJ	50.000000		100.000000		1.0000e-06		AIR		F
1	-50.000000	P	0.000000		50.000000		AIR		F
AST	-100.000000		-100.000000	P	20.000000	A	REFL_HATCH		D
IMS	50.000000	P	0.000000		50.000000				F

Note that the object and image surfaces both have a radius of curvature equal to the Rowland circle radius. The special data is set to the following values. For the DRW ON data, use the SPECIAL>>Surface Control>>General spreadsheet, and for the diffractive surface, use the SPECIAL>>Diffractive Surface>>Linear Grating spreadsheet.

```
*SURFACE TAG DATA
  0      DRW ON
  1      DRW ON
  2      GOR   1          GSP      0.001667    GBO    0          GDP       --
  3      DRW ON
```

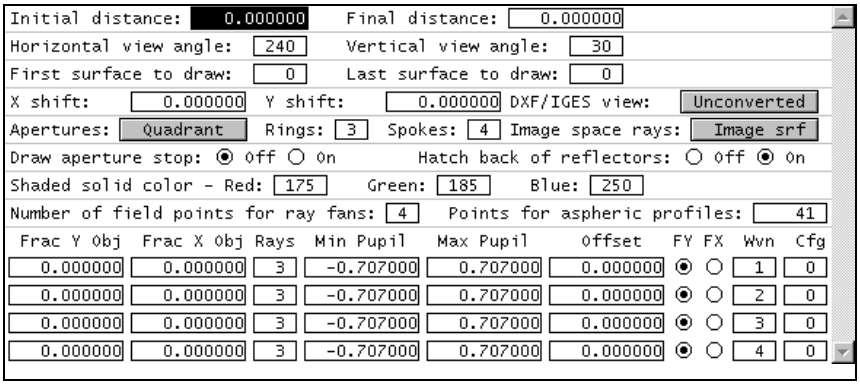


If you are viewing this example in color, you see that the drawing shows black rays incident on the grating, and colored rays reflecting to the image surface. To make this drawing of the system, you need to modify the system wavelengths and set special Lens Drawing conditions. First, add a fourth wavelength, 2 microns. This wavelength is long enough so the ray trace will fail at the grating surface.

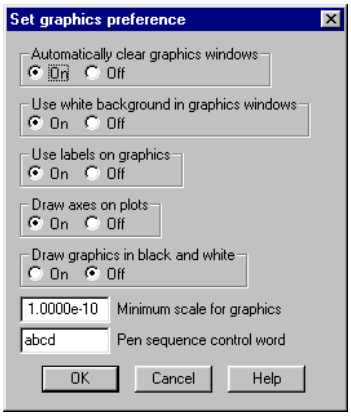
```
*WAVELENGTHS
CURRENT   WV1/WW1      WV2/WW2      WV3/WW3      WV4/WW4
```

```
1     0.587560     0.486130     0.656270     2.000000
      1.000000     1.000000     1.000000     1.000000
```

Next, set the Lens>>Lens Drawing conditions. Note that all 4 field points are from the axis, but have different wavelengths.



Finally, you need to reset a preference. Use File>>Preferences>>Preference Groups>>Graphics, which will show the following dialog.



You need to set the Pen sequence control word as shown (normally abcdefgh). The Pen sequence control word sets the order of pens used for drawings. In drawing ray trajectories, when a new field point is encountered, OSLO uses the next pen from the sequence. When it reaches the end, it starts over again. The first four colors are (normally) black, green, blue, red. In the present system, the lens is drawn in black, the first field point in green, the second in blue, the third in red, and the fourth in black. By setting the fourth wavelength to 2 microns, we create a situation where the ray fails at the grating. Therefore the black rays are only shown up to the grating surface.

This example illustrates a trick that can be used to produce a special drawing. Fundamentally, OSLO is a numerical optimization program, not a drawing program. Drawings are provided mainly to give the designer some idea of what system he or she is working with. However, occasionally it may be necessary to produce a non-standard drawing for a special purpose, and tricks such as the one used here can be helpful.

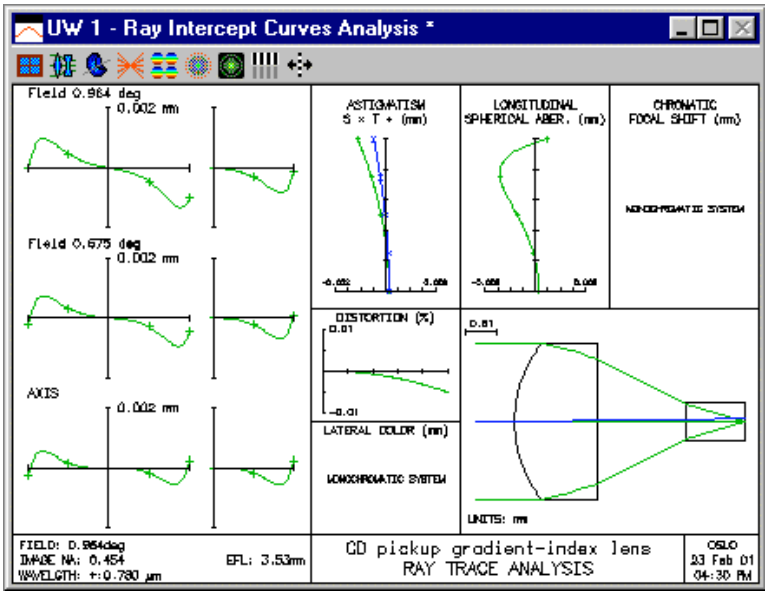
Gradient index

Gradient-lens for a CD pickup

Gradient-index technology has been applied to several diverse application areas. One of the common ones is in consumer optics, where small gradient lenses or arrays of gradient lenses are used in CD-ROMS and copiers. The lens shown here was designed by Nishi and Tayuma at Nippon Sheet Glass (NSG), manufacturer of gradient lenses known by the trademark Selfoc.

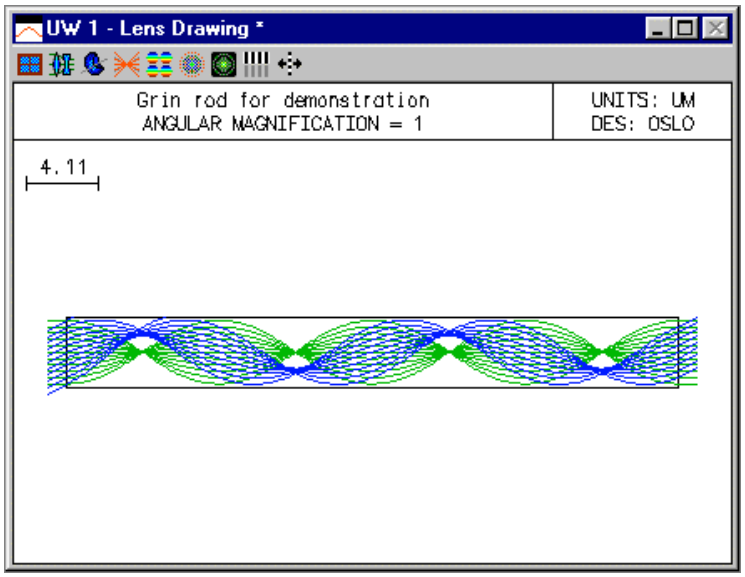
In connection with Selfoc lenses, it is worth noting that although NSG makes the paraxial data for their lenses available to the public, the actual data describing the real index distributions has not been published. The data for the index distributions used in this lens do not necessarily coincide with that for commercial Selfoc lenses. The lens is included here to show an example of how it can be entered into OSLO, as well as to assess the general performance level that might be expected from a GRIN lens of this type.

In the drawing shown below, the block on the right is an optical disc. In analyzing this system, please recall that since the standard paraxial trace does not handle gradient index materials, you must use the ***pxc** and ***pxt** commands to obtain paraxial data.

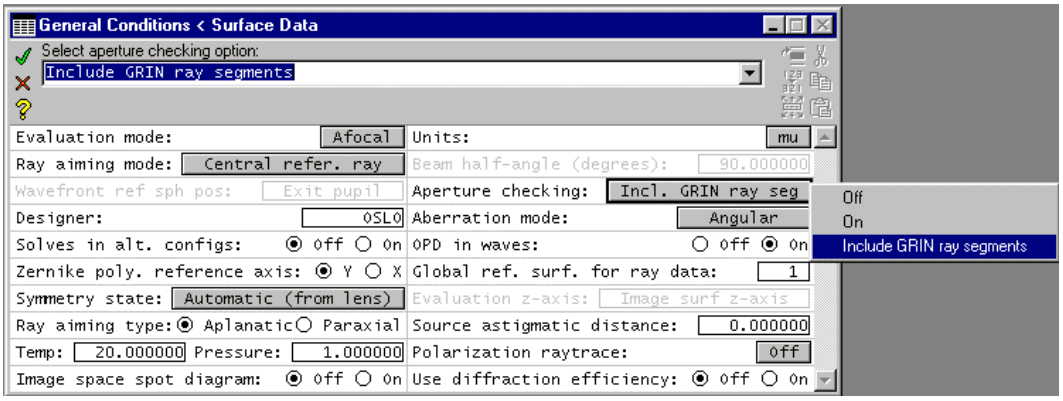


Gradient index rod

As noted before, gradient index technology is used in many diverse applications. The system in this example is a gradient index rod that is more like a fiber than a lens. It is, however, an imaginary system, designed using OSLO without regard for actual refractive indices that can be manufactured.



The drawing above shows both on-axis and off axis beams. Note that the off-axis beam for both the upper and lower rim rays is truncated by the edge of the rod. In order for this to happen, you must turn on the Aperture check all GRIN segs general operating condition. If you don't, the rays will be free to propagate outside the boundary defined by the aperture on the surface for with the GRIN material is defined.



Lasers and Gaussian beams

Basic Gaussian beam imaging

In this example, we will consider the propagation of a Gaussian beam in a simple imaging system. In Chapter 1, it was shown that the *ABCD* matrix relating object and image planes can be written as

$$\mathbf{M} = \begin{bmatrix} m & 0 \\ -1/f & 1/m \end{bmatrix} \tag{10.74}$$

where m is the transverse magnification and f is the focal length of the imaging lens. Using the *ABCD* law and assuming $n = n' = 1$, we can propagate a Gaussian beam described by q in the object plane to a beam described by q' in the image plane via

$$q' = \frac{mq}{\frac{-1}{f}q + \frac{1}{m}} \tag{10.75}$$

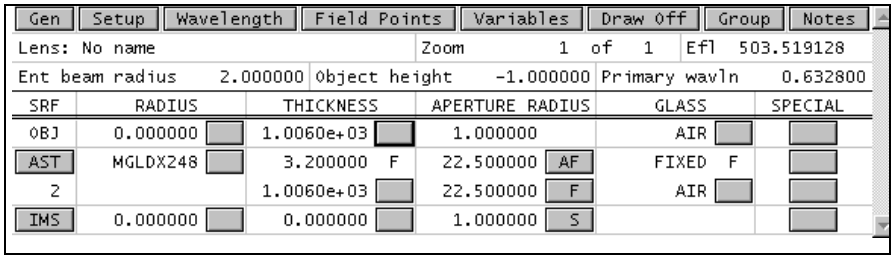
Using the definition of the q parameter, it is easy to separate Eq. (10.75) into its real and imaginary parts and find expressions for the spot size w' and wavefront radius of curvature R' in the paraxial image plane

$$w' = |m| w \tag{10.76}$$

$$R' = \frac{m^2 R f}{f - mR} \tag{10.77}$$

Several interesting conclusions can be drawn from the above relations. Not surprisingly, the ratio of the spot sizes is just the paraxial magnification. Perhaps less obvious is an implication of the image radius of curvature equation. Consider the case where we place the input beam waist in the object plane, so $R = \infty$. Taking the limit of Eq. (10.77) for this case, we find that $R' = -mf$. For the usual case of a positive lens with real object and image distances, f is positive and m is negative. Thus, R' is seen to be positive, which in the beam sign convention means that the image space beam has already passed through its waist before intersecting the paraxial image plane, i.e., the beam waist is inside the paraxial image location. This phenomenon is sometimes called the *focal shift*, since the point of maximum axial irradiance is not at the geometrical focal point. In order to have a beam waist in the paraxial image plane ($R' = \infty$), we must have a radius $R = f/m$ in the object plane.

The focal shift phenomenon is more dramatic for "slow" beams with a small divergence angle, or in other words, beams with a small Fresnel number. (The Fresnel number for a circular aperture of radius a and wavefront radius of curvature R is given by $a^2/\lambda R$.) We can illustrate this using the interactive ABCD analysis spreadsheet in OSLO. We will select a lens from the catalog database with a focal length of about 500 mm and use the paraxial setup spreadsheet to set the paraxial magnification to −1. Be sure to change the primary wavelength to 0.6328 μm and delete wavelengths 2 and 3 before setting the magnification. Using, for example, the Melles Griot lens MGLDX248, the lens prescription is



Using the interactive ABCD analysis spreadsheet, we will examine the propagation of a Gaussian beam through this lens. Use a waist size of 0.25 mm and a waist distance of 0 to place the incident beam waist at surface 0.

There are a few conventions to remember in using the Gaussian beam spreadsheet in OSLO.

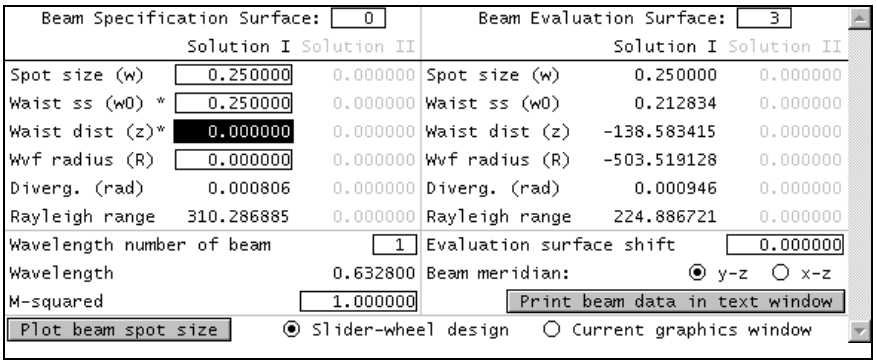
- To use the spreadsheet, you must enter data in two of the four fields (w, w0, z, R) on the specification surface. The remaining 2 fields will be calculated automatically. The data entry field will be indicated by an asterisk (*) once you enter a value. It is not possible to enter impossible input data; the program will display an error message.

- Sign conventions. The waist position is entered relative to the specification surface. If the waist is to the left of the specification surface, it has a negative sign (if there are no mirrors). The sign convention for wavefront radius is the same as for surface radius of curvature. That is, considering a wavefront diverging to the right, the wavefront radius of curvature is negative. In most laser literature, such a divergent wavefront has a positive radius of curvature.

- OSLO uses a convention that source distances less than 1e8 are considered finite, while source distances more than 1e8 are considered infinite. In the case of Gaussian beam propagation, infinite distance cannot be handled, so OSLO uses a convention that when the object distance is greater than 1e8, the beam waist is considered to be on surface 1. This makes it easier to compare Gaussian beam propagation with ordinary geometrical propagation, because when the object distance is infinite, the wavefront on surface 1 is plane for either case.

- The OSLO Gaussian beam spreadsheet compares the beam given on a specification surface to the beam on an evaluation surface. The default specification surface is the object surface, and the default evaluation surface is the image surface. However, there is no requirement that the specification surface be in object space, or even that the evaluation surface have a higher surface number than the specification surface. It is possible to make the specification surface an interior surface, and find the solution in either object of image space by just changing the evaluation surface number.

In the present example, the object distance is finite, so the waist is on the object surface (surface 0).



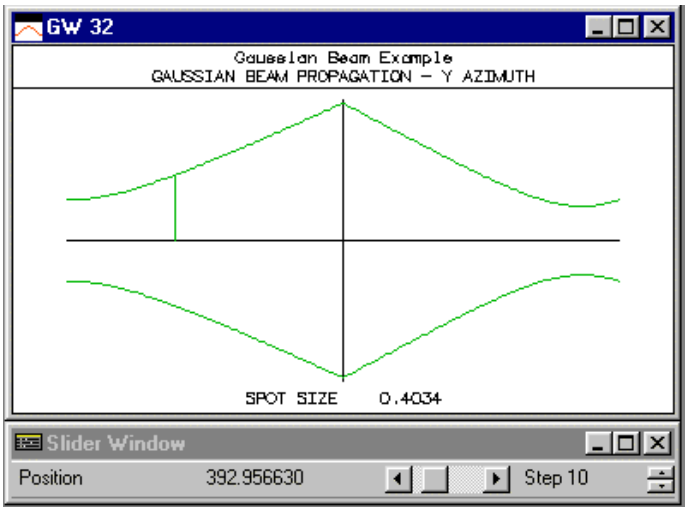
Now click the Print beam data in text window button to see a complete analysis on all surfaces.

```
*GAUSSIAN BEAM - YZ PLANE
 WAVELENGTH =   0.632800      M-SQUARED =   1.000000
 SRF    SPOT SIZE  DIVERGENCE  WAIST SIZE  WAIST DIST  INC RADIUS  RFR RADIUS RAYLEIGH
RG
  0     0.250000    0.000806    0.250000       --          --          --
310.286885

  1     0.848204    0.000164    0.810525  1.5242e+03 -1.1017e+03  1.7545e+04
4.9415e+03
  2     0.848050    0.000946    0.212834  867.397688  1.7575e+04  925.703167
224.886721
```

```
  3     0.250000    0.000946    0.212834 -138.583415 -503.519128 -503.519128
224.886721
```

The spreadsheet and surface-by-surface analysis confirm the above discussion: the spot size in the paraxial image plane is the same as the spot size in the object plane (since $|m| = 1$) and the output beam waist lies to the left of the paraxial image plane, as can be seen in the schematic beam spot size plot. Also note that, as expected, the output wavefront radius of curvature is equal to the focal length.

If you click the *Plot beam spot size* button, you will produce a graphical depiction of the beam propagation through the system. This is an anamorphic drawing in which the scale in the *y*-direction is greatly expanded so you can see the changing spot size. If you select the slider-wheel design option, a graphics slider will be created that lets you drag a cursor along the z-axis and displays the current spot size in the graphics window, as shown below.



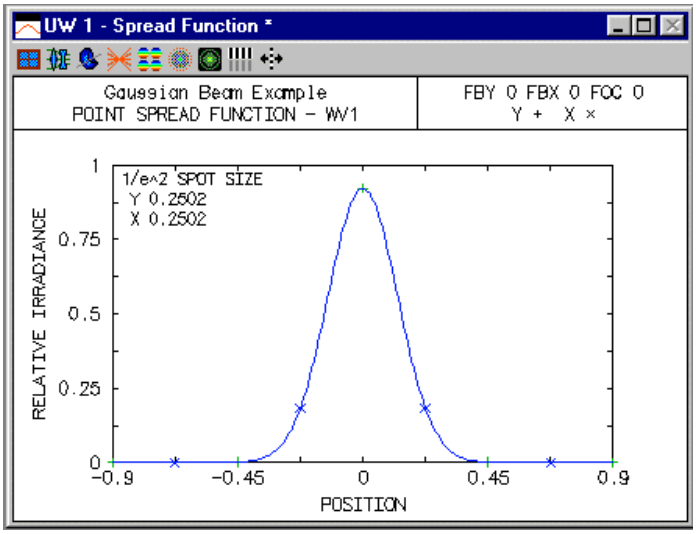
Point Spread Function calculation

We can use the point spread function computation in OSLO to verify the *ABCD* analysis. From the Gaussian beam data, we see that the radius of curvature of the wavefront at surface 1 is -1.1017×10^3 mm and the spot size at surface 1 is 0.848204 mm. To set up an equivalent PSF calculation, we change the object distance (the thickness of surface 0) to match this wavefront radius of curvature value; then the geometric wavefront will have the same radius of curvature at the lens as the Gaussian beam we have just traced. Also, we set the setup operating conditions to use a Gaussian beam with an entering spot size equal to the Gaussian beam spot size at surface 1. We also increase the number of aperture divisions to 41.04, for increased accuracy. The entrance beam radius is set to 2 mm (2.35 spot sizes), so that the spot diagram grid approximates an untruncated incident Gaussian.

Aperture		Field		Conjugates	
Entr beam rad*	2.000000	Field angle	0.052007	Object dist	1.1017e+03
Object NA	0.001815	Object height*	-1.000000	Object to PP1	1.1028e+03
Ax. ray slope	-0.002160	Gaus image ht	0.840266	Gaus img dist	925.551802
Image NA	0.002160			PP2 to image	926.608955
Working f-nbr	231.430548			Magnification	-0.840266
Aperture divisions across pupil for spot diagram:					41.050000
Gaussian beam	Spot	1/e^2 radius on srf 1:	sdgx 0.848204	sdgy	0.848204

After entering the data, we close the Setup spreadsheet and use the Evaluate>>Spread Function>>Plot PSF Scans commmand with default options to compute the PSF. We see from the

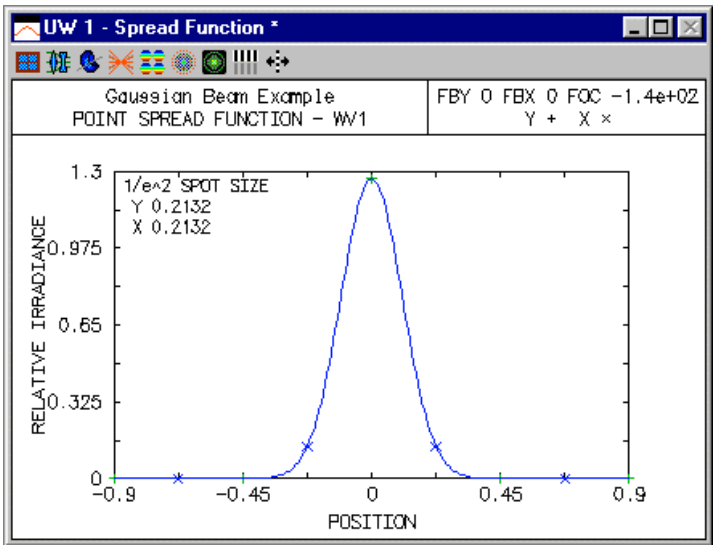
output (below) that the computed spot size is 0.2502 mm, essentially the same as the size predicted by the paraxial Gaussian beam trace.



If we compute the central value of the point spread function at the paraxial image plane and at the waist distance (z = –138.583 mm) the ratio if the irradiances is 0.726, essentially the same value as computed from the fundamental Gaussian beam solution to the wave equation, which predicts an irradiance ratio of $(w_0/w)^2 = (0.213/0.25)^2 = 0.725$.

```
*POINT SPREAD FUNCTION
 WAVELENGTH 1
      Y           X           Z           PSF       AMPLITUDE     PHASE
      --          --          --        0.921462    0.959928 -106.967090

*POINT SPREAD FUNCTION
 WAVELENGTH 1
      Y           X           Z           PSF       AMPLITUDE     PHASE
      --          --     -138.583415    1.267969    1.126041  -75.772180
```

We can repeat the above plot with a focus shift of -138.583 to compare the spot size at the beam waist with that found using the Gaussian beam spreadsheet. We need to increase the scale of the plot to accommodate the increased peak PSF (1.268). Again using the Evaluate>>Spread Function>>Plot PSF Scans commmand, we find that the point spread function size at z = –138.583 is 0.2132, consistent with the waist calculation of 0.2128 mm.



Truncated Gaussian beam

Now we will consider the effect of a finite aperture on the beam. If there is a diffracting aperture in the system that is not much larger than the spot size at the aperture, then we cannot use the usual Gaussian beam formulae to analyze the propagation of the beam. We need to compute the diffraction integrals taking into account the finite limits imposed by the aperture. Thus, we must use the OSLO PSF analysis routines based on the spot diagram. As an example, we will insert a real circular diaphragm (i.e. a checked aperture) just before the lens. The radius of the diaphragm will be equal to the $1/e^2$ spot size of the beam at that point. Make the aperture a checked aperture so that the beam is truncated at this point.

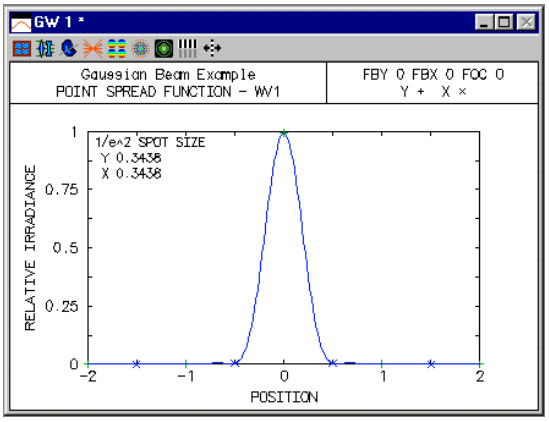
Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: Gaussian Beam Example | Zoom 1 of 1 | Efl 503.519128

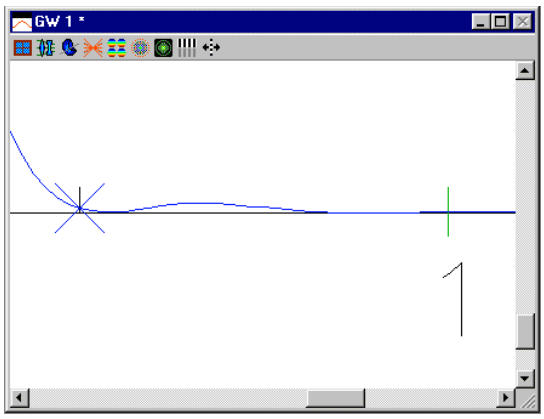
Ent beam radius 1.000000 | Object height -1.000000 | Primary wavln 0.632800

SRF	RADIUS	THICKNESS	APERTURE RADIUS	GLASS	SPECIAL
OBJ	0.000000	1.1017e+03	1.000000	AIR	
1	0.000000	0.000000	0.848204 K	AIR	
AST	MGLDX248	3.200000 F	22.500000 AF	FIXED F	
3		1.0060e+03	22.500000 F	AIR	
IMS	0.000000	0.000000	1.000000 S		

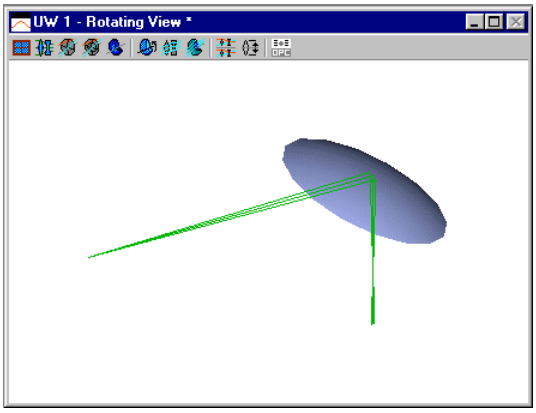
Now if we plot the *x* and *y* scans through the point spread function, we see that the spot size is no longer equal to the prediction of the Gaussian beam trace and the beam is no longer a Gaussian. By zooming the graphics window, you can see that the diffraction pattern exhibits evidence of the ring structure that is familiar from the analysis of uniformly illuminated pupils. (See, for example, Mahajan(8). Remember that OSLO normalizes the point spread function values to the peak of the perfect PSF for the same pupil size and focusing distance, so the irradiance normalizations are different for the above and below PSF plots.



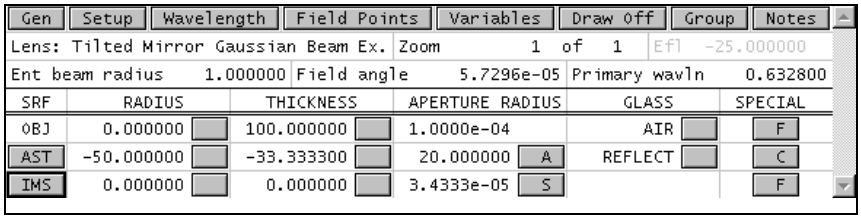
8 V. N. Mahajan, "Uniform versus Gaussian beams: a comparison of the effects of diffraction, obscuration, and aberrations," J. Opt. Soc. Am. A **3**, 470-485 (1986).


GW 1 *
1

Tilted spherical mirror



This example is taken from a paper by DeJager and Noethen(9) The system is a spherical mirror tilted 45 degrees. The input beam is circular but the output beam is highly astigmatic because of the large tilt. The system considered is a mirror with a radius of curvature of –50 mm, operating at a paraxial magnification of –1/3. Following DeJager and Noethen, enter the following system.




```
*TILT/DECENTER DATA
  1     DT    1          DCX        --        DCY        --        DCZ        --
                         TLA     45.000000    TLB        --        TLC        --
```

Setup an on-axis object point and use the astigmatic Gaussian beam trace to propagate a circular beam with an object surface spot size of 1 mm. The input waist is at the object surface.

```
*SET OBJECT POINT
          FBY          FBX          FBZ
          --           --           --
         FYRF         FXRF          FY           FX
          --           --           --           --
          YC           XC          YFS          XFS          OPL     REF SPH RAD
          --           --       11.859577  -21.358516   33.333300  -33.333300

*TRACE GAUSSIAN BEAM
 WAVELENGTH =   0.632800       M-SQUARED =    1.000000
 SRF     Y SPT SIZE  X SPT SIZE  BEAM AZMTH    Y RFR RAD   X RFR RAD PHASE AZMTH
         Y WST SIZE  X WST SIZE                Y WST DST   X WST DST
  0       1.000000    1.000000       --          --          --          --
          1.000000    1.000000                   --          --

  1       1.414500    1.000203       --        17.678937   35.360409       --
          0.003560    0.007121                  17.678713   35.358617

  2       0.885686    0.057730       --       -15.654840    2.056608       --
          0.003560    0.007121                 -15.654587    2.025317
```

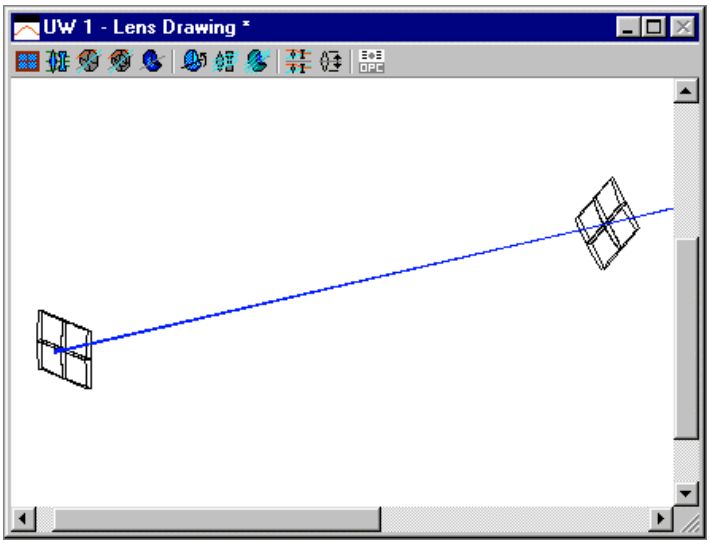
These results are nearly identical with the image space beam calculated by DeJager and Noethen. Note the difference between the wavefront radii of curvature for the Gaussian beam and the geometric field sags (YFS and XFS of the reference ray output).

General astigmatism

This example is taken from the paper by Arnaud and Kogelnik mentioned earlier in this chapter. The system consists of two cylindrical lenses, with a relative orientation between their cylinder

9 D. DeJager and M. Noethen, “Gaussian beam parameters that use Coddington-based *Y-NU* paraprincipal ray tracing,” Appl. Opt. **31**, 2199-2205 (1992); errata: Appl. Opt. **31**, 6602 (1992).

axes of 45 degrees. This is a nonorthogonal system, and we would expect that a stigmatic incident beam should suffer from general astigmatism after passing through the two lenses. The paper states that the two lenses have focal lengths of 250 mm and 200 mm, and are separated by 500 mm. It is also stated that the input beam, of wavelength 0.6328 μm, has a waist that is located 500 mm in front of the first cylindrical lens. We can use the catalog database to find cylindrical lenses of the proper focal lengths and construct, for example, the following system.



Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: Arnaud/Kogelnik Example Lens Zoom 1 of 1 Efl -977.798815

Ent beam radius 1.000000 Field angle 5.7296e-05 Primary wavln 0.632800

SRF	RADIUS		THICKNESS		APERTURE RADIUS		GLASS		SPECIAL
OBJ	0.000000		500.000000		0.000500		AIR		
AST	MGLCP017		4.500000	F	11.250000	AFX	FIXED	F	A
2			500.000000		11.250000	FX	AIR		A
3	MGLCP015		5.000000	F	11.250000	FX	FIXED	F	CA
4			609.500000		11.250000	FX	AIR		CA
5	0.000000		100.000000		0.205230	S	AIR		
6	0.000000		100.000000		0.404114	S	AIR		
7	0.000000		100.000000		0.602998	S	AIR		
8	0.000000		100.000000		0.802091	S	AIR		
9	0.000000		100.000000		1.001277	S	AIR		
IMS	0.000000		0.000000		1.200462	S			

```
*TILT/DECENTER DATA
  3     DT    1          DCX        --        DCY        --        DCZ        --
                         TLA        --        TLB        --        TLC    45.000000
  4     RCO   1
        DT    1          DCX        --        DCY        --        DCZ        --
                         TLA        --        TLB        --        TLC        --
```

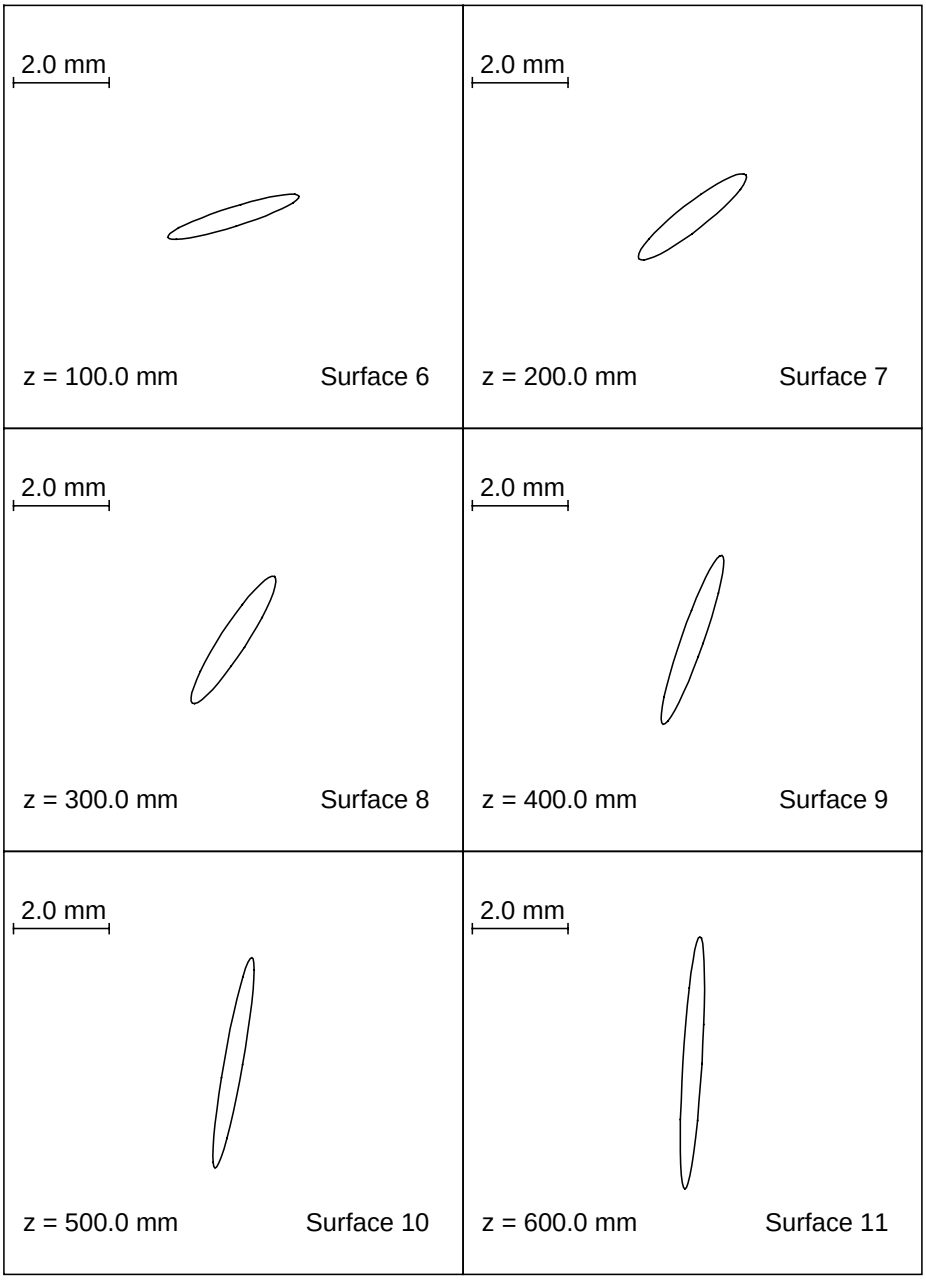
Note that surface 4 has a return coordinates specification to restore the remaining surface to untilted coordinates. Surfaces 5 through 10 are dummy surfaces, placed at 100 mm intervals, to correspond to the observation planes in Fig. 6 of the Arnaud and Kogelnik paper. Unfortunately, the paper does not give the value of the input beam waist that was used to generate the photographs in Fig. 6. We will assume a circular input beam with a 250 μm diameter, i.e., a waist size of 0.125 mm.

```
>> trr 0

*SET OBJECT POINT
         FBY          FBX          FBZ
         --           --           --
        FYRF         FXRF          FY          FX
```

```
         --          --          --          --
         YC          XC          YFS         XFS         OPL    REF SPH RAD
         --          --     -602.938147   65.736979  1.1144e+03  433.336428

>> tgb ful all 0.125 0.125 0.0 0.0 0.0

*TRACE GAUSSIAN BEAM
 WAVELENGTH =   0.632800       M-SQUARED =    1.000000
 SRF     Y SPT SIZE  X SPT SIZE  BEAM AZMTH   Y RFR RAD   X RFR RAD PHASE AZMTH
         Y WST SIZE  X WST SIZE               Y WST DST   X WST DST
  0        0.125000    0.125000      --          --          --          --
           0.125000    0.125000                  --          --

  1        0.815345    0.815345      --      744.967305 -775.778310      --
           0.120145    0.125000              728.791376 -757.544599
  2        0.810420    0.820075      --      488.794855 -514.933799      --
           0.120145    0.125000              478.051970 -502.970122

  3        0.125654    1.621025  -45.000000  517.175680 -532.661386   15.166995
      No waist information; beam has general astigmatism.
  4        0.126339    1.613054  -44.520709  353.080556 -310.473989   17.350078
      No waist information; beam has general astigmatism.

  5        0.203564    1.434653   17.423235 -160.508935  446.318575   -7.393698
      No waist information; beam has general astigmatism.

  6        0.275956    1.415702   37.991769 -229.910974  407.348510   -3.418038
      No waist information; beam has general astigmatism.

  7        1.575798    0.277228  -32.997671 -310.935616  361.858250   -0.610825
      No waist information; beam has general astigmatism.

  8        1.866401    0.234750  -19.257529 -387.981434  355.943169    3.149425
      No waist information; beam has general astigmatism.

  9        2.230996    0.196775   -9.949000 -442.951614  611.649234   11.359960
      No waist information; beam has general astigmatism.

 10        2.636856    0.198718   -3.472671 -414.322269 -1.9493e+03   37.601892
      No waist information; beam has general astigmatism.
```

These results are consistent with the photographs in Fig. 6 and the discussion of Section VII of the paper. Just after the second lens (surface 4) the beam is nearly horizontal. After the second cylindrical lens, the beam suffers from general astigmatism and no waist information can be calculated. As the beam propagates (surfaces 5 – 10), it changes size and rotates toward a more vertical orientation. The spot size and wavefront axes are never aligned as the beam propagates, since the BEAM AZMTH and PHASE AZMTH angle are never the same.

Using the values of the Y SPT SIZE, X SPT SIZE, and BEAM AZMTH computed on surfaces 5 through 10, we can draw the spot ellipse as the beam propagates beyond the cylindrical lenses. The ellipses, shown below, can be compared to the experimental results presented in Fig. 6 of the Arnaud and Kogelnik paper.


2.0 mm
z = 100.0 mm
Surface 6
2.0 mm
z = 200.0 mm
Surface 7
2.0 mm
z = 300.0 mm
Surface 8
2.0 mm
z = 400.0 mm
Surface 9
2.0 mm
z = 500.0 mm
Surface 10
2.0 mm
z = 600.0 mm
Surface 11

Laser cavity design

Usually, a laser consists of some gain medium which is placed in a cavity (usually two mirrors) to provide feedback. These laser cavities (or resonators) support *modes* of propagation; the fundamental mode is the Gaussian beam studied in this chapter. A resonator mode is a configuration of the optical field that is self-consistent, i.e., the beam parameters for the field are the same after each complete round trip of the wave through the cavity. Thus, the modes are axial standing wave patterns in the cavity. For a stable mode, the beam wavefront radius of curvature is equal to the radius of curvature of the mirror, when the field is incident upon each cavity mirror. In this example, we will design a simple Fabry-Perot cavity (two plane mirrors).

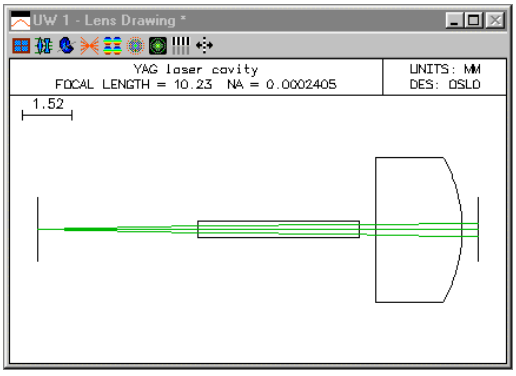
Since the cavity has plane mirrors and the wavefront radii of curvature at the mirrors are to equal to the mirror radii for a mode, the beam radii must be infinite on the mirrors. In other words, there must be a beam waist located at each mirror. To study the propagation of the beam from one mirror to the other, we only need to enter the optical system such that the object surface corresponds to one of the cavity mirrors and the image surface corresponds to the other mirror. Inside the cavity, we have the gain medium and a focusing lens. The gain medium is a 5 mm long, 0.5 mm diameter tube of neodymium-doped yttrium aluminum garnet (Nd:YAG, refractive index 1.82) and the lens is a 10 mm focal length, plano-convex fused silica lens from Melles Griot (Part No. 01LQF005). The lens is separated from the YAG rod by 0.5 mm and the second mirror (the image surface in our case) is 0.5 mm from the convex surface of the lens. We start with the YAG rod 5 mm from the object surface (the first mirror). Note that the lens has been reversed from its orientation in the catalog lens data base.

Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: YAG laser cavity | Zoom 1 of 1 | Efl 10.232836

Object num aper 0.020000 | Object height -1.000000 | Primary wavln 1.064000

SRF	RADIUS	THICKNESS	APERTURE RADIUS	GLASS	SPECIAL
OBJ	0.000000	5.000000	1.000000	AIR	F
AST	0.000000	5.000000	0.250000 A	ND-YAG	
2	0.000000	0.500000	0.250000	AIR	
3	MGLQF005	2.700000 F	2.250000 F	FIXED F	
4		0.500000	2.250000 F	AIR	
IMS	0.000000	0.000000	1.000000		F



We need to find the correct separation from the object to the YAG rod in order to have a Gaussian beam waist on both mirrors. Thus our first variable is the thickness of surface 0. The other unknown quantity is what the beam waist size is for the mode. Unfortunately, the beam size is not one of the variable types in OSLO. We can, however, use a “dummy” variable to represent to waist size. For example, since surface 0 has a curvature of zero, changing the conic constant has no effect on the shape of the surface or the optical properties of the system. Thus, we can make the conic constant of surface 0 a variable, with the understanding that it represents the object space beam waist. Since a waist size of 0 is not allowed, we start with a value of 0.01.

```
*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF        CC          AD          AE          AF          AG
  0       0.010000      --          --          --          --
*VARIABLES
 VB   SN  CF  TYP       MIN         MAX        DAMPING      INCR        VALUE
V 1    0   -  TH      4.000000    6.000000    1.000000  1.0002e-05     5.000000
```

```
V 2    0   -  CC    1.0000e-05    0.500000    1.000000  1.0000e-05     0.010000
```

We will use the astigmatic beam trace and SCP to compute the necessary operands. The beam will have a waist on surface 0 and a spot size (i.e., waist size) equal to the value of the object surface conic constant. One operand will be the waist distance for the image surface (surface 5). We want the image space waist to be at surface 5, so this value should be zero. Also, the beam should be confined to the YAG rod, so we target the beam size exiting the rod (surface 2) to be 2/3 of the radius of the rod. The SCP command "***yagmode**" computes these operand components.

```
*OPERATING CONDITIONS: OPTIMIZATION
   …….
   CCL/SCP operands command:    *yagmode
```

```
*yagmode
set_preference(outp, off);
i = sbrow;
ssbuf_reset(i, 16);
trace_ref_ray(0.0, 0.0, 0.0, 0.0, 0.0);
trace_gaussian_beam(ful, all, cc[0], cc[0], 0.0, 0.0, 0.0);
Ocm[1] = ssb(9, 1);  // Spot size on surface 2
Ocm[2] = ssb(16, 3); // Waist distance from image surface
ssbuf_reset(-i, 0);
set_preference(outp, on);
```

In terms of the above callback command, the operands are as follows:

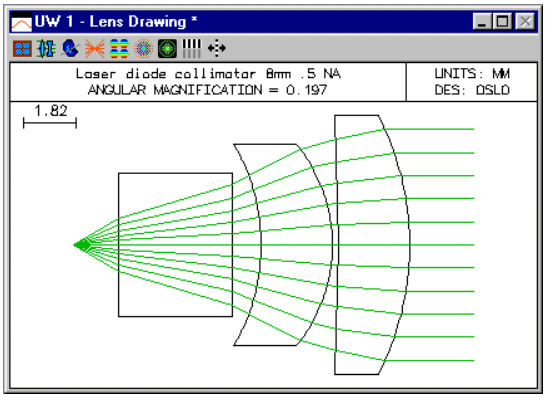
OP	MODE	WGT	NAME	DEFINITION
1	Min	1.000000	Spotsize	OCM1-0.1667
2	Min	1.000000	Waistdist	OCM2

Once the operands and variables are properly entered, we can use the Ite command on the text output toolbar to iterate the design. After the optimization process has converged, we examine the variables and operands. Tracing the resulting mode beam confirms that the output waist is located on surface 5 and that the beam size at the YAG rod is the desired value.

```
*VARIABLES
 VB   SN  CF  TYP       MIN         MAX        DAMPING      INCR        VALUE
V 1    0   -  TH      4.000000    6.000000  720.576161  1.0002e-05     5.070257
V 2    0   -  CC    1.0000e-05    0.500000  4.2833e+04  1.0000e-05     0.015956

*OPERANDS
 OP    DEFINITION                MODE     WGT     NAME          VALUE   %CNTRB
O 1    "OCM1-0.1667"               M    1.000000 Spot size   9.7145e-16   5.16
O 2    "OCM2"                      M    1.000000 Waist dist  4.1662e-15  94.84
MIN ERROR:   3.0250e-15

*TRACE GAUSSIAN BEAM
 WAVELENGTH =   1.064000
 SRF    Y SPT SIZE  X SPT SIZE  BEAM AZMTH   Y RFR RAD   X RFR RAD PHASE AZMTH
        Y WST SIZE  X WST SIZE               Y WST DST   X WST DST
  0       0.015956    0.015956      --          --          --          --
          0.015956    0.015956                  --          --

  1       0.108798    0.108798      --       -9.430706   -9.430706      --
          0.015956    0.015956               -9.227868   -9.227868
  2       0.166700    0.166700      --       -7.889794   -7.889794      --
          0.015956    0.015956               -7.817510   -7.817510

  3       0.177267    0.177267      --      -12.155806  -12.155806      --
          0.015956    0.015956              -12.057320  -12.057320
  4       0.216670    0.216670      --      3.8427e+04  3.8427e+04      --
          0.216669    0.216669                0.500000    0.500000

  5       0.216669    0.216669      --      4.6117e+18  4.6117e+18      --
          0.216669    0.216669              4.1662e-15  4.1662e-15
```

Laser-diode collimating lens

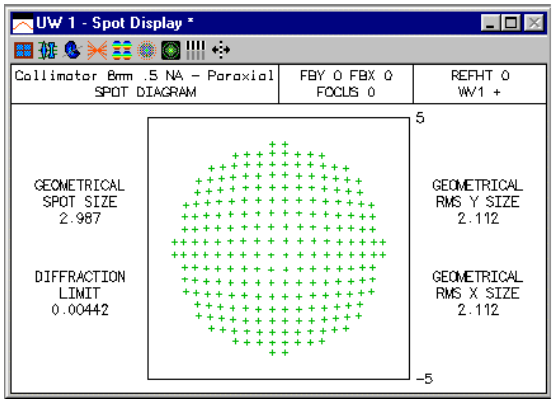
This example illustrates the difference between paraxial and aplanatic ray aiming used for evaluation of high-speed lenses. The lens here is designed to take light from a laser diode and produce a collimated beam. It is a commercially available design available from Melles Griot as their part number 06GLC002. It is designed for a numerical aperture of 0.5 on the short conjugate side, and has a focal length of about 8mm.

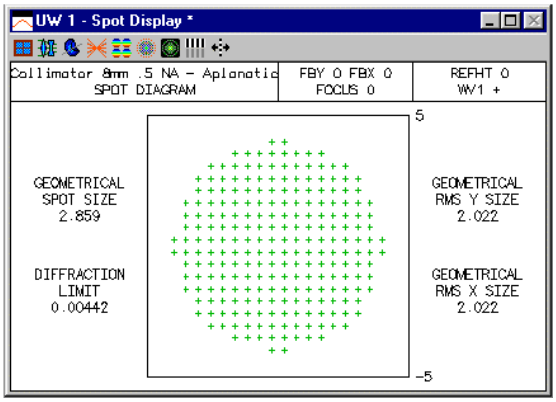


In traditional optical design, lenses are designed with the long conjugate side on the left. There are two reasons for this convention. First, there is a maximum distance that rays can be traced without loss of numerical accuracy using ordinary ray trace equations (in OSLO, this distance is 10^8 units). Many programs are set up to take object distances greater than this as being at infinity, for which special equations are used. When the long distance is on the image side, the system must be evaluated in afocal mode. This is not a problem for OSLO, which has built-in afocal mode support.

The second reason has to do with the way that rays are aimed at the lens from object space. In traditional programs, rays are aimed at a flat entrance pupil. This means that fractional coordinates of rays are proportional to their direction tangents in object space. When the object is at a great distance, this is ok, but actually fractional coordinates should be proportional to the direction cosines of rays in object space. We call this aplanatic ray aiming, as opposed to paraxial ray aiming. Aplanatic ray aiming was used in GENII for many years, and has been introduced into OSLO since the programs were merged in 1994. It has the advantage that OSLO can now be used to evaluate systems from short to long conjugate, which is not possible with a program that uses paraxial ray aiming.

The laser diode collimator is a fast enough system for the differences between paraxial and aplanatic ray aiming to be readily observable. The figures below shows the two cases (the system was changed to focal mode to produce these plots).





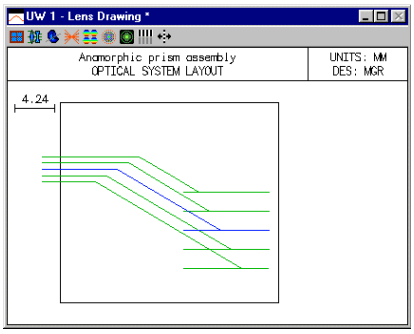
Beam circularizer for diode lasers

Most diode lasers emit asymmetric beams. The numerical aperture is different in the *yz* and *xz* planes, and often the beam has astigmatism, which is a separate issue. One way to make the beam circular is to use a pair of anamorphic prisms, as shown in this example. The prisms used here are available as Melles Griot part number 06GPU001. They work in collimated light (otherwise they would add astigmatism), so in an actual application the prisms must be used in combination with a collimator. Such a system is included as the file diodassy.len, described below.

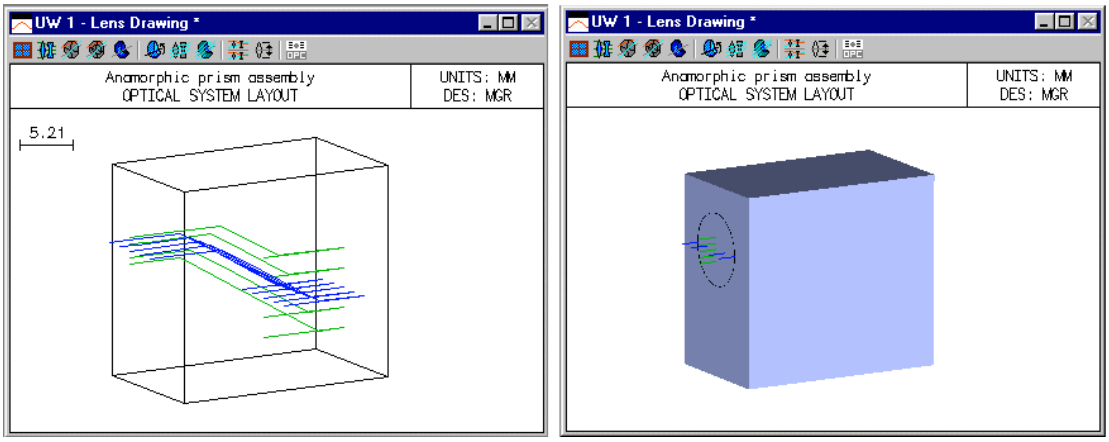
The prisms have a vertex angle of 29.43333 degrees, a width of 12mm, and a maximum thickness perpendicular to the back face of 8.5mm. For a given entry angle to the first prism, the angle of the

second prism is fixed by the requirement that the beam emerge parallel to itself. The displacement of the beam depends on the prism separation. The listing below shows how the system should be set up in OSLO. Surfaces 4 and 6 are expressed in the coordinate system of surface 1 using a return_coordinates (**rco**) command. The **rco** command goes on the preceding surface and indicates that the coordinates of the next surface are to be taken according to the dcx, dcy, dcz, tla, tlb, and tlc relative to a base surface (here, surface 1).

In order to prevent a confusing drawing caused by the tilted surfaces, the surfaces themselves are marked not drawable (in the Surface Control spreadsheet). A plan view of the system shows just the ray trajectories. Note that although the drawing makes it look like there are only two surfaces, there are actually 4. The rays are close enough to normal incidence on the other two that the drawing doesn't show them.



To substitute for the missing prism surfaces, the entire assembly has been placed in a box, using **bdi** (boundary data information) data. OSLO graphics routines can be instructed to put 3D objects on a drawing that are totally unrelated to the optical function of the depicted system. These objects are specified by a list of vertices (**vx**) and polygon faces (**pf**), as shown in the listing below. To enter such data yourself, open the lens file in the text editor and use the same scheme. The vertex and face information must be preceded by a **bdi** command, which gives the number of data items. The final solid-model drawing of the system is as follows:

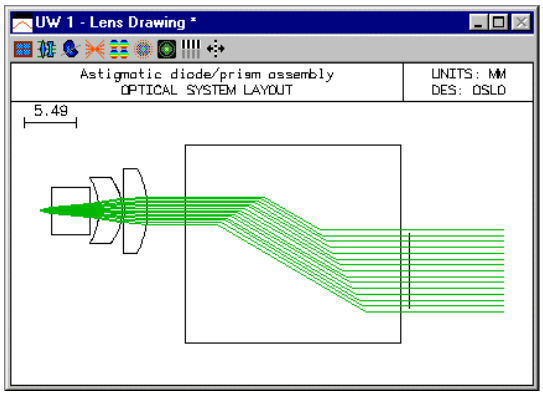



```
*LENS DATA
Anamorphic prism assembly
 SRF       RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0          --       1.0000e+20    1.0000e+18             AIR

  1          --         6.000000      4.000000 AS          AIR   *

  2          --         5.872557    1.0000e-06           SF11 C *
  3          --            --       1.0000e-06             AIR   *

  4          --         5.872557    1.0000e-06           SF11 C *
  5          --            --       1.0000e-06             AIR   *

  6          --            --         4.000000             AIR   * Prism assy

  7          --        -0.003198      4.179133 S
```

```
*TILT/DECENTER DATA
  2     DT    1         DCX        --       DCY        --       DCZ        --
                        TLA   -59.800000   TLB        --       TLC        --
  3     RCO   1
        DT    1         DCX        --       DCY        --       DCZ        --
                        TLA    29.433333   TLB        --       TLC        --
  4     DT    1         DCX        --       DCY    -6.400000    DCZ    16.866459
                        TLA    29.292833   TLB        --       TLC        --
  5     RCO   1
        DT    1         DCX        --       DCY        --       DCZ        --
                        TLA   -29.433333   TLB        --       TLC        --
  6     DT    1         DCX        --       DCY    -6.400000    DCZ    23.000000
                        TLA        --       TLB        --       TLC        --

*SURFACE TAG DATA
  1     LMO EGR  (6 surfaces)
  1     DRW AP
  6     DRW AP

*BOUNDARY DRAWING DATA
 SRF 1:
 VX NBR         X            Y            Z        COORD SURF
    1        7.000000     7.000000       --             1
    2       -7.000000     7.000000       --             1
    3       -7.000000   -14.000000       --             1
    4        7.000000   -14.000000       --             1
    5        7.000000     7.000000    23.000000         1
    6       -7.000000     7.000000    23.000000         1
    7       -7.000000   -14.000000    23.000000         1
    8        7.000000   -14.000000    23.000000         1
 PF NBR        VX1          VX2          VX3          VX4
    1           1            2            3            4
    2           1            5            6            2
    3           5            8            7            6
    4           8            7            3            4
    5           1            4            8            5
    6           2            3            7            6
```

Shaping a diode laser beam

This file combines a diode-laser collimator (diodcoll.len), a cylindrical lens, and an anamorphic prism assembly (anaprism.len) to create an overall system that converts the light from a hypothetical diode laser having a beam divergence ratio of 3:1 and 10 microns of astigmatism into a collimated circular Gaussian beam having a wavefront quality of better than 0.25λ. The diode is assumed to be single mode, and to have a numerical aperture in the *xz* plane of 0.3, and a numerical aperture in the *yz* plane of 0.1. The general layout of the system is as shown below. For additional information on the collimator see p 345 and for additional information on the prism assembly see p 346




```
*LENS DATA
Astigmatic diode/prism assembly
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0         --          1.578414    1.0000e-06            AIR

  1   ELEMENT GRP     10.200000      2.500000 A          SF11 C *
  6    Collimator      3.000000      4.500000             AIR     Collimator

  7       ELEMENT      1.000000      4.000000             BK7 C *
  8    Astig corr        --          4.000000             AIR   * Astig corr

  9   ELEMENT GRP     17.745114      4.000000             AIR   *
 14    Prism assy        --          4.000000             AIR   * Prism assy

 15       ELEMENT      1.000000      4.000000             BK7 C *
 16    Out window        --          4.000000             AIR     Out window

 17         --            --          2.515146 S
```

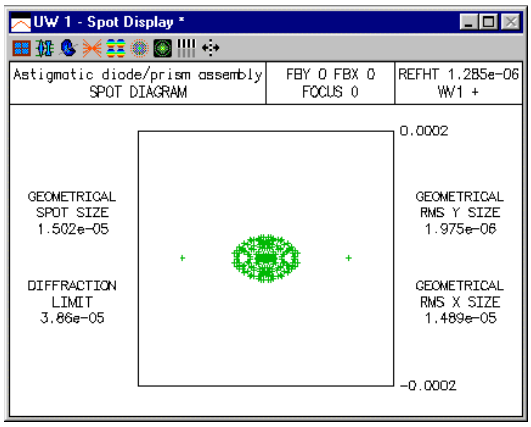
The astigmatism of the source is listed as the general operating condition **sasd** on the general operating conditions, as shown below. The value is the distance between the apparent source locations in the *yz* and *xz* meridians, 0.01 millimeters in the present example.

```
*OPERATING CONDITIONS: GENERAL
   .
   .
   Source astigmatic dist:    0.010000    Ray aiming mode:          Aplanatic
   Temperature:              20.000000    Pressure:                  1.000000
```

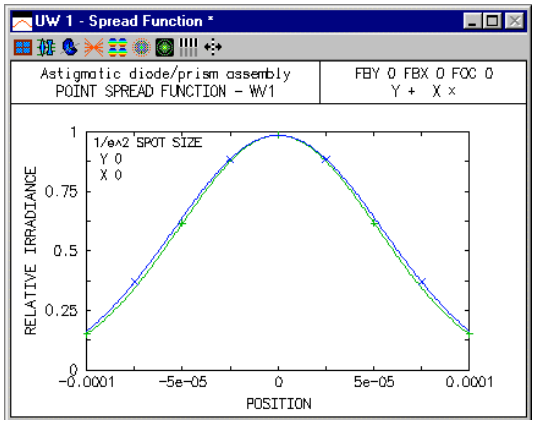
The numerical aperture of the system is listed as 0.3 on the surface data spreadsheet. This tacitly assumes that the beam is circular. The ellipticity of the beam is indicated in the spot diagram operating conditions, since that is the place where it is important. The spot size in the *y*-direction is called **ssy**, and the spot size in the *x* direction is called **ssx**. Since the diode aperture is specified in NA, the spot size must be given as ss = th[0]*tan(asin(NA)). Which yields ssy = .159, ssx = .496. The data below show the results of a spot diagram. Note that since the system is afocal, the spot data appears in angular measure (radians). Note also that the spot is much larger in the *x* direction than the *y* direction, as confirmed by the plot.

```
*SPOT DIAGRAM: MONOCHROMATIC APODIZED
 APDIV    11.050000
 WAVELENGTH 1
 WAV WEIGHTS:
       WW1
```

```
    1.000000
 GAU   SSY         SSX
    0.100000    0.300000
 NUMBER OF RAYS TRACED:
       WV1
        96
 PER CENT WEIGHTED RAY TRANSMISSION:     6.532767

*SPOT SIZES
  GEO RMS YA  GEO RMS XA  GEO RMS RA  DIFFR LIMIT    CENTYA       CENTXA
  1.3927e-06  9.4077e-06  9.5102e-06  6.8377e-05      --           --

*WAVEFRONT RS
 WAVELENGTH 1
   PKVAL OPD     RMS OPD  STREHL RATIO    RSY          RSX          RSZ
    0.031658    0.007022    0.998711  1.1417e-10      --           --
```



Spot diagrams only show the intersection points of rays with the image surface, not the ray weights. In the present case, the different ssx and ssy values put different weights on the rays (you can confirm this using the Calculate >> Display spot diagram command and selecting ray weights). The weights affect calculations such as energy distributions, and more particularly Fourier transforms, which are used to compute the intensity distribution in the emergent beam. The plot below shows the point spread function (i.e. the far-field intensity distribution) for the present system. The abscissa is in radians, since the evaluation is in afocal mode.



Gaussian beam movie

OSLO contains commands for making and viewing movies. Movies are sequences of graphics displays ("frames") that are saved in a single file that can be "played back" using the Show_movie command, which is executed when you select a movie from the User >> Movies submenu. As an example, the file gbmovie.mov is shipped with all versions of OSLO. It illustrates Gaussian beam propagation through a system where there are two lenses within the Rayleigh range of the beam. In the movie, a laser emitting a collimated beam having a spot size that ranges between .02 and .5 mm is placed at the focal point of a singlet lens that is separated from another identical singlet by its focal length. To run the movie, select the Gaussian Beam entry on the Movies submenu, or try the command

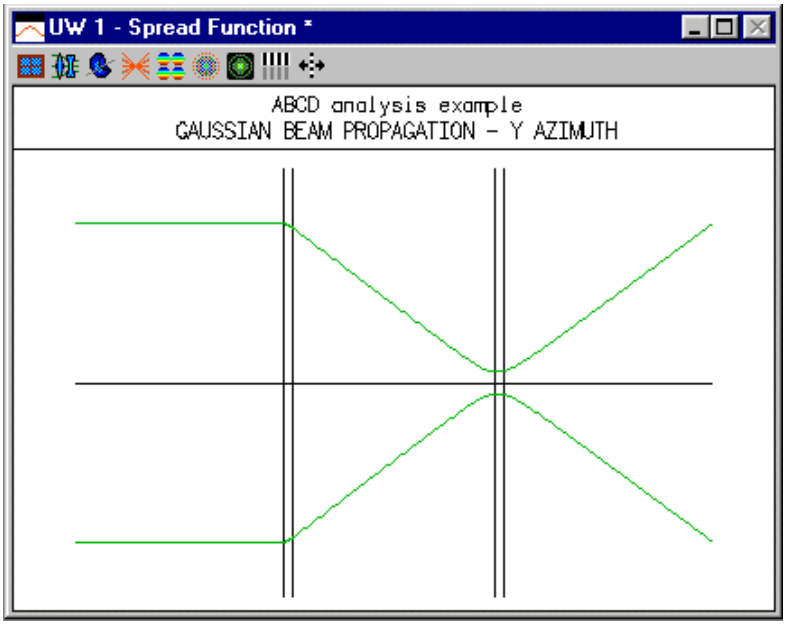
```
show_movie gbmovie fab 0 10
```

A typical output screen is shown below.

You can make movies yourself. There are two ways to make a movie. One, available in all versions of OSLO, is to open a movie file and save frames in it one by one. You can use SCP to automate the process. The commands required for this are as follows:

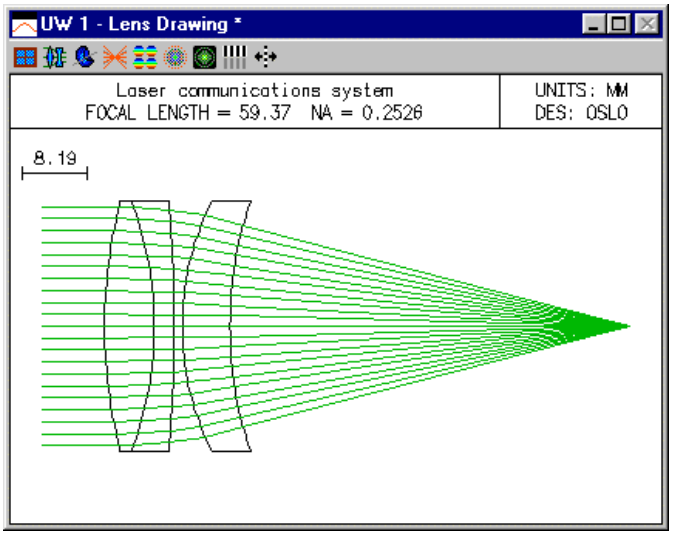
```
Open_movie(char Filename[])

Save_frame(void)  /* repeat as needed */

Close_movie(void)
```

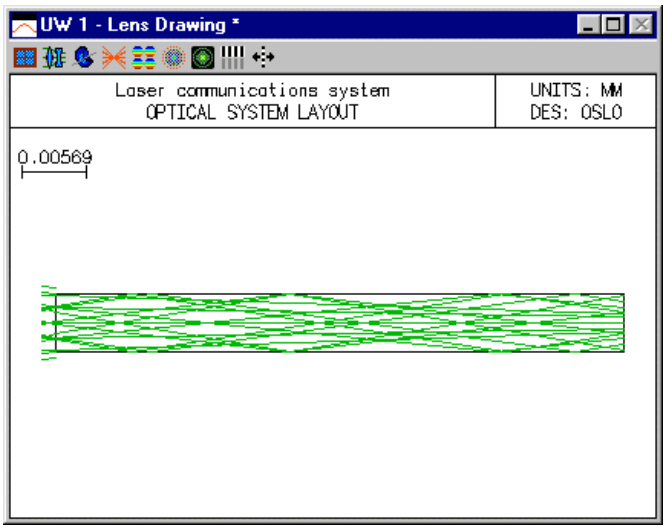
For more information on movies, see the OSLO Help system.



Aplanatic laser focusing system

This system is designed in chapter 5 using Melles Griot catalog lenses. Here, a small “light pipe” has been added in the image plane to simulate a fiber. The light pipe has a diameter of 0.005mm, and a length of 0.05mm, so it is similar in geometrical size to the one postulated for the example. To see the fiber, you must make a special drawing in which you limit the surfaces drawn to 6 and 7, then you can zoom in as much as possible. The result is the second drawing below.





You can readily see that the extreme rays miss the edge of the fiber. Of course no particular quantitative information can be obtained from this, since the system is close to the diffraction limit. If you want to obtain detailed information on the coupling into the fiber, you should use the Options >> Fiber coupling command. If you want to use this command, you should first remove the light pipe (surfaces 6 and 7) from the system, since the command assumes that the fiber is located in the image plane.

Fiber coupling

As a simplified example of the calculation of fiber coupling efficiency, consider the case of coupling the output of a diode laser into a fiber. A common technique to accomplish this is the use of ball lenses, since small diameter spheres are easier to manufacture than thin lenses of the same diameter. We assume that the diode has far-field divergence half angles of 30° in *y* and 10° in *x*. In the Gaussian beam chapter, it is shown that the relationship between beam waist size w_0 and divergence angle θ is

$$\theta = \tan^{-1}\left(\frac{\lambda}{\pi w_0}\right) \tag{10.78}$$

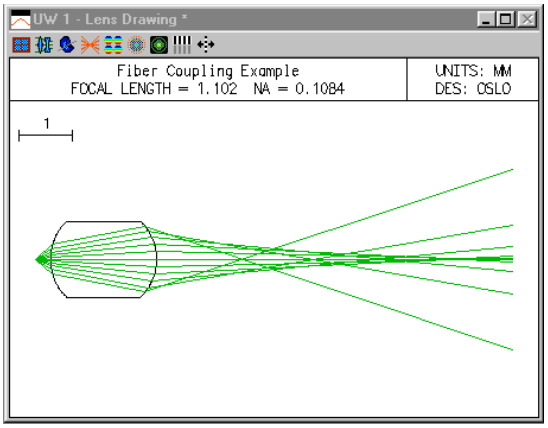
Assuming a wavelength of 0.83 μm, this leads to beam waists of w_{0y} = 0.458 μm and w_{0x} = 1.498 μm. This beam is, of course, elliptical. Since we will be using a rotationally symmetric ball lens, we need to choose an appropriate magnification for coupling to the fiber mode, which is circular. A circular beam with the same cross-sectional area at the waist would have a waist size of $w_0 = (w_{0x}w_{0y})^{1/2}$ = 0.828 μm. Thus, assuming a 5 μm radius Gaussian mode and the diode waist as the object, we choose a nominal paraxial magnification of $m = -(5/0.828) \approx -6$. Using a 1 mm radius fiber coupling sphere from the Melles Griot catalog, we construct the following system.

Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: Fiber Coupling Example | Zoom 1 of 1 | Efl 1.101696

Ent beam radius 1.100000 | Field angle 5.7296e-05 | Primary wavln 0.830000

SRF	RADIUS	THICKNESS	APERTURE RADIUS	GLASS	SPECIAL
OBJ	0.000000	1.285313	1.2853e-06	AIR	F
AST	0.000000	-1.000000	1.100000 AS	AIR	
2	MGLMS202	2.000000 F	0.720000 F	FIXED F	
3		6.711875	0.720000 F	AIR	
IMS	0.000000	0.000000	9.8887e-06 S		

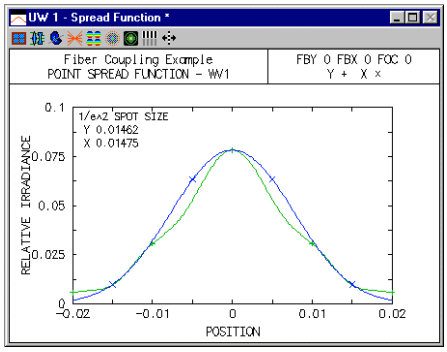


Note that we have located the aperture stop at the center of the sphere. The spot diagram operating conditions are set so that the Gaussian apodization matches our assumed divergence angles of 30° and 10°. Thus the entering spot sizes are th[0]*tan(30°) = 1.285*0.577 = 0.742 mm and th[0]*tan(10°) = 1.285*0.176 = 0.227 mm.

```
*OPERATING CONDITIONS: SPOT DIAGRAM
   Aperture divisions:      100.000000    Use Gaussian pupil apodization:  On
   X 1/e^2 entr. irrad.:      0.226635    Y 1/e^2 entr. irrad.:      0.742076
   Use all wavelengths in diagram:  On    P-V OPD for MTF switch:    3.000000
   Use equal image space incrmnts.:Off    Through-foc. frequency:   25.000000
   Diffraction efficiency calcs.:  Off
```

A Gaussian beam trace confirms that the input beam is imaged with a spot size magnification of –6 and the average spot size is $(2.75*8.99)^{1/2}$ = 5 μm. This Gaussian beam analysis only considers the propagation of the beam in a small region around the axis. This ball lens has a large amount of spherical aberration and the actual diffraction pattern is not the ideal Gaussian shape.

```
*TRACE GAUSSIAN BEAM
 WAVELENGTH =   0.830000
 SRF     Y SPT SIZE  X SPT SIZE  BEAM AZMTH   Y RFR RAD   X RFR RAD PHASE AZMTH
         Y WST SIZE  X WST SIZE               Y WST DST   X WST DST
  0       0.000458    0.001498      --          --          --          --
          0.000458    0.001498                  --          --

  1       0.742076    0.226640      --       -1.285313   -1.285369      --
          0.000458    0.001498               -1.285313   -1.285313

  2       0.164726    0.050331      --       -0.684757   -0.685547      --
          0.000600    0.001964               -0.684748   -0.684503
  3       0.645851    0.197248      --        6.711874    6.711661      --
          0.002746    0.008980                6.711752    6.697749

  4       0.002746    0.008990      --       -6.610179   -6.610179      --
          0.002746    0.008980               -0.000123   -0.014127
```



This departure from the 5 µm Gaussian shape of the mode is reflected in the computation of the coupling efficiency, which is about 18%.

```
*FIBER COUPLING EFFICIENCY - WAVELENGTH 1
 GAUSSIAN MODE - 1/e**2 RADIUS =    0.005000
 FIBER DISPLACEMENT   Y       --       X       --
 FIBER TILT          TLB      --      TLA      --
 POWER COUPLING =    0.1826       (  -7.385    dB)
 AMPLITUDE COUPLING    REAL =   -0.1981     IMAGINARY =    0.3786
```

The above efficiency was calculated at paraxial focus. It is well known, of course, that in the presence of spherical aberration, best focus is not located at the paraxial focus. Introduction of a focus shift can also be used to increase the coupling efficiency. For example, shifting the end of the fiber by 620 µm towards the lens increases the efficiency to just over 40%.

```
*LENS DATA
Fiber Coupling Example
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE   NOTE
  0        --          1.285313    1.2853e-06            AIR

  1        --         -1.000000      1.100000 AS         AIR

  2      06LMS202 F    2.000000 F    0.800000 F         FIXED F *
  3               F    6.711875      0.800000 F          AIR

  4        --         -0.620000      0.088442 S

*FIBER COUPLING EFFICIENCY - WAVELENGTH 1
 GAUSSIAN MODE - 1/e**2 RADIUS =    0.005000
 FIBER DISPLACEMENT   Y       --       X       --
 FIBER TILT          TLB      --      TLA      --
 POWER COUPLING =    0.4255       (  -3.711    dB)
 AMPLITUDE COUPLING    REAL =    0.418      IMAGINARY =    0.5008
```

Polarization and vector diffraction

Malus's law

Malus's law states that if linearly polarized light is incident upon an ideal linear polarizer, the intensity of the transmitted light is given by

$$I(\theta) = I(0)\cos^2\theta \tag{10.79}$$

where θ is the angle between the pass-plane of the polarizer and the azimuth of the incident linear polarization and $I(0)$ is the transmitted intensity when $\theta = 0$. We can prove this result using the Jones calculus formalism developed in the previous section. Assume we have incident light that is linearly polarized in the *y* direction. Assuming, for simplicity, that the light has unit intensity, the incident Jones vector is

$$\mathbf{E}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{10.80}$$

The Jones matrix for a linear polarizer is given by Eq. **(2.78)**. Since the incident light is *y*-polarized, the angle ϕ in Eq. **(2.78)** is equivalent to the angle θ in Eq. (10.79). Using the general transformation law for the Jones calculus [Eq. **(2.74)**], we find that the Jones vector for the transmitted wave is

$$\mathbf{E}_t = \begin{bmatrix} \cos\phi\sin\phi \\ \cos^2\phi \end{bmatrix} \tag{10.81}$$

The intensity of the transmitted light is the sum of the squared moduli of the *x* and *y* components of $\mathbf{E}_t$, i.e., $I(\phi) = |E_{tx}|^2 + |E_{ty}|^2$ or

$$\begin{aligned} I(\phi) &= \cos^2\phi\sin^2\phi + \cos^4\phi = \cos^2\phi\left(\sin^2\phi + \cos^2\phi\right) \\ &= \cos^2\phi \end{aligned} \tag{10.82}$$

This is Malus's law, with $I(0) = 1$. We can set up a simple system in OSLO to analyze this case, using the polarization element to model the linear polarizer. We start with the pass-plane of the polarizer aligned with the incident polarization ($\phi = 0°$).

```
*LENS DATA
Malus's Law Example
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0         --        1.0000e+20    1.0000e+14            AIR

  1         --            --          1.000000 AS         AIR

  2         --         10.000000      1.000000 S          AIR   *

  3         --            --          1.000010 S

*POLARIZATION ELEMENT DATA
               AMPLITUDE     PHASE              AMPLITUDE     PHASE
  2      JA       --          --          JB       --          --
         JC       --          --          JD     1.000000      --

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.500000
        1.000000

*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:      1.000000    Image axial ray slope:   1.0000e-20
   Object num. aperture:    1.0000e-20    F-number:                   --
   Image num. aperture:     1.0000e-20    Working F-number:        5.0000e+19
 FIELD
   Field angle:              5.7296e-05    Object height:          -1.0000e+14
   Gaussian image height:   -1.0000e+14    Chief ray ims height:    1.0000e-05
 CONJUGATES
   Object distance:          1.0000e+20    Srf 1 to prin. pt. 1:       --
   Gaussian image dist.:    -1.0000e+20    Srf 2 to prin. pt. 2:       --
   Overall lens length:         --         Total track length:      1.0000e+20
```

```
   Paraxial magnification:    1.000000    Srf 2 to image srf:       10.000000
 OTHER DATA
   Entrance pupil radius:     1.000000    Srf 1 to entrance pup.:      --
   Exit pupil radius:         1.000000    Srf 2 to exit pupil:         --
   Lagrange invariant:     -1.0000e-06    Petzval radius:          1.0000e+40
   Effective focal length:      --

*OPERATING CONDITIONS: GENERAL
   Image surface:                  3     Aperture stop:                   1
   Evaluation mode:           Afocal     Reference surface:               1
   Aberration mode:           Angular    Aperture checking in raytrace:   On
   Number of rays in fans:         21    Designer:                      OSLO
   Units:                          mm    Program:   OSLO SIX Rev. 5.10 SIN-G
   Wavefront ref sph pos:   Exit pupil   OPD reported in wavelengths:     On
   Callback level:                 0     Print surface group data:       Off
   Compute solves in configs:      Off
   Telecentric entrance pupil:     Off   Wide-angle ray aiming mode:     Off
   Aper check all GRIN ray segs:   Off   Extended-aper ray aiming mode:  Off
   Plot ray-intercepts as H-tan U: Off   XARM beam angle:          90.000000
   Source astigmatic dist:      --       Ray aiming mode:          Aplanatic
   Temperature:             20.000000    Pressure:                  1.000000

*OPERATING CONDITIONS: POLARIZATION
   Use polarization raytrace:       On    Degree of polarization:    1.000000
   Ellipse axes ratio:          --        Y to major axis angle:       --
   Handedness of ellipse:        Right    Use 1/4 wave MgF2 coating:      Off
```

Since the incident polarization and the pass-plane of the polarizer are aligned, all of the incident light should be transmitted, as the polarization ray trace data indicates.

```
*TRACE RAY - LOCAL COORDINATES
 SRF        Y           X           Z           YANG        XANG        D
        INTENSITY  DEG. POLRZ. ELL. RATIO  ELL. ANGLE   HANDEDNESS
  1         --          --          --          --          --          --
          1.000000    1.000000      --          --          --

  2         --          --          --          --          --          --
          1.000000    1.000000      --          --          --

  3         --          --          --          --          --       10.000000
          1.000000    1.000000      --          --          --
 PUPIL      FY          FX                                              OPD
            --          --                                              --
```

We now change the pass-plane angle of the polarizer to 30° by changing the elements of the Jones matrix for surface 2.

```
*POLARIZATION ELEMENT DATA
              AMPLITUDE     PHASE              AMPLITUDE     PHASE
  2      JA     0.250000      --          JB     0.433013      --
         JC     0.433013      --          JD     0.750000      --
```

Malus's law predicts that the transmitted intensity should be $\cos^2(30°) = 0.75$. The transmitted light should be linearly polarized at an angle of 30° from the *y*-axis. Tracing a polarization ray confirms this.

```
*TRACE RAY - LOCAL COORDINATES
 SRF        Y           X           Z           YANG        XANG        D
        INTENSITY  DEG. POLRZ. ELL. RATIO  ELL. ANGLE   HANDEDNESS
  1         --          --          --          --          --          --
          1.000000    1.000000      --          --          --


  2         --          --          --          --          --          --
          0.750000    1.000000      --       30.000000      --


  3         --          --          --          --          --       10.000000
          0.750000    1.000000      --       30.000000      --

 PUPIL      FY          FX                                              OPD
            --          --                                              --
```

A pass-plane angle of 60° results in a transmitted intensity of $\cos^2(60°) = 0.25$.

```
*POLARIZATION ELEMENT DATA
              AMPLITUDE     PHASE              AMPLITUDE     PHASE
  2      JA     0.750000      --          JB     0.433013      --
         JC     0.433013      --          JD     0.250000      --

*TRACE RAY - LOCAL COORDINATES
 SRF        Y           X           Z           YANG        XANG        D
        INTENSITY  DEG. POLRZ. ELL. RATIO  ELL. ANGLE   HANDEDNESS
```

```
  1         --          --          --          --          --          --
          1.000000    1.000000      --          --          --

  2         --          --          --          --          --          --
          0.250000    1.000000      --       60.000000      --

  3         --          --          --          --          --       10.000000
          0.250000    1.000000      --       60.000000      --

 PUPIL      FY          FX                                              OPD
            --          --                                              --
```

Finally, orienting the pass-plane of the polarizer along the *x*-axis ($\phi \Subset 90°$) results in complete attenuation of the incident light.

```
*POLARIZATION ELEMENT DATA
               AMPLITUDE     PHASE              AMPLITUDE     PHASE
  2      JA     1.000000      --          JB       --          --
         JC       --          --          JD       --          --

*TRACE RAY - LOCAL COORDINATES
 SRF        Y           X           Z           YANG        XANG        D
          INTENSITY  DEG. POLRZ. ELL. RATIO  ELL. ANGLE   HANDEDNESS
  1         --          --          --          --          --          --
          1.000000    1.000000      --          --          --

  2         --          --          --          --          --          --
            --          --          --          --          --

  3         --          --          --          --          --       10.000000
            --          --          --          --          --

 PUPIL      FY          FX                                              OPD
            --          --                                              --
```

Fresnel rhomb

It is easy to see that the Fresnel equations [Eqs. **(2.61) - (2.64)**] predict different amounts of reflected and transmitted electric fields for *s* and *p* components, except for the case of normal incidence ($\theta_i = 0$). This means that, in general, all surfaces in an optical system act as polarization elements, to a greater or lesser degree. In many systems, this is an undesirable effect, and a great amount of effort has been extended in order to produce coatings that are insensitive to polarization. On the other hand, there are optical elements that put the differences between *s* and *p* reflection coefficients to advantageous use. One of these devices is the *Fresnel rhomb*, which is used to convert linearly polarized light to circularly polarized light.

For angles greater than the critical angle, the Fresnel reflection coefficients are complex, indicating that the reflected waves undergo a phase shift. In addition, this phase shift is different for the *s* and *p* components. It can be shown that the relative phase difference δ between *s* and *p* polarization for a totally internally reflected wave is

$$\tan\frac{\delta}{2} = \frac{\cos\theta_i\sqrt{\sin^2\theta_i - \left(\frac{n'}{n}\right)^2}}{\sin^2\theta_i} \tag{10.83}$$

where the refractive index ratio n'/n is less than 1 and the angle of incidence is greater than the critical angle, i.e., $\sin\theta_i \geq n'/n$. The maximum relative phase difference δ_m is

$$\tan\frac{\delta_m}{2} = \frac{1-\left(\frac{n'}{n}\right)^2}{2\frac{n'}{n}} \tag{10.84}$$

Fresnel demonstrated how to use the phase difference to convert linearly polarized light to circularly polarized light. The incident, linearly polarized wave is oriented such that the electric vector makes an angle of 45° with the plane of incidence. Then, the incident *s* and *p* amplitudes are equal and if the beam is totally internally reflected, the reflected amplitudes remain equal. The refractive index ratio and angle of incidence must be chosen so that the relative phase difference δ is equal to 90°. If this is to be achieved with a single reflection, Eq. (10.84) implies that (using $\delta_m/2 = 45°$, so that $\tan\delta_m/2 = 1$)

$$1 < \frac{1-\left(\frac{n'}{n}\right)^2}{2\frac{n'}{n}} \tag{10.85}$$

This means that the refractive index ratio must be

$$\frac{n'}{n} < \sqrt{2} - 1 \tag{10.86}$$

or

$$\frac{n}{n'} > \frac{1}{\sqrt{2}-1} = 2.4142 \tag{10.87}$$

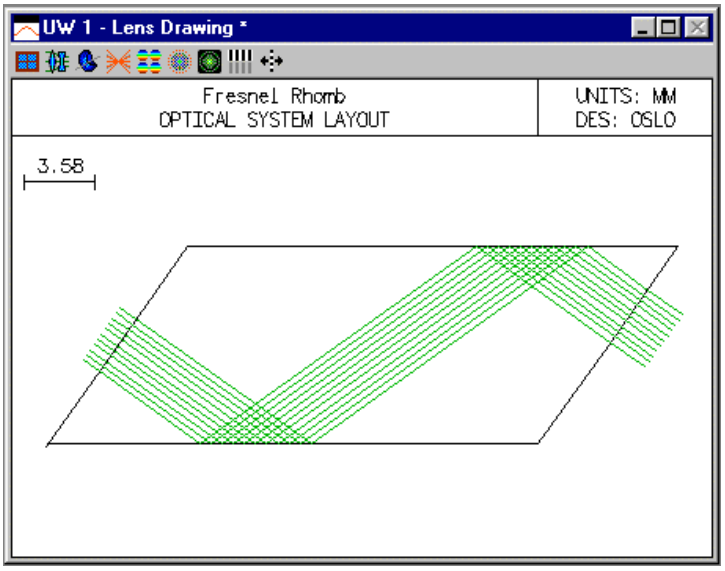
This is not a realistic value if it is desired to use an optical glass in air. Fresnel observed that if $n/n' = 1.51$, then Eq. (10.84) indicates that the maximum phase difference is 45.94°, so it should be possible to choose an angle of incidence such that $\delta = 45°$ and then use two total internal reflections to achieve a total phase shift of 90°. Using $n/n' = 1.51$ and $\delta = 45°$ in Eq. (10.83) yields two values for the angle of incidence: $\theta_i = 48.624°$ or $\theta_i = 54.623°$. A glass block, that produces two total internal reflections at either of these two angles, is called a Fresnel rhomb.

We can enter a Fresnel rhomb in OSLO by making use of the total internal reflection only surface. We do not want to designate the surfaces as mirrors, since we need the phase shift of total internal

reflection to achieve the desired change in polarization state. Using the larger value for the desired angle of incidence on the TIR surfaces, the prescription for the Fresnel rhomb is given below.

```
*LENS DATA
Fresnel Rhomb
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    7.1006e+19             AIR

  1        --            --          2.000000 AS          AIR

  2        --         12.500000      7.912300 X         GLASS2   *
  3        --         10.000000      8.000000 X           AIR    *
  4        --         25.000000     10.000000 X           AIR    *
  5        --         10.000000      7.912300 X           AIR    *

  6        --            --         11.451742 S

*TILT/DECENTER DATA
  2     RCO   2
        DT    1         DCX       --       DCY       --       DCZ       --
                        TLA   -35.376895   TLB       --       TLC       --
  3     DT    1         DCX       --       DCY    -5.000000   DCZ       --
                        TLA    90.000000   TLB       --       TLC       --
  4     RCO   1
        DT    1         DCX       --       DCY       --       DCZ       --
                        TLA       --       TLB       --       TLC       --
  5     DT    1         DCX       --       DCY       --       DCZ       --
                        TLA   -35.376895   TLB       --       TLC       --

*SURFACE TAG DATA
  3     TIR   1
  4     TIR   1

*OPERATING CONDITIONS: GENERAL
   Image surface:                  6    Aperture stop:                   1
   Evaluation mode:          Afocal    Reference surface:               1
   Aberration mode:          Angular   Aperture checking in raytrace:   On
   Number of rays in fans:        21   Designer:                      OSLO
   Units:                         mm   Program:   OSLO SIX Rev. 5.10 SIN-G
   Wavefront ref sph pos:   Exit pupil  OPD reported in wavelengths:     On
   Callback level:                 0   Print surface group data:       Off
   Compute solves in configs:    Off
   Telecentric entrance pupil:   Off   Wide-angle ray aiming mode:     Off
   Aper check all GRIN ray segs: Off   Extended-aper ray aiming mode:  Off
   Plot ray-intercepts as H-tan U: Off XARM beam angle:          90.000000
   Source astigmatic dist:       --    Ray aiming mode:          Aplanatic
   Temperature:           20.000000    Pressure:                  1.000000

*APERTURES
 SRF   TYPE APERTURE RADIUS
  0     SPC   7.1006e+19
  1     CMP    2.000000
  2     SPC    7.912300

     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -5.000000  AX2     5.000000  AY1    -6.132251  AY2     6.132251

  3     SPC    8.000000

     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -5.000000  AX2     5.000000  AY1    -8.949719  AY2    16.050281

  4     SPC   10.000000


     Special Aperture Group 0:
     A  ATP      Ellipse  AAC     Transmit  AAN       --
        AX1    -5.000000  AX2     5.000000  AY1   -16.050281  AY2     8.949719

  5     SPC    7.912300

     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -5.000000  AX2     5.000000  AY1    -6.132251  AY2     6.132251

  6     CMP   11.451742

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.500000
```

```
       1.000000

*REFRACTIVE INDICES
 SRF     GLASS            RN1         TCE
  0      AIR           1.000000       --
  1      AIR           1.000000   236.000000
  2      GLASS2        1.510000    84.000000
  3      AIR           1.000000   236.000000
  4      AIR           1.000000   236.000000
  5      AIR           1.000000   236.000000
  6      IMAGE SURFACE

*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:      2.000000    Image axial ray slope:   2.0000e-20
   Object num. aperture:    2.0000e-20    F-number:                    --
   Image num. aperture:     2.0000e-20    Working F-number:        2.5000e+19
 FIELD
   Field angle:              -35.376895    Object height:           7.1006e+19
   Gaussian image height:   7.1006e+19    Chief ray ims height:     -9.451742
 CONJUGATES
   Object distance:         1.0000e+20    Srf 1 to prin. pt. 1:        --
   Gaussian image dist.:   -1.0000e+20    Srf 5 to prin. pt. 2:        --
   Overall lens length:      47.500000    Total track length:      1.0000e+20
   Paraxial magnification:    1.000000    Srf 5 to image srf:       10.000000
 OTHER DATA
   Entrance pupil radius:     2.000000    Srf 1 to entrance pup.:      --
   Exit pupil radius:         2.000000    Srf 5 to exit pupil:      -3.311258
   Lagrange invariant:        1.420113    Petzval radius:          1.0000e+40
   Effective focal length:      --
 Note: This optical system contains special surface data.
       Calculations based on a paraxial raytrace may be invalid.
```



We select the incident polarization state by setting the polarization operating conditions to define linearly polarized light, oriented at 45° to the *x* and *y* axes.

```
*OPERATING CONDITIONS: POLARIZATION
   Use polarization raytrace:       On    Degree of polarization:    1.000000
   Ellipse axes ratio:              --    Y to major axis angle:    45.000000
   Handedness of ellipse:        Right    Use 1/4 wave MgF2 coating:      Off
Now we trace a ray from full-field (i.e., an angle of incidence of 54.623° on
surfaces 3 and 4) and observe the state of polarization as the ray passes through
the rhomb.
*TRACE REFERENCE RAY
         FBY          FBX          FBZ
       1.000000       --           --
        FYRF         FXRF          FY           FX
         --           --           --           --
         YCA          XCA          YFSA         XFSA         OPL
         --           --      -8.1536e-18  -8.1536e-18   58.264253

*TRACE RAY - LOCAL COORDINATES
 SRF        Y/L          X/K          Z/M       YANG/IANG    XANG/RANG      D/OPL
        INTENSITY   DEG. POLRZ.  ELL. RATIO   ELL. ANGLE    HANDEDNESS
  1         --           --           --       -35.376895       --           --
```

```
        -0.578952      --        0.815361   35.376895   35.376895      --
         1.000000    1.000000      --       45.000000      --

  2        --          --          --          --          --          --
           --          --        1.000000      --          --          --
         0.958715    1.000000      --       45.000000      --
  3      5.458304      --      8.8818e-16  -54.623105      --        8.636288
        -0.815361      --        0.578952   54.623105   54.623105   13.040795
         0.958715    1.000000    0.414214   44.999999      LEFT =   -1.000000
  4     -8.625087      --          --      -125.376895      --       17.272577
        -0.815361      --       -0.578952  -54.623105  -54.623105   39.122386
         0.958715    1.000000    1.000000  -27.956108      LEFT =   -1.000000
  5      1.833416      --          --          --          --        6.054216
           --          --        1.000000      --          --       48.264253
         0.919134    1.000000    1.000000  -27.956110      LEFT =   -1.000000

  6      1.833416      --          --          --          --       10.000000
           --          --        1.000000      --          --       58.264253
         0.919134    1.000000    1.000000  -27.956110      LEFT =   -1.000000
 PUPIL      FY          FX                                            OPD
           --          --                                          1.4211e-11
```

After the first total internal reflection (surface 3), the light is elliptically polarized; the ratio of the axes of the polarization ellipse is 0.414. After the second reflection (surface 4), the light is circularly polarized; the ratio of the axes is 1.0. The wave is normally incident on the end faces of the rhomb (surfaces 2 and 5), so the polarization state is not changed at these surfaces, but some of the incident intensity is lost due to reflection losses of about 4% at each surface.

The Fresnel rhomb can also be used “in reverse.” If circularly polarized light is incident upon the rhomb, linearly polarized light exits the rhomb.

```
*OPERATING CONDITIONS: POLARIZATION
   Use polarization raytrace:       On    Degree of polarization:    1.000000
   Ellipse axes ratio:        1.000000    Y to major axis angle:       --
   Handedness of ellipse:         Left    Use 1/4 wave MgF2 coating:      Off

*TRACE REFERENCE RAY
        FBY         FBX         FBZ
       1.000000     --          --
        FYRF        FXRF        FY          FX
         --          --          --          --
         YCA         XCA         YFSA        XFSA        OPL
         --          --     -8.1536e-18 -8.1536e-18   58.264253

*TRACE RAY - LOCAL COORDINATES
 SRF        Y/L         X/K         Z/M      YANG/IANG   XANG/RANG      D/OPL
          INTENSITY  DEG. POLRZ. ELL. RATIO  ELL. ANGLE   HANDEDNESS
  1         --          --          --       -35.376895      --          --
          -0.578952      --        0.815361   35.376895   35.376895      --
           1.000000    1.000000    1.000000    3.925177      LEFT =   -1.000000

  2         --          --          --          --          --          --
            --          --        1.000000      --          --          --
          0.958715    1.000000    1.000000    3.925177      LEFT =   -1.000000
  3       5.458304      --      8.8818e-16  -54.623105      --        8.636288
         -0.815361      --        0.578952   54.623105   54.623105   13.040795
          0.958715    1.000000    0.414214  -45.000001      LEFT =   -1.000000
  4      -8.625087      --          --      -125.376895      --       17.272577
         -0.815361      --       -0.578952  -54.623105  -54.623105   39.122386
          0.958715    1.000000  4.9216e-16  -44.999999     RIGHT =    1.000000
  5       1.833416      --          --          --          --        6.054216
            --          --        1.000000      --          --       48.264253
          0.919134    1.000000  4.9216e-16  -45.000001     RIGHT =    1.000000

  6       1.833416      --          --          --          --       10.000000
            --          --        1.000000      --          --       58.264253
          0.919134    1.000000  4.9216e-16  -45.000001     RIGHT =    1.000000
 PUPIL      FY          FX                                            OPD
           --          --                                          1.4211e-11
```

Wollaston Prism

OSLO Premium Edition has the capability of tracing rays through uniaxial birefringent materials such as calcite. Two types of waves (and rays) can propagate in uniaxial media; these waves are called the *ordinary* wave (or *o*-ray) and the *extraordinary* wave (or *e*-ray). Ordinary waves and rays can be traced using the same techniques as are used for ray tracing in isotropic media. Tracing extraordinary rays, however, is more complicated. The refractive index for extraordinary rays is a function of the angle of incidence. Also, for the extraordinary ray, the wave vector (the normal vector to the wavefront) is generally not in the same direction as the ray vector (the vector in the direction of energy flow, i.e., the Poynting vector).

The interaction of an electric field with a material is characterized by the permittivity (dielectric constant) ε of the material. The permittivity relates the electric field **E** to the electric displacement **D**. For the nonmagnetic materials used in optical systems, *Maxwell's relation* states that the refractive index n is equal to the square root of the permittivity, i.e., $n^2 = \varepsilon$. For isotropic materials, the permittivity is a scalar quantity (although a function of wavelength). By contrast, the permittivity of an anisotropic medium such as a crystal must be described by a tensor. In other words, $\boldsymbol{\varepsilon}$ is a 3 x 3 matrix that relates the components of **E** to the components of **D**. (The difference between the wave vector and the ray vector for the extraordinary ray is a consequence of **D** no longer being collinear with **E**.)

Since the refractive index is no longer a constant at a particular point in the material, the medium is termed *birefringent*, since the index of refraction depends on the propagation direction. For the crystal materials under consideration here, a coordinate system can be found in which only the diagonal elements of the *dielectric tensor* are non-zero. The coordinate axes of this system are called the *principal axes* and the diagonal elements of the tensor are called the *principal values* of the permittivity. For *uniaxial* media, two of the principal values are equal. For *biaxial* media, all three of the principal values are different. (Note that OSLO does not treat biaxial materials.) For a uniaxial material, the axis along which the permittivity differs is the crystal axis, i.e., the axis of symmetry of the crystal. The principal values and principal axes define the *index ellipsoid*. In order to trace rays through this uniaxial birefringent medium, then, we must specify the ordinary refractive index, the extraordinary refractive index, and the orientation of the crystal axis.

In OSLO, the data for the ordinary indices is taken from the normal glass specification for the surface. To specify that a medium is birefringent, click the glass options button for the desired surface, and select Birefringent medium from the pop-up menu. In this spreadsheet, you can specify the material that defines the extraordinary refractive indices and the orientation of the crystal axis.

The extraordinary indices may either be calculated from a catalog material, or the indices may be specified explicitly. Click the appropriate radio button to specify your choice. The orientation of the crystal axis is determined by the specification of direction numbers for the axis. The direction numbers (denoted by CAK, CAL, and CAM, which are the direction numbers in *x*, *y*, and *z*, respectively) are the Cartesian components of a vector in the direction of the crystal axis. If the direction numbers are normalized (i.e., the magnitude of the vector is unity), the direction numbers are the direction cosines of the crystal axis. It is not necessary, however, to enter the direction numbers in normalized form. For example, if the crystal axis is parallel to the *y*-axis of the surface, the direction numbers would be CAK = 0, CAL = 1, CAM = 0. (For this case, where CAK = CAM =0, CAL could be any non-zero value.) For birefringent materials, the crystal axis direction numbers may be made optimization variables and operand components.

In general, a ray incident upon a birefringent material will be split into two rays (*o* and *e*), with orthogonal linear polarizations. Since OSLO does not split rays, you must specify which ray is to be traced through the material. This designation is also made in the birefringent medium spreadsheet. By default, the ordinary ray is traced. The easiest way to see the results of tracing the other ray is to make the system a multiconfiguration system, where the configuration item is the ray that is traced through the medium (the name of the configuration item is BRY).

As mentioned above, for the extraordinary ray the wave vector and the ray vector are generally not in the same direction. Thus, we need two sets of direction cosines to characterize the propagation

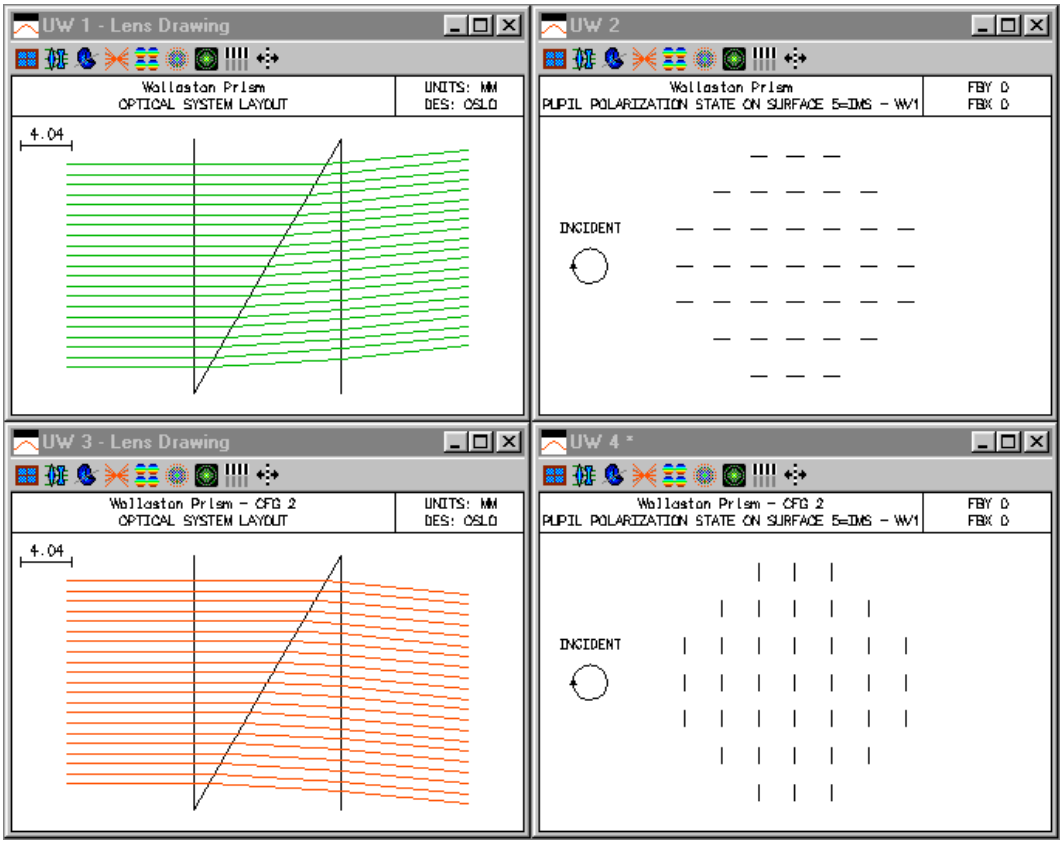
of the ray through the medium. In OSLO, the direction cosines (K, L, and M) reported in the trace_ray command (Evaluate >> Single Ray) correspond to the *ray* vector. Similarly, the RVK, RVL, and RVM operands refer to the data for the ray vector. If an extraordinary ray is being traced, the output of the trace_ray command will contain three more columns of numbers (columns 7, 8, and 9 of the spreadsheet buffer). These columns contain the values of the direction cosines for the *wave* vector. These columns are labeled LWV, KWV, and MWV, keeping with the OSLO convention of outputting the *y* value before the *x* value. If it is desired to use the wave vector direction cosines in ray operands, the components KWV, LWV, and MWV are available. For ordinary rays in birefringent media and for rays in isotropic media, the KWV, LWV, and MWV components have the same value as the RVK, RVL, and RVM components.

Five common uniaxial materials are contained in the miscellaneous glass catalog: calcite ($CaCO_3$), ADP, KDP, MgF_2, and sapphire (Al_2O_3). With the exception of the *o*-indices for calcite, the dispersion equations for these materials were taken from the *Handbook of Optics*, Volume II, Chapter 33, “Properties of Crystals and Glasses”, by W. J. Tropf, M. E. Thomas, and T. J. Harris, Table 22 (McGraw-Hill, New York, 1995). The calcite *o*-index dispersion equation data was calculated by performing a least-squares Sellmeier fit to the data in Table 24 of the reference. The RMS error of this fit is 0.000232, as compared to the tabulated values, over the wavelength range from 0.200 μm to 2.172 μm represented by the values in Table 24. (Also, several minor typographical errors in Table 22 have been corrected. The equations for calcite should be in terms of n^2 not n. The absorption wavelength in the third term for the *e*-index of MgF_2 should be 23.771995, not 12.771995.) Note that each material corresponds to two entries in the glass catalog: one for the *o*-indices (CALCITE_O, ADP_O, KDP_O, MGF2_O, SAPPHIRE_O) and one for the *e*-indices (CALCITE_E, ADP_E, KDP_E, MGF2_E, SAPPHIRE_E).

As an example of the use of birefringent materials, consider the following system, which is a Wollaston prism. This prism consists of two wedges of a birefringent material, in this case calcite. In the first wedge, the crystal axis is in the *y*-direction, while in the second wedge, the crystal axis is in the *x*-direction. With this orientation of crystal axes, an *o*-ray in the first wedge becomes an *e*-ray in the second wedge, and vice-versa. If a beam of circularly polarized light is normally incident on the prism, two beams exit the prism: one deflected upwards, with horizontal linear polarization, and the other downwards, with vertical linear polarization.

```
*LENS DATA
Wollaston Prism
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ       --        1.0000e+20    1.0000e+14            AIR

 AST       --           --         10.000000 A           AIR

  2        --          5.773503     10.000000       CALCITE_O CB
  3        --          5.773503     11.547005       CALCITE_O CB*
  4        --          3.000000     10.000000            AIR

 IMS       --           --          8.000010 S

*TILT/DECENTER DATA
  3     RCO   3
        DT    1         DCX       --       DCY       --       DCZ       --
                        TLA   -30.000000   TLB       --       TLC       --

*SURFACE TAG DATA
  3     DRW ON


*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.589290
        1.000000

*REFRACTIVE INDICES
 SRF     GLASS           RN1        TCE
  0      AIR          1.000000      --
  1      AIR          1.000000  236.000000
  2      CALCITE_O    1.658342      --
 EXTRA   CALCITE_E    1.486440
 BRY       Ordinary   Crystal Axis (K,L,M)      --        1.000000      --
  3      CALCITE_O    1.658342      --
 EXTRA   CALCITE_E    1.486440
 BRY  Extraordinary   Crystal Axis (K,L,M)    1.000000      --          --
```

```
  4      AIR           1.000000  236.000000
  5      IMAGE SURFACE

*CONFIGURATION DATA
TYPE   SN    CFG   QUALF        VALUE
BRY     2     2      0            EXT
BRY     3     2      0            ORD
```



Vector diffraction

In addition to the propagation of the state of polarization through an optical system, a polarization ray trace allows for the computation of vector diffraction patterns. Conventional diffraction analysis of optical systems makes the assumption that the optical field is a scalar quantity. Comparison with experiment shows that this is an excellent approximation for numerical apertures of less than about 0.55 or 0.60. For focusing at larger numerical apertures however, the longitudinal (z) component of the field can not be ignored. In this case, the point spread function integral [See the point spread function section of the Image Evaluation chapter.] must be computed separately for each of the three (x, y, and z) orthogonal polarization fields in the exit pupil. The resultant observed irradiance is the sum of the squared moduli of the three component fields, i.e., the electric field energy density.

In OSLO, if the polarization ray trace operating condition is on, all point spread function calculations will compute the vector diffraction pattern, based on the vector electric field in the exit pupil of the lens. Since the diffraction integral must be evaluated for each Cartesian component of the electric field, the calculation will take at least three times as long as a scalar *PSF* calculation. (If the degree of polarization is not unity, then the integral must also be computed for the orthogonal incident polarization, which means that six integrals must, in total, be evaluated.) As mentioned above, vector diffraction effects are only noticeable for large numerical aperture systems, so it is usually not necessary to do the vector calculation for familiar, low *NA* systems.

NA = 0.966 perfect lens

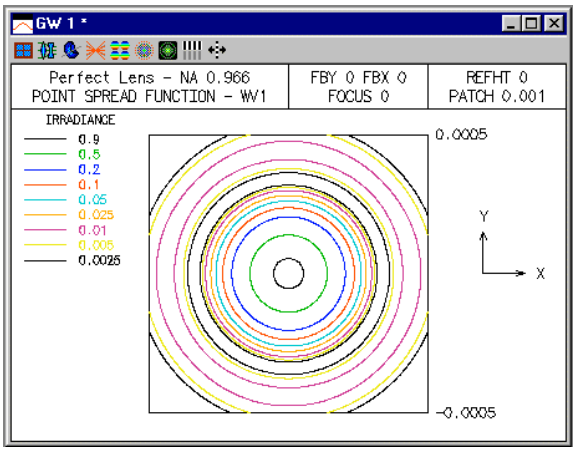
As an example of focusing at high numerical apertures, we will consider a perfect lens with a numerical aperture of 0.966 (an image space cone half-angle of 75°). Since this is a perfect lens with no aberration, the effects of the vector nature of light should be readily apparent. We enter a perfect lens with a focal length of 100 mm, object at infinity (magnification of zero), numerical aperture of 0.966 = sin(75°), and a single wavelength of 0.5 μm.

```
*LENS DATA
Perfect Lens - NA = 0.966
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        1.0000e+20    1.0000e+14            AIR

  1   ELEMENT GRP        --          96.592583 AS          AIR   *

  2       PERFECT    100.000000 S   96.592583 S           AIR   * PERFECT

  3        --            --        1.0000e-04 S

*SURFACE NOTES
  2    PERFECT

*SURFACE TAG DATA
  1     LMO EGR  (2 surfaces)
  1     DRW ON

*PERFECT LENS DATA
  2     PFL   100.000000   PFM       --

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.500000
        1.000000

*PARAXIAL SETUP OF LENS
 APERTURE
   Entrance beam radius:     96.592583    Image axial ray slope:    -0.965926
   Object num. aperture:    9.6593e-19    F-number:                  0.517638
   Image num. aperture:       0.965926    Working F-number:          0.517638
 FIELD
   Field angle:             5.7296e-05    Object height:          -1.0000e+14
   Gaussian image height:   1.0000e-04    Chief ray ims height:    1.0000e-04
 CONJUGATES
   Object distance:         1.0000e+20    Srf 1 to prin. pt. 1:        --
   Gaussian image dist.:    100.000000    Srf 2 to prin. pt. 2:        --
   Overall lens length:         --        Total track length:      1.0000e+20
   Paraxial magnification: -1.0000e-18    Srf 2 to image srf:      100.000000
 OTHER DATA
   Entrance pupil radius:    96.592583    Srf 1 to entrance pup.:      --
   Exit pupil radius:        96.592583    Srf 2 to exit pupil:         --
   Lagrange invariant:     -9.6593e-05    Petzval radius:          1.0000e+40
```

```
   Effective focal length:  100.000000
 Note: This optical system contains special surface data.
       Calculations based on a paraxial raytrace may be invalid.
```

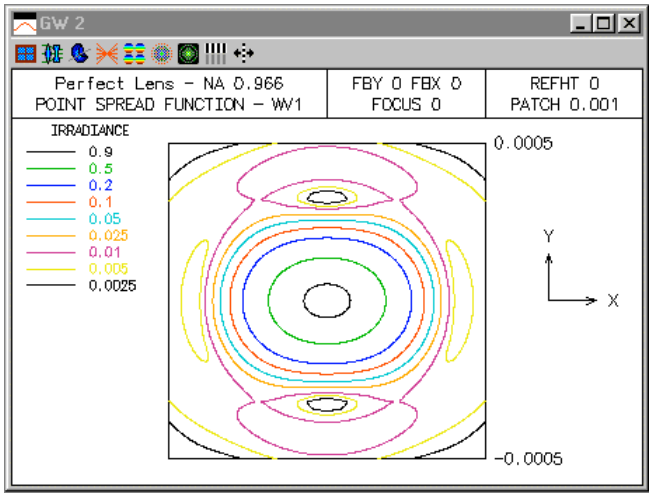
On-axis, this is a completely rotationally symmetric system, so our intuition tells us that we should expect the point spread function to also be rotationally symmetric. Computation of the usual (scalar) point spread function confirms this. Because of the large numerical aperture, we need to include the effects of non-uniform pupil amplitude, caused by the transformation from a planar wavefront to a spherical wavefront. We do this by using equal image space ray increments in spot diagram related calculations.

```
*OPERATING CONDITIONS: SPOT DIAGRAM
   Aperture divisions:       40.000000    Use Gaussian pupil apodization: Off
   X 1/e^2 entr. irrad.:      1.000000    Y 1/e^2 entr. irrad.:      1.000000
   Use all wavelengths in diagram:  On    P-V OPD for MTF switch:    3.000000
   Use equal image space incrmnts.: On    Through-foc. frequency:   25.000000
   Diffraction efficiency calcs.:  Off
```



Now we want to consider the effect on the point spread function if the incident wave is linearly polarized. We choose the polarization to be in the *x* direction and recompute the *PSF*.

```
*OPERATING CONDITIONS: POLARIZATION
   Use polarization raytrace:       On    Degree of polarization:    1.000000
   Ellipse axes ratio:              --    Y to major axis angle:    90.000000
   Handedness of ellipse:        Right    Use 1/4 wave MgF2 coating:      Off
```



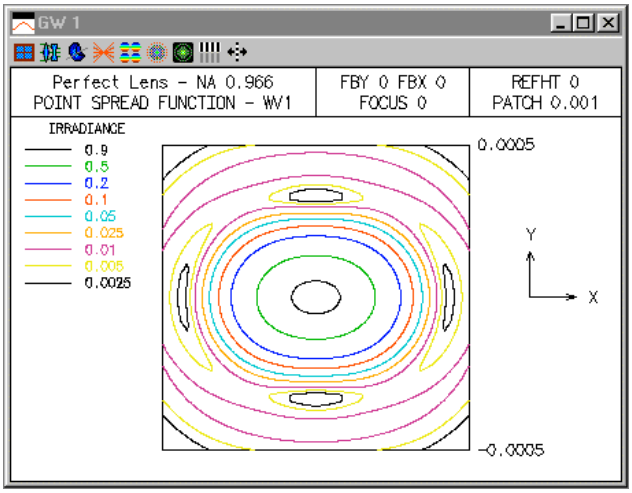
From the figure, we see that the *PSF* is no longer rotationally symmetric, even though the optical system has rotationally symmetry, except for the polarization. The *PSF* exhibits the well-known characteristic of being narrower in the azimuth that is perpendicular to the polarization direction. In this case, the polarization direction is *x*, so the *PSF* is narrower in *y*. This irradiance distribution has different effective "spot sizes" in *x* and *y*, so the polarization orientation is important when computing quantities such as two-point resolution for high *NA* systems.

We can examine the *x* and *y* components of the point spread function by inserting a linear polarizer after the perfect lens.

```
*LENS DATA
Perfect Lens - NA = 0.966
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE    NOTE
  0         --        1.0000e+20    1.0000e+14            AIR

  1   ELEMENT GRP         --          96.592583 AS          AIR    *

  2       PERFECT         --          96.592583 S           AIR    * PERFECT

  3         --        100.000000      96.592583 S           AIR    *

  4         --            --          1.0000e-04 S
```

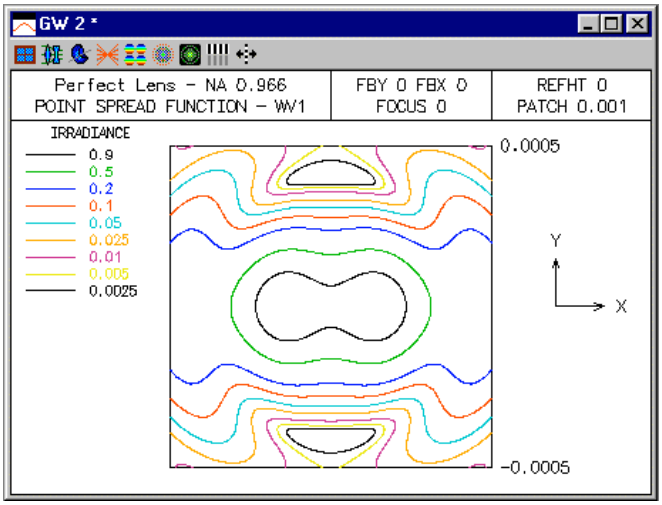
- *x*-axis polarizer

```
*POLARIZATION ELEMENT DATA
               AMPLITUDE      PHASE              AMPLITUDE      PHASE
  3      JA     1.000000       --          JB       --           --
         JC       --           --          JD       --           --
```

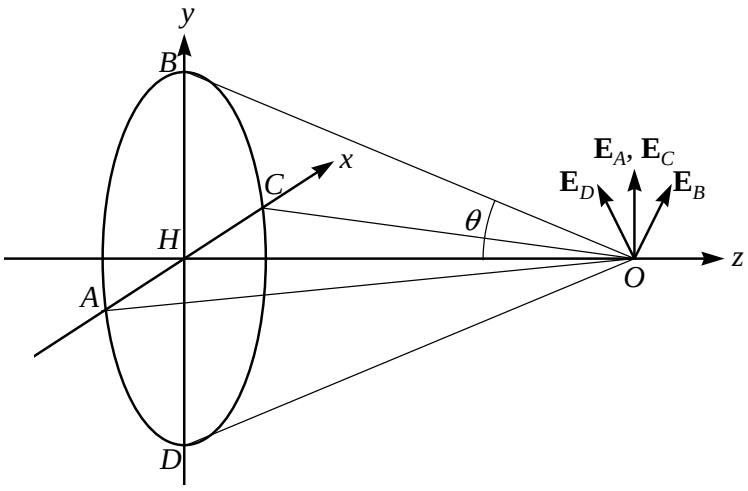


- *y*-axis polarizer

```
*POLARIZATION ELEMENT DATA
               AMPLITUDE      PHASE              AMPLITUDE      PHASE
  3      JA       --           --          JB       --           --
         JC       --           --          JD     1.000000       --
```



A qualitative explanation of this effect has been given by Hopkins(10). With reference to the figure below, ABCD is an annulus of the spherical wave that is converging to the focus O. The incident wave is polarized in the HB direction, i.e., along the y-axis. The electric field contributions from A, B, C, and D are EA, EB, EC, and ED, respectively. EA and EC have no axial (z) component, while EB and ED have axial components that are opposite in direction, and thus cancel. In the direction of EA and EC, EB and ED have effective magnitudes EBcos and EDcos , where is the angular half angle of ABCD. Thus, relative to the x direction, the amplitude of the resultant field in the y direction is diminished by the cos factor. The effective amplitude in the pupil is then larger along the x-axis than it is along the y-axis. Since there is more energy in the outer portion of the pupil in x as compared to y, the diffraction spot is smaller in the x direction, i.e., orthogonal to the direction of the incident polarization.



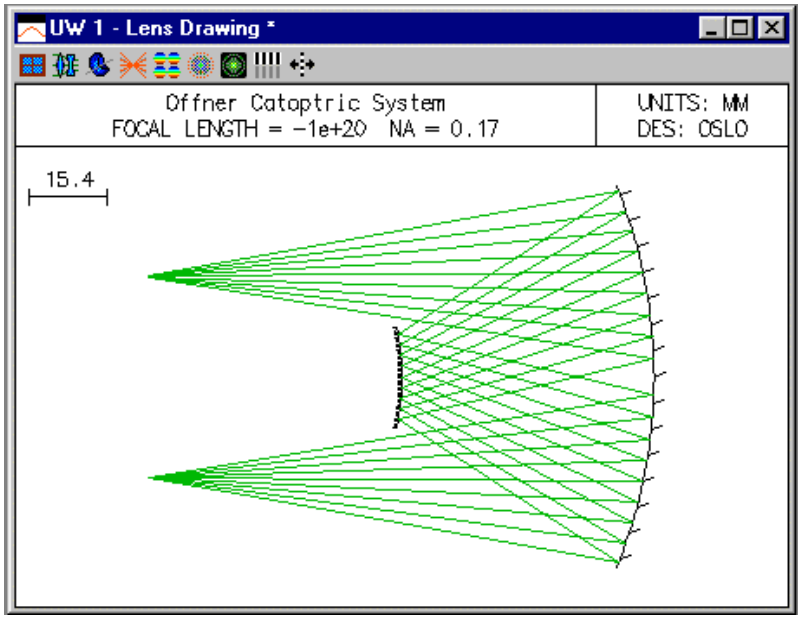
10 H. H. Hopkins, “Resolving power of the microscope using polarized light,” Nature **155**, 275 (1945).

Partial Coherence

Offner catoptric system

As an example of the effects of coherence on imaging, we will use a two-mirror, monocentric system of the type originally designed by Offner (U.S. Patent 3,748,015). For an object placed in the plane containing the common centers of curvature, the imagery is 1:1 and all of the Seidel aberrations are zero. This type of system has been widely used in photolithographic systems. The radius of curvature of the large, concave mirror is twice the radius of curvature of the small, convex mirror. The aperture stop is located at the small mirror so this system is essentially telecentric. We start with the following system and use the mercury *i*-line at 0.365 μm.

```
*LENS DATA
Offner Catoptric System
 SRF      RADIUS      THICKNESS    APERTURE RADIUS       GLASS  SPE   NOTE
  0        --        100.000000     20.000000             AIR

  1   -100.000000    -50.000000     38.000000       REFL_HATCH

  2    -50.000000     50.000000     10.000000 A     REFL_HATCH

  3   -100.000000   -100.000000     38.000000          REFLECT

  4        --            --         20.000000 S

*PARAXIAL SETUP OF LENS
APERTURE
   Object num. aperture:      0.170000    F-number:                    --
FIELD
   Gaussian image height:   -20.000000    Chief ray ims height:     20.000000

*WAVELENGTHS
CURRENT  WV1/WW1
   1    0.365010
```



Obviously, because of the location of the secondary mirror, this system is only used with off-axis object points. In this nominal design, the performance is limited by fifth-order astigmatism. If the separation of the mirrors is changed slightly, a small amount of third-order astigmatism can be introduced and the third-order and fifth-order astigmatism can be made to balance at one object height. Thus, the resulting system has a single (object) zone of good correction and can be used as a "ring-field" system (i.e., a field of view in the shape of a section of an annulus or ring). In order to make this modification to the lens, we first enter minus thickness pickups for surfaces 2 and 3, in order to maintain the desired system geometry. Also, we make thicknesses 0 and 1 variable.

```
*LENS DATA
Offner Catoptric System
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE   NOTE
  0        --        100.000000 V   20.000000             AIR

  1   -100.000000    -50.000000 V   38.000000      REFL_HATCH

  2    -50.000000     50.000000 P   10.000000 A    REFL_HATCH

  3   -100.000000   -100.000000 P   38.000000         REFLECT

  4        --            --         20.000000 S

*PICKUPS
  2     THM    1
  3     THM    0

*VARIABLES
 VB   SN  CF  TYP       MIN         MAX        DAMPING      INCR        VALUE
V 1    0   -  TH      0.100000  1.0000e+04    1.000000    0.001725   100.000000
V 2    1   -  TH   -1.0000e+04   -0.100000    1.000000    0.001725   -50.000000
```

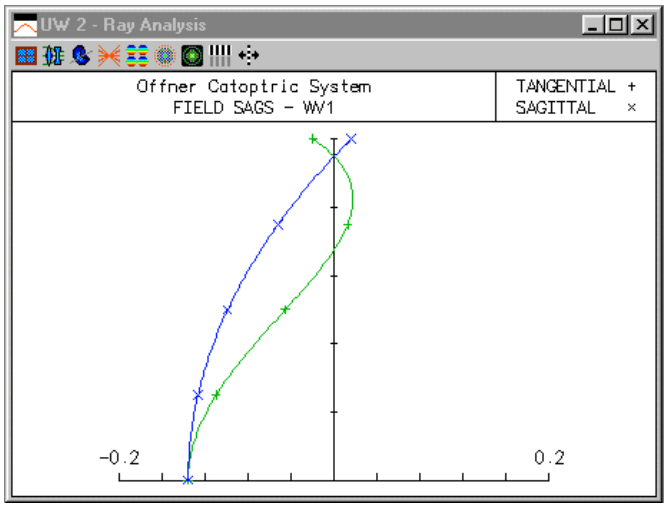
We could do the optimization in several ways, but the simplest is probably to use OSLO's automatic error function generation to create an error function that measures the RMS OPD at the selected object point. We choose to balance the astigmatism at a fractional object height of 0.95 (i.e., an object height of –19.0 mm). With this field point, the result of using the error function generator is

```
*RAYSET
FPT       FBY/FY1       FBX/FY2       FBZ/FX1       YRF/FX2       XRF/WGT
F 1       0.950000        --            --            --            --
         -1.000000      1.000000     -1.000000      1.000000      1.000000
RAY         TYPE          FY            FX            WGT
R 1       Ordinary        --            --          0.041667
R 2       Ordinary      0.525731        --          0.208333
R 3       Ordinary      0.262866      0.455296      0.208333
R 4       Ordinary     -0.262866      0.455296      0.208333
R 5       Ordinary     -0.525731        --          0.208333
R 6       Ordinary      0.850651        --          0.208333
R 7       Ordinary      0.425325      0.736685      0.208333
R 8       Ordinary     -0.425325      0.736685      0.208333
R 9       Ordinary     -0.850651        --          0.208333
R 10      Ordinary      1.000000        --          0.041667
R 11      Ordinary      0.500000      0.866025      0.041667
R 12      Ordinary     -0.500000      0.866025      0.041667
R 13      Ordinary     -1.000000        --          0.041667

*OPERANDS
 OP    DEFINITION                  MODE     WGT     NAME          VALUE   %CNTRB
O 15   "RMS"                         M    0.500000 Orms1        2.771379 100.00
MIN ERROR:     2.771379
```

After a few iterations, the resulting system is as given below.

```
*LENS DATA
Offner Catoptric System
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS SPE   NOTE
  0        --        100.870668 V   20.000000             AIR

  1   -100.000000    -49.078862 V   38.000000      REFL_HATCH

  2    -50.000000     49.078862 P   10.000000 A    REFL_HATCH

  3   -100.000000   -100.870668 P   38.000000         REFLECT

  4        --            --         20.023377 S
```

The field curves indicate that the desired astigmatism balance has been achieved.

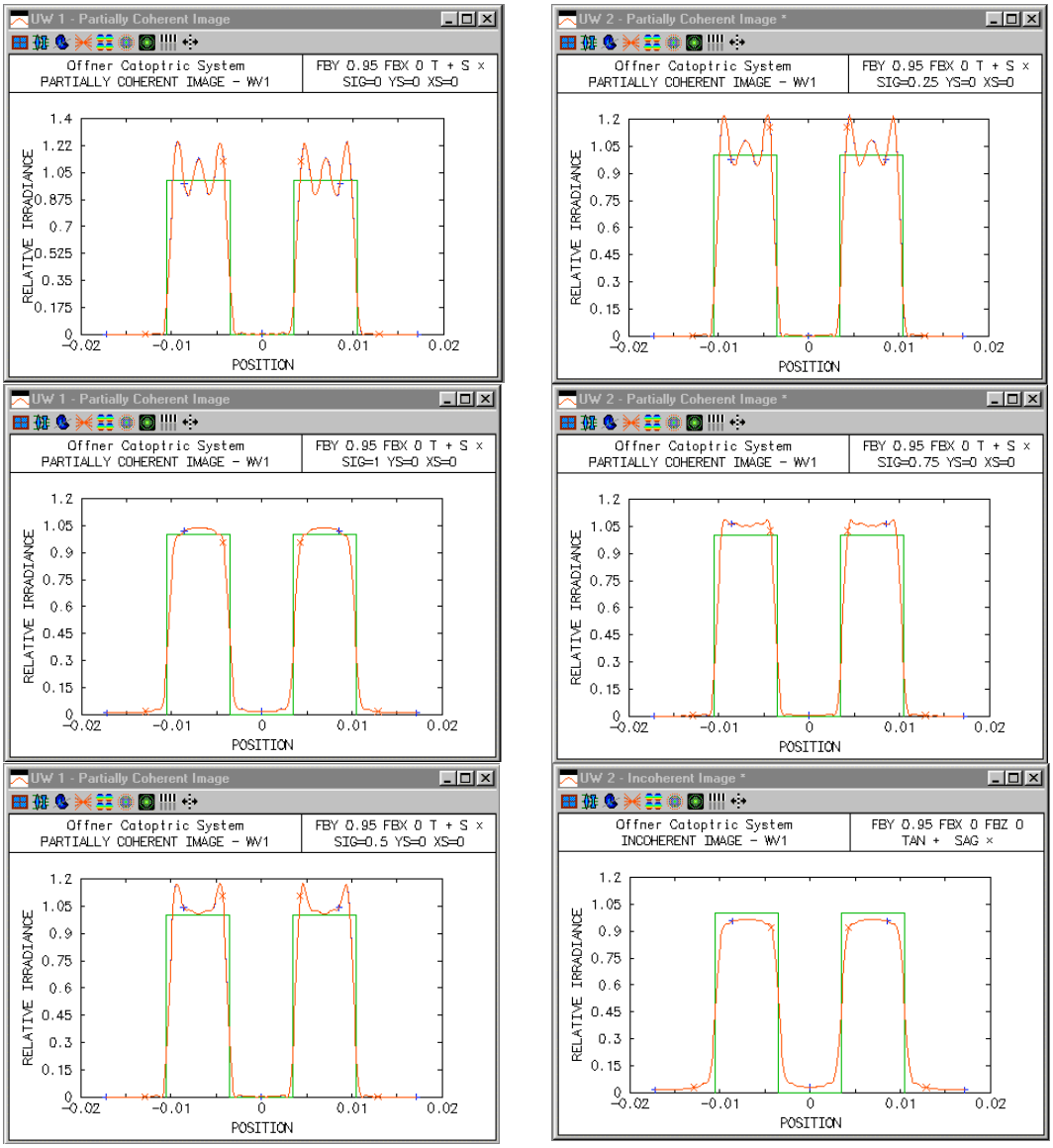


The numerical aperture of this lens is 0.17, so the diameter of the Airy disk is 1.22 λ_0 / *NA* = 1.22 (0.365 μm) / 0.17 = 2.62 μm. Thus a perfect image bar width of 7 μm should be easily resolved and be suitable to demonstrate coherence effects. The optimized lens is essentially diffraction limited at the design field of 0.95, so the resulting image at this object point will be indicative of the effects of coherence and diffraction.

In the partial coherence operating conditions, we define the ideal image to consist of two bars, each of width 7 μm and separated by 14 μm.

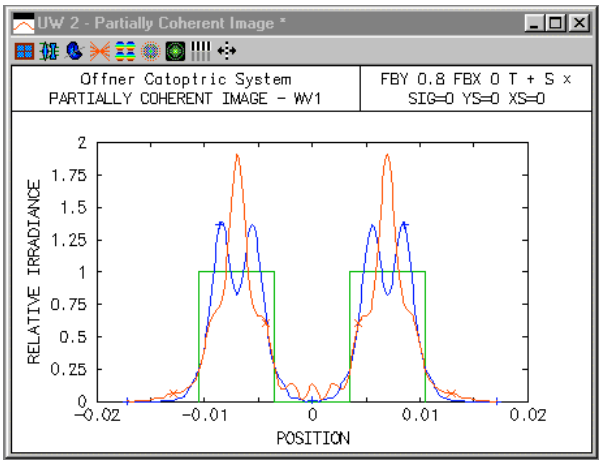
```
*OPERATING CONDITIONS: PARTIAL COHERENCE
   Effective source rad.:       --        Inner annular radius:        --
   X shift of source:           --        Y shift of source:           --
   X 1/e^2 of source:           --        Y 1/e^2 of source:           --
   Number of points in image:       64    Number of clear bars in image:    2
   Width of clear bar:        0.007000    Period of clear bars:      0.014000
   Irrad. between bars:         --        Phase between bars:          --
   Background irradiance:       --
   Normalization:    Object irradiance    Use equal image space incrmnts.:Off
```

We will examine the image as we change the illumination from a point effective source (i.e., fully coherent; $\sigma = 0$) to an effective source that completely fills the entrance pupil ($\sigma = 1$). We also examine the incoherent limit.

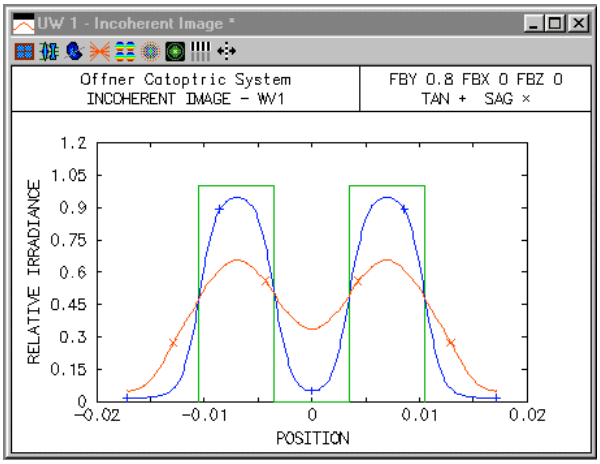

UW 1 - Partially Coherent Image
Offner Catoptric System
PARTIALLY COHERENT IMAGE - WV1
FBY 0.95 FBX 0 T + S x
SIG=0 YS=0 XS=0
RELATIVE IRRADIANCE
POSITION
UW 2 - Partially Coherent Image *
Offner Catoptric System
PARTIALLY COHERENT IMAGE - WV1
FBY 0.95 FBX 0 T + S x
SIG=0.25 YS=0 XS=0
RELATIVE IRRADIANCE
POSITION
UW 1 - Partially Coherent Image
Offner Catoptric System
PARTIALLY COHERENT IMAGE - WV1
FBY 0.95 FBX 0 T + S x
SIG=1 YS=0 XS=0
RELATIVE IRRADIANCE
POSITION
UW 2 - Partially Coherent Image *
Offner Catoptric System
PARTIALLY COHERENT IMAGE - WV1
FBY 0.95 FBX 0 T + S x
SIG=0.75 YS=0 XS=0
RELATIVE IRRADIANCE
POSITION
UW 1 - Partially Coherent Image
Offner Catoptric System
PARTIALLY COHERENT IMAGE - WV1
FBY 0.95 FBX 0 T + S x
SIG=0.5 YS=0 XS=0
RELATIVE IRRADIANCE
POSITION
UW 2 - Incoherent Image *
Offner Catoptric System
INCOHERENT IMAGE - WV1
FBY 0.95 FBX 0 FBZ 0
TAN + SAG x
RELATIVE IRRADIANCE
POSITION

We see that as the coherence decreases, the “interference-like” ringing of the edges of image decreases. In photolithography, it is usually the slopes of the edges of the image that are of interest; higher slopes lead to smaller changes in linewidth with changes in exposure. As the above figures indicate, in addition to controlling the aberrations of the imaging lens, the illumination coherence (i.e., the value of σ) must be considered when calculating overall system performance. If we look at the structure of the image for a fractional object height of 0.8 (i.e., an image height of 16 mm), we see the effects of the astigmatism on the coherent and incoherent images.

- Coherent



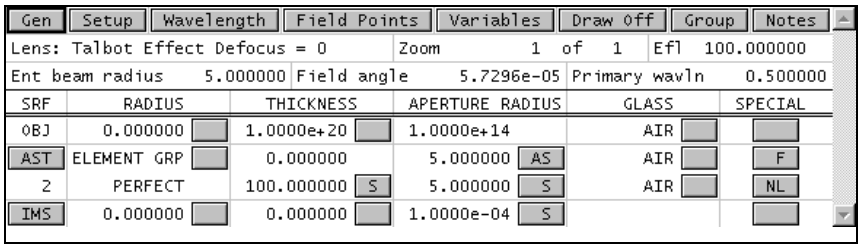
- Incoherent



Talbot effect

A striking example of the influence of coherence upon imaging is provided by the *Talbot effect*. If a coherent field has a periodic spatial amplitude distribution, the propagating field exhibits self-imaging, i.e., the image replicates itself at prescribed longitudinal distances. Compare this to the familiar case of incoherent illumination, where, in general, the modulation of an image decreases as we move the observation plane longitudinally from focus.

As a simple demonstration, we start with a 100 mm focal length perfect lens of numerical aperture 0.05 and monochromatic illumination of wavelength 0.5 μm.



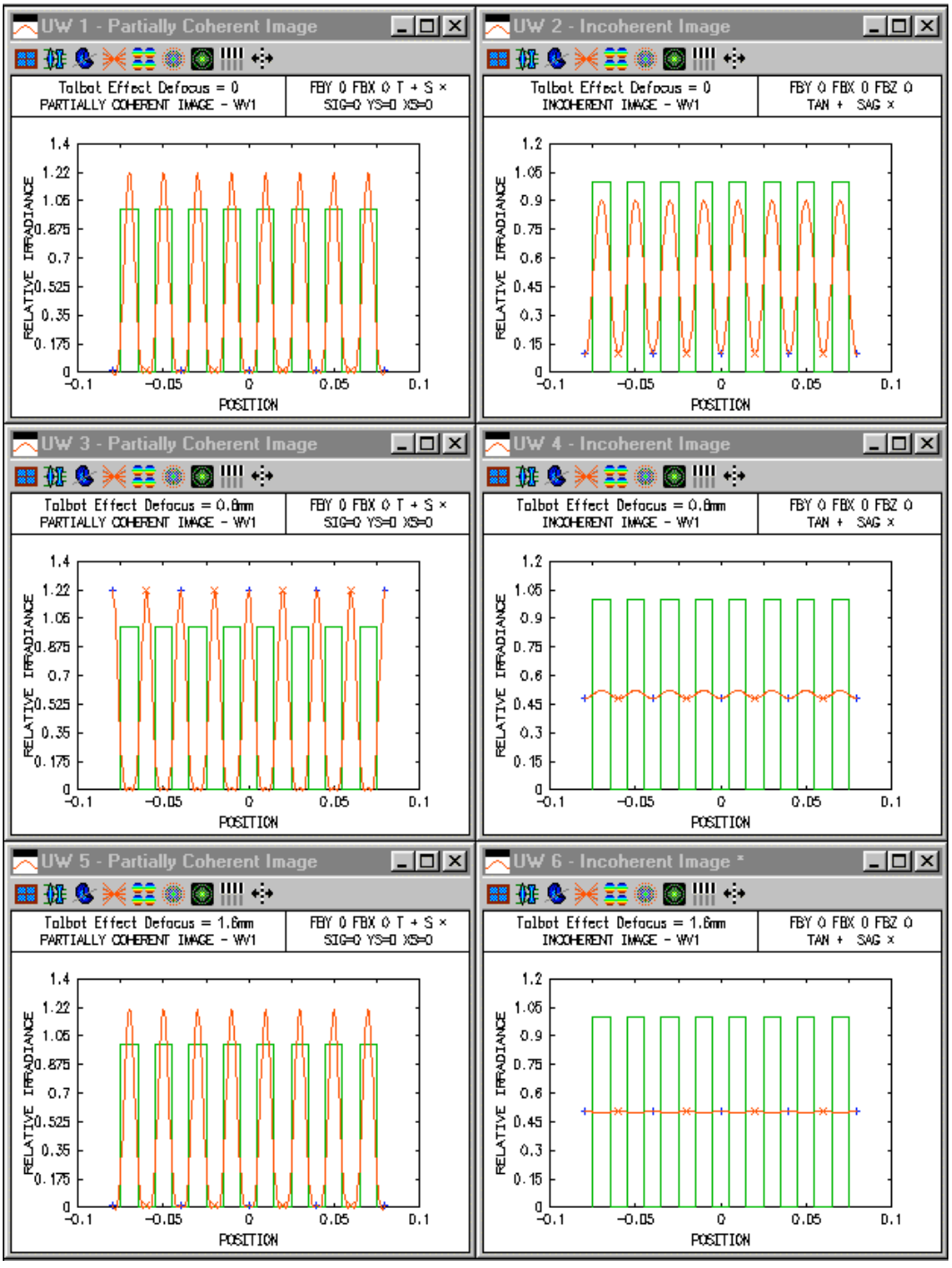
We will use a perfect image that consists of an infinite pattern of equal width bars and spaces, with a fundamental period of 20 μm. If we use an FFT size of 64 points, Eq. **(7.62)** indicates that the size of the image patch for this lens is (64)(0.5 μm)/(4*0.05) = 160 μm. Thus, if we specify that the ideal image has 8 or more bars, the ideal image is effectively an infinite square wave, of period 20 μm. (The infinite periodicity is a result of using the FFT algorithm, which implicitly produces the output for one cycle of an infinite, periodic object.)

```
*OPERATING CONDITIONS: PARTIAL COHERENCE
   Effective source rad.:       --        Inner annular radius:        --
   X shift of source:           --        Y shift of source:           --
   X 1/e^2 of source:           --        Y 1/e^2 of source:           --
   Number of points in image:       64    Number of clear bars in image:    8
   Width of clear bar:        0.010000    Period of clear bars:      0.020000
   Irrad. between bars:         --        Phase between bars:          --
   Background irradiance:       --
   Normalization:    Object irradiance    Use equal image space incrmnts.:Off
```

We can now evaluate the in-focus images for both coherent and incoherent light. As expected, there is some ringing of the edges in the coherent image, while the incoherent images exhibits a decrease in modulation from the unit-modulation object. For an object period of p and wavelength λ_0, the Talbot distance is given by

$$d_{\text{Talbot}} = \frac{p^2}{\lambda_0} \qquad (10.88)$$

In this case, the Talbot distance is d_{Talbot} = (0.02 mm)^2/(0.0005 mm) = 0.8 mm. The coherent and incoherent images with focus shifts of 0.8 mm and 1.6 mm are shown below. In general, the coherent image replicates itself at integer multiples of the Talbot distance, and is also shifted laterally by one-half period if the integer is odd. With 0.8 mm of defocus, the incoherent image is virtually nonexistent (there are about 2 waves of defocus), but the coherent image is essentially identical to the in-focus image, except that it is shifted laterally by one-half of a period. If we examine the coherent image at two Talbot distances (1.6 mm) from focus, we see that the coherent image is the same as the nominal, in-focus image, while the incoherent image is gone.



We can also examine the incoherent in-focus image of the square wave using the modulation transfer function. The image modulation of a square wave of frequency f_0 can be computed by resolving the square wave into its Fourier (i.e., sine wave) components and using the *MTF* value for each sine wave frequency. The resulting square wave modulation $S(f_0)$ is given by

$$S(f_0) = \frac{4}{\pi}\left[MTF(f_0) - \frac{1}{3}MTF(3f_0) + \frac{1}{5}MTF(5f_0) + \cdots\right] \qquad (10.89)$$

(Equation (10.89) can be found in, for example, Smith(11). For this lens, the cutoff frequency is 2 NA/λ_0 = 200 cycles/mm. Since our square wave has a frequency of f_0 = 1/0.02 mm = 50 cycles/mm, only the f_0 and $3f_0$ terms are non-zero in Eq. (10.89). We need to compute the on-axis *MTF* with a frequency increment of 50 cycles/mm. The number of aperture divisions in the spot diagram has been set to 32, so that the pupil sampling is the same as the partial coherence calculations.

```
*MODULATION TRANSFER FUNCTION Y
 WAVELENGTH 1
 NBR   FREQUENCY     MODULUS      PHASE               DIFF LIM MTF
  1       --        1.000000      --                    1.000000
  2    50.000000    0.684729      --                    0.684729
  3   100.000000    0.394089      --                    0.394089
  4   150.000000    0.147783      --                    0.147783
  5   200.000000      --          --                      --
CUTOFF FREQUENCY  193.654321
```

Using the above and Eq. (10.89), we find that the square wave modulation is S(50 cycles/mm) = $(4/\pi)(0.685 – 0.148/3) = 0.81$. To compare this with the output of the incoherent image calculation, we print out the incoherent image irradiance with an image plane increment of 0.01 mm, so that the minimum and maximum irradiance values are displayed.

```
*INCOHERENT IMAGE: MONOCHROMATIC
 WAVELENGTH 1
 NBR       Y       IRRADIANCE
  1   -0.080000     0.100449
  2   -0.070000     0.899551
  3   -0.060000     0.100449
  4   -0.050000     0.899551
  5   -0.040000     0.100449
  6   -0.030000     0.899551
  7   -0.020000     0.100449
  8   -0.010000     0.899551
  9       --        0.100449
 10    0.010000     0.899551
 11    0.020000     0.100449
 12    0.030000     0.899551
 13    0.040000     0.100449
 14    0.050000     0.899551
 15    0.060000     0.100449
 16    0.070000     0.899551
 17    0.080000     0.100449
```

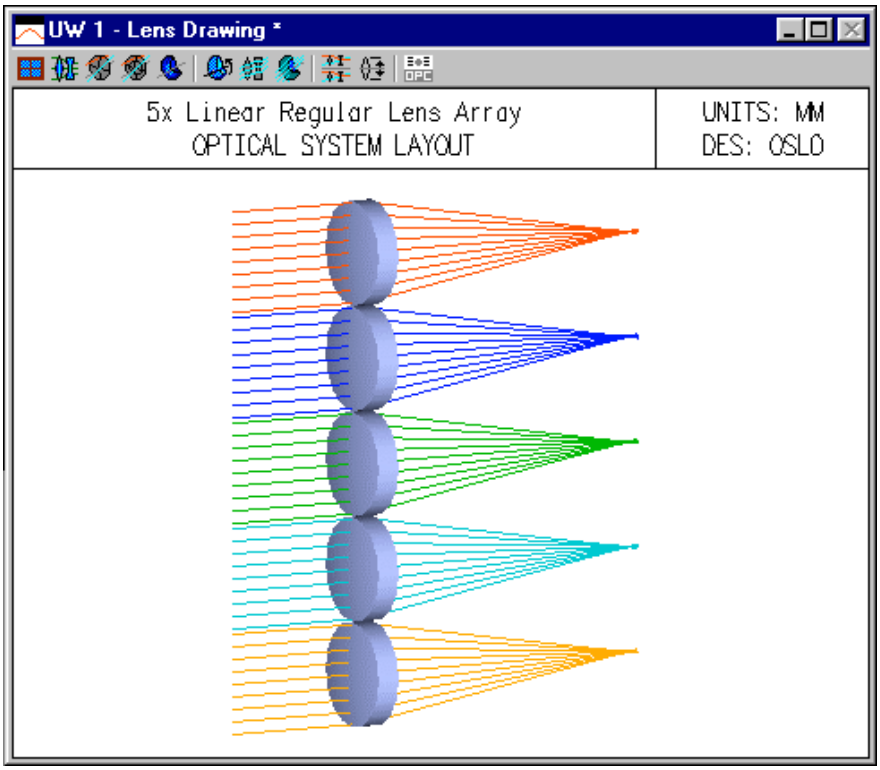
Using the minimum (I_{min} = 0.100449) and maximum (I_{max} = 0.899551) irradiance values, the computed modulation is $S = (I_{max} – I_{min})/(I_{max} + I_{min}) = 0.80$, very close to the square wave modulation value given above, which was computed using a completely different technique.

11 W. J. Smith, *Modern Optical Engineering,* Second Edition, McGraw-Hill 1990, p. 355.

Array ray tracing

Regular array

There are many applications of lens arrays, ranging from micro-optics switching systems to multiple mirror telescopes. The following example shows a simple system comprising a 5-element linear array of lenses, set up as a regular array.



Gen	Setup	Wavelength	Field Points	Variables	Draw Off	Surfs	Notes
Lens: 5x Linear Regular Lens Array		Zoom 1 of 1		Efl 29.527797			
Ent beam radius 5.000000		Field angle 5.7296e-05		Primary wavln 0.587562			

SRF	RADIUS		THICKNESS		APERTURE RADIUS		GLASS		SPECIAL
OBJ	0.000000		1.0000e+20		1.0000e+14		AIR		
AST	0.000000		0.000000		25.000000	A	AIR		F
2	30.000000		3.000000		20.000000	X	BK7	C	
3	-30.000000		28.521734	S	20.000000	X	AIR		C
IMS	0.000000		3.000000		20.000000				

The array data is entered using SPECIAL>>Surface Control>>Regular Lens Array. Since there is a single row of lenses, the *x* spacing is 0. The number of lenses is controlled by the aperture of the channel surface (surface 1). Only the vertex of each channel needs to be within the aperture of the channel surface to be included in the array, although here the aperture has been set to enclose the entire array surface.

```
*LENS ARRAY DATA
 SRF 1:
 TYPE Regular              END SURF 3                DRAW ALL CHANNELS: Yes
 X SPACING      --         Y SPACING   10.000000     Y OFFSET      --
```

The aperture of the elements themselves are determined by rectangular special apertures on surface 2 and 3:

```
*APERTURES
 SRF   TYPE APERTURE RADIUS
  2     SPC    20.000000
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -5.000000  AX2     5.000000  AY1    -5.000000  AY2     5.000000

  3     SPC    20.000000
     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -5.000000  AX2     5.000000  AY1    -5.000000  AY2     5.000000
```

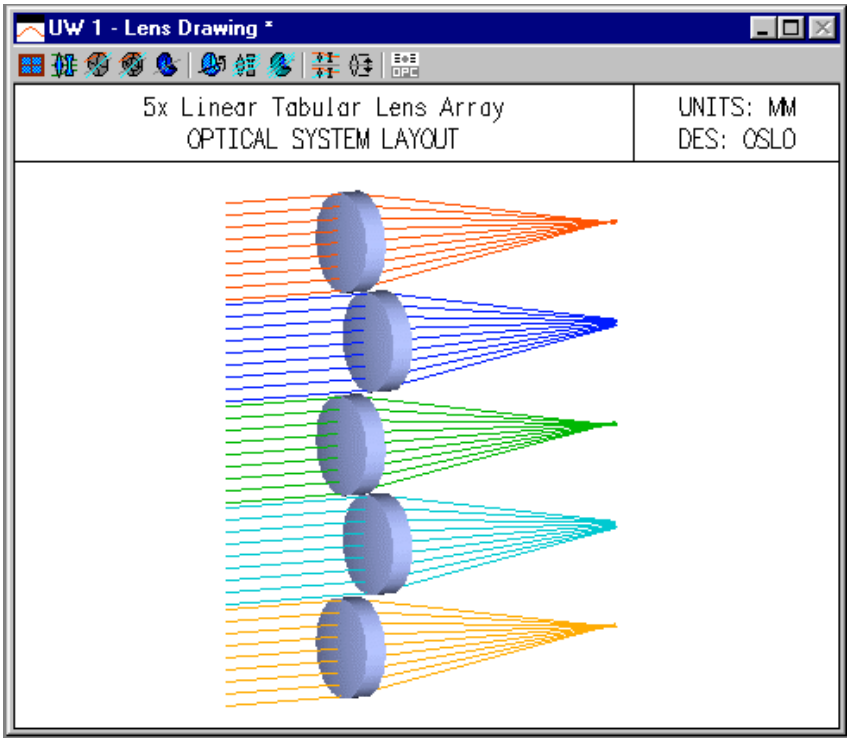
The system shown here has five light sources. In OSLO, these are modeled as separate field points. The required lens drawing conditions (non-default only) are shown below.

```
*CONDITIONS: LENS DRAWING
   Drawn apertures (solid):       Full    Image space rays:         Image srf
   Number of field points (rays):    5    DXF/IGES file view:     Unconverted
   Fpt Frac Y Obj Frac X Obj Rays Min Pupil  Max Pupil    Offset  Fan Wvn Cfg
    1     1.00000      --       9   -0.95000    0.95000      --     Y   1   0
    2     1.00000      --       9    1.05000    2.95000      --     Y   1   0
    3     1.00000      --       9    3.05000    4.95000      --     Y   1   0
    4     1.00000      --       9   -2.95000   -1.05000      --     Y   1   0
    5     1.00000      --       9   -4.95000   -3.05000      --     Y   1   0
```

Since a spot diagram pertains to a single field point, the data obtained for an array of the type shown here may not be what is desired, and it may be preferable to construct custom CCL commands to carry out evaluation that is tailored to the system at hand. Please note that since lens arrays use **rco** (return coordinates) surfaces, paraxial analysis will not be correct. In the system here, a 3mm image focus shift has been added to the paraxial solve value, to make up for the thickness of the array elements.

Tabular array

This example shows a modification the preceding regular array, to make a tabular array. Two of the elements have been offset to illustrate the difference between the two types.



The main surface data spreadsheet is identical to the one for the regular array. The difference is in the array data spreadsheet (SPECIAL>>Surface Control>>Tabular Lens Array), which enumerates the coordinates of the vertices of each element (channel) in the array. Note that a *z* displacement has been added to elements 2 and 3. This is not accounted for in the above drawing, which shows rays traced to the nominal image surface, from a field point 10 degrees off axis.

Surface 1 — Delete Lens Array

Array type: Tabular Number of channels: 5 Draw all channels: (•) Yes ( ) No

End surface: 3

CH NBR	X CTR	Y CTR	Z CTR	TLA	TLB	TLC
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	10.000000	3.000000	0.000000	0.000000	0.000000
3	0.000000	-10.000000	3.000000	0.000000	0.000000	0.000000
4	0.000000	20.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	-20.000000	0.000000	0.000000	0.000000	0.000000

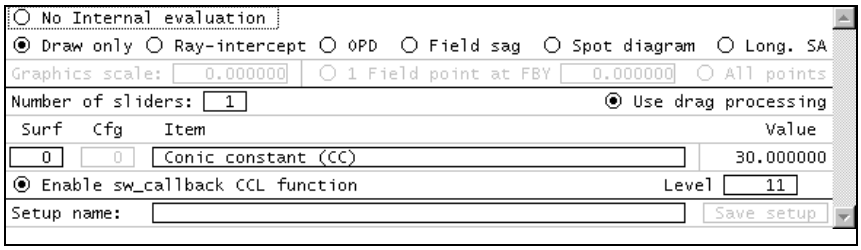
Array ray tracing is comparatively fast to non-sequential ray tracing, because surfaces are selected according to the nearest channel vertex rather than the actual surface. For many situations, this is a good model, but for this tabular array, it is not adequate for large field angles. To see this, it is worth attaching the field angle to a graphic slider so that it can be adjusted by dragging while the ray trajectories are observed.

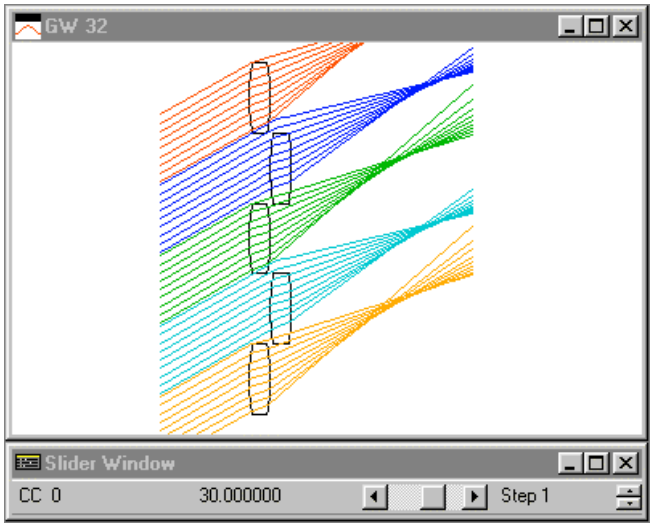
In order to attach the field angle to a slider, we use the same technique used elsewhere in these examples, making use of the fact that the conic constant of the object surface has no optical function when the surface is flat. We make a slider-wheel callback function as shown below, and put it in the private CCL directory.

```
cmd Sw_callback(int cblevel, int item, int srf)
```

```
{
        if (cblevel == 11)
        {
                stp outp off;
                ang cc[0];
                stp outp on;
        }
        else
                ite cblevel;
}
```

After recompiling the private CCL, we setup a slider-wheel window as follows.





When the setup window is closed, the slider-wheel window appears, and you can see that at wide angles, rays do not follow their actual trajectories, because of the way that channels are selected. This is not a problem for narrow fields or when surfaces are not displaced from the channel surface, as you can verify by manipulating the slider.

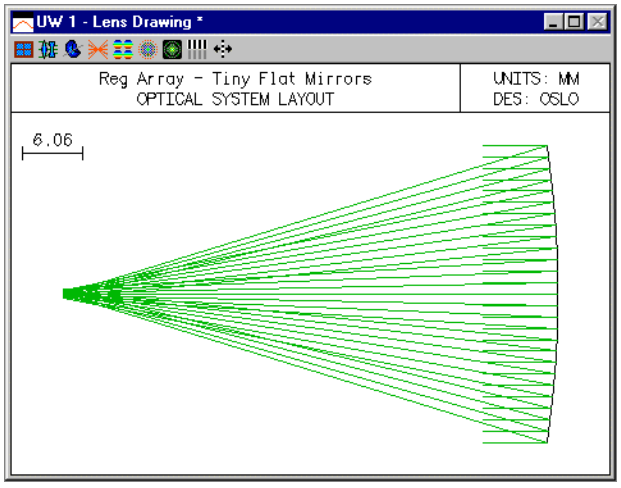
Note that for the slider to work properly in this example, the Fractional Y object height for all the field points must be set to 1, as shown in the table below. You may also note that it is not possible to set the field angle to zero using the slider. This is a feature of OSLO, which automatically converts field angles of 0.0 to 1 micro-degree, since 0.0 is not an allowed value for the paraxial field angle.

```
*CONDITIONS: LENS DRAWING
   Drawn apertures (solid):       Full    Image space rays:         Image srf
   Number of field points (rays):    5    DXF/IGES file view:     Unconverted
   Fpt Frac Y Obj Frac X Obj Rays Min Pupil  Max Pupil    Offset  Fan Wvn Cfg
    1     1.00000      --      9   -0.95000    0.95000      --     Y   1   0
    2     1.00000      --      9    1.05000    2.95000      --     Y   1   0
    3     1.00000      --      9    3.05000    4.95000      --     Y   1   0
    4     1.00000      --      9   -2.95000   -1.05000      --     Y   1   0
    5     1.00000      --      9   -4.95000   -3.05000      --     Y   1   0
```

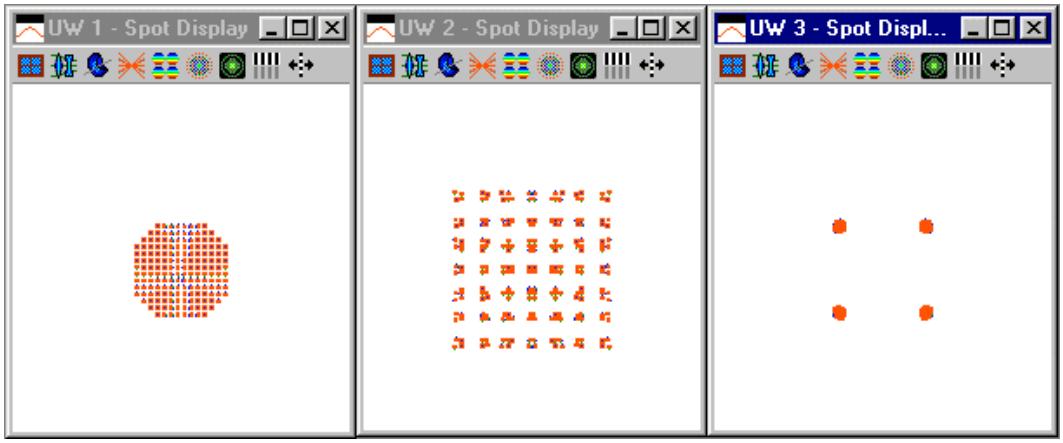
2D Array

As an example of a 2D array, we show a system comprising a large number of small flat mirrors mounted on a parabolic substrate with a focal length of 50mm and a diameter of 30mm (f/1.67). The mirrors have 1 mm width, and a center spacing of 1 mm. The data for the system (mirorary.len) are shown below.

```
*LENS DATA
Reg Array - Tiny Flat Mirrors
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --        1.0000e+20    8.7489e+18            AIR

 AST  -100.000000        --          15.000000 A           AIR   *

  2         --        -50.000000      0.707100 KX      REFLECT   *

 IMS        --            --          25.000000

*CONIC AND POLYNOMIAL ASPHERIC DATA
 SRF        CC          AD          AE          AF          AG
  1      -1.000000      --          --          --          --

*TILT/DECENTER DATA
  2     RCO   1

*LENS ARRAY DATA
 SRF 1:
 TYPE Regular              END SURF 2               DRAW ALL CHANNELS:  No
 X SPACING    1.000000     Y SPACING    1.000000    Y OFFSET      --

*APERTURES
 SRF   TYPE APERTURE RADIUS
  0     SPC   8.7489e+18
  1     SPC    15.000000
  2     SPC     0.707100  CHK

     Special Aperture Group 0:
     A  ATP    Rectangle  AAC     Transmit  AAN       --
        AX1    -0.500000  AX2     0.500000  AY1    -0.500000  AY2     0.500000

  3     SPC    25.000000
```

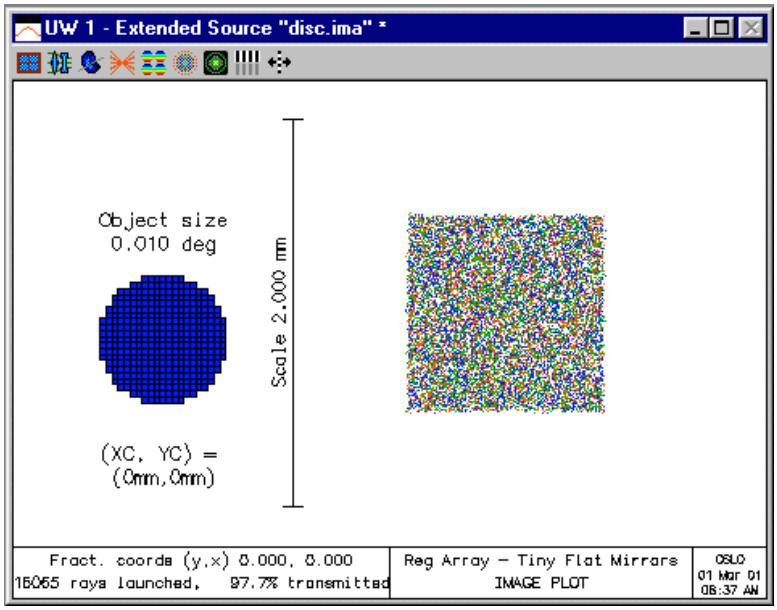


Evaluating the system using a spot diagram produces results that depend strongly on the aperture divisions used, and the focus shift from the focal point of the parabolic substrate. (Since the system has only flat mirrors, it actually has an infinite focal length.) The figure below shows spot diagrams for various aperture divisions (15, 17.5, and 20), with a focal shift of 0.1 mm. The command used was **pls cen sym 0.1 1.0.**



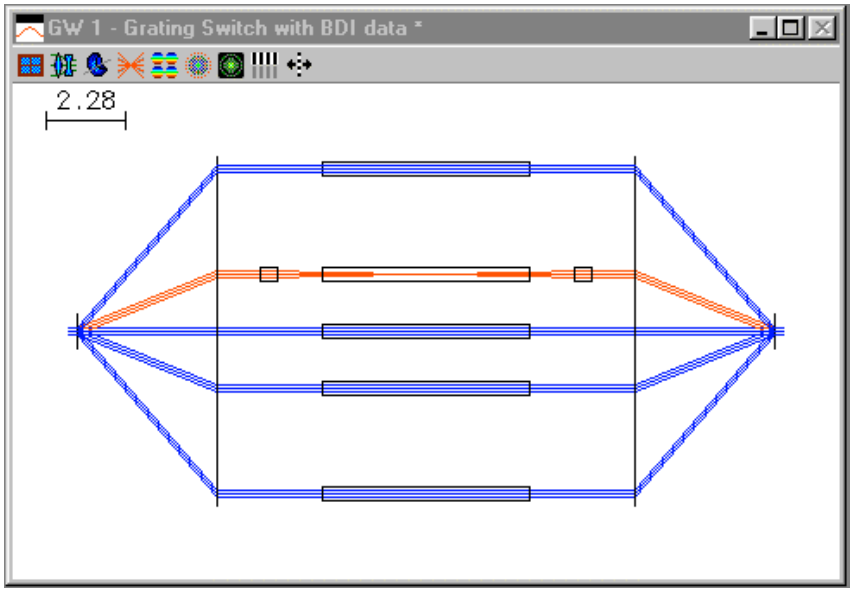
The explanation for these curious results is that there is aliasing between the ray grid and the mirror grid. The overall diameter of the paraboloid is 30 mm, so when APDIV = 15, there is one ray that strikes the center of every other mirror. When APDIV = 17.5, the mirror spacing and the ray spacing are not coupled, so rays hit in nearly random points on the mirrors, and we see a (reversed) shadow of the 1 mm square mirrors. When APDIV = 20, no rays strike the center of a mirror, but all rays strike one of four possible locations on a mirror. This leads to the four-dot pattern shown above, which of course bears no similarity to the real light distribution. (The center pattern above gives the closest approximation to the real light distribution.)

When using spot diagrams (or any type of evaluation routine) with lens arrays, it is well to be aware of the possibility of aliasing effects between the ray grids used for evaluation, and the lens array grid itself. Often the best solution to these types of problems is to use random ray tracing. The figure below, for example, shows the image distribution computed using the xsource routine (Source>>Pixelated Object), using a small disc object that subtends a field angle of 0.01 degrees.



Grating Switch

Array ray tracing can be combined with multiconfiguration (zoom) data to handle a wide variety of different systems. Here, we show as an example a grating switch, which uses diffraction gratings to disperse a beam into several different channels and then recombine it for further transmission. The particular system shown here is a simple 5x one-dimensional switch, but the techniques used are also applicable to 2D arrays. The final system is shown below.



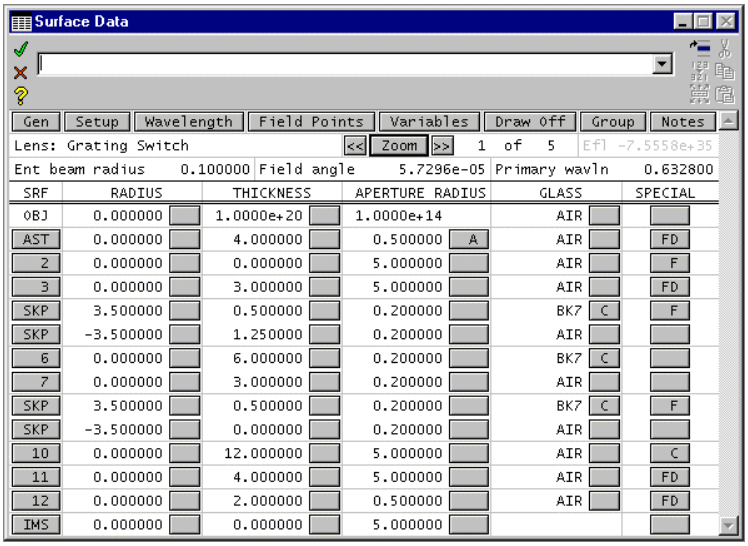
The switch shown here doesn't actually do anything useful, it is just an example of how to set up such a system in OSLO. It consists of four 600 line/mm gratings, 5 glass rods, and 2 lenses, arranged as shown. An input beam enters the system from the left, is diffracted into several orders (5 shown), redirected by a second grating, transmitted through various channels, and then recombined by a symmetrical set of gratings into an output beam that exits to the right.

There are several steps needed to set up such a system in OSLO, some optical ones needed for design/evaluation, and some cosmetic ones need to draw a picture of the final system. An important consideration is that OSLO, being fundamentally a design program, does not split rays; for each input ray there is one output ray. In the above figure, 15 rays are traced, 3 for each channel. In order to handle the multichannel arrangement of the system, we use zoom data, with each configuration representing one chanel. Although the present system has only 5 channels, OSLO accommodates an unlimited number of zoom configurations, so the potential exists to extend the method used here to dozens or even hundreds of channels.

The switch shown here switches channels by diffraction into various orders. In the layout, we associate a diffraction order with each zoom position, ordered 0, 1, 2, -2, -1. This is combined with array ray tracing so that each configuration uses one channel. Because diffraction orders are spaced according to the sines of angles, the various channels are not equally spaced, so a tabular array is used to define them.

In array ray tracing, each channel must be the same, but in the present system there is a need for one of the channels to be different from the others. This is accommodated by a capability in OSLO to handle what are called *skip* surfaces. A skip surface is one that redirects the ray trace to another surface (further on). For example, in the present system surface 4 is made a skip surface with a target of 6, in all configurations except cfg 2. The result is that surfaces 4 and 5 are ignored except in configuration 2.

The data for the grating switch is shown in the spreadsheet below, and the special data listing that follows.




```
*TILT/DECENTER DATA
 10     RCO   3

*SURFACE TAG DATA
  1     GOR   0          GSP     0.001667   GBO   0          GDP       --
        DRW ON
  2     DRW ON
  3     GOR   0          GSP     0.001667   GBO   0          GDP       --
        DRW ON
  4     SKP   6
  8     SKP  10
 11     GOR   0          GSP     0.001667   GBO   0          GDP       --
        DRW ON
 12     GOR   0          GSP     0.001667   GBO   0          GDP       --
        DRW ON

*LENS ARRAY DATA
 SRF 3:
 TYPE Tabular   END SURF 10    MAX CHANNELS 5           DRAW ALL CHANNELS: Yes
 CH NBR    X CTR       Y CTR       Z CTR       TLA         TLB         TLC
   1         --          --          --          --          --          --
   2         --        1.641655      --          --          --          --
   3         --        4.668182      --          --          --          --
   4         --       -4.668182      --          --          --          --
   5         --       -1.641655      --          --          --          --

*CONFIGURATION DATA
TYPE  SN   CFG  QUALF       VALUE
GOR    1    2     0             1
GOR    3    2     0            -1
GOR   11    2     0            -1
GOR   12    2     0             1
SKP    4    2     0             0
SKP    8    2     0             0
TH     3    2     0      1.250000
TH     7    2     0      1.250000
GOR   12    3     0             2
GOR    1    3     0             2
GOR   11    3     0            -2
GOR    3    3     0            -2
GOR    3    4     0             2
GOR   11    4     0             2
GOR   12    4     0            -2
GOR    1    4     0            -2
GOR   12    5     0            -1
GOR    1    5     0            -1
GOR    3    5     0             1
GOR   11    5     0             1
```

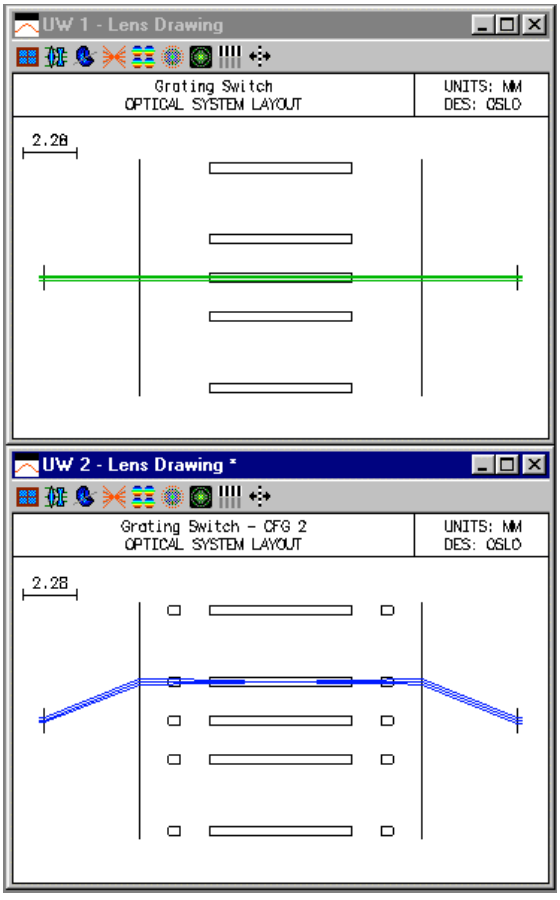
Surfaces 1 and 3 accommodate the gratings that diffract the beam into various orders and redirect the orders so they are parallel to the axis of the system. Since there is no refractive index associated with these surfaces, they need to be tagged (SPECIAL>>Surface Control>>General) in order to be drawn. Surface 3 also serves as the channel surface for the array. Surface 2 is a dummy surface in contact with the channel surface, used for drawing *bdi* data (to be discussed), and has no optical function.

Surface 3 is marked as an array surface (SPECIAL>>Surface Control>>Tabular array), and holds the *y*-decentration data for each channel. In order to determine the data values, a chief ray is traced from the edge of the field (here defined to be at 1 micro-radian) using the **Chf** button in the text window. The required data from the spreadsheet buffer is copied to the clipboard using CTRL+click, and pasted into the appropriate cell in the tabular array spreadsheet. Note that you must activate an array element by clicking on the row button prior to entering data for it.

Surface 4 is tagged as a skip surface, as discussed above (SPECIAL>>Surface Control>>General) and data for a bi-convex lens is entered on surfaces 4 and 5. Surfaces 6 and 7 define a BK7 glass rod that will be seen in all configurations. Surface 8 is tagged as a skip to surface 10, and surfaces 8 and 9 define a second BK7 lens like the first.

The configuration data for the system mostly defines the requisite diffraction orders so that the channels are numbered 0, 1, 2, -2, -1 corresponding to cfg 1, 2, 3, 4, 5. In addition, the *skip* parameter is turned off in configuration 2, and the thickness of surface 3 is changed in configuration 2 so that the glass rod is not moved.

Surface 10 is the EAR (end-of-array) surface. Note that the array definition automatically places an RCO (return-coordinates) transfer to the channel surface, so that surface 11 (the first of the final grating surfaces) is located with respect to the channel surface, and hence does not require any configuration specification. Surface 12 contains the final grating that recombines the beams from the various array channels. At this point, the system is set up as shown below.



In order for the darwing rays to change color in configuration 2, you should set up your Len drawing operating conditions as shown below.

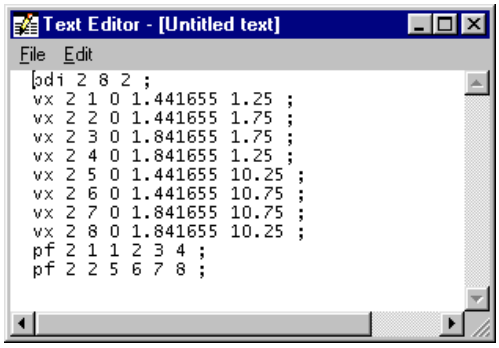
Number of field points for ray fans: 2 Points for aspheric profiles: 41

Frac Y Obj	Frac X Obj	Rays	Min Pupil	Max Pupil	Offset	FY	FX	Wvn	Cfg
1.000000	0.000000	3	-1.000000	1.000000	0.000000	◉	○	1	0
1.000000	0.000000	3	-1.000000	1.000000	0.000000	◉	○	1	2

The drawings show the system functioning optically as expected, with the lenses exerting focusing action only in configuration 2. However, for consmetic reasons, the drawings are less than satisfactory, partly because only one configuration is shown at a time, and partly because lenses are shown in all the channels, although they are only active in one. In order to circumvent these problems, we need to set up some additional data for drawing purposes only.

First, we can remove the extra lens drawings by marking the lens surfaces *not drawn* in the SPECIAL>>Surface Control>>General spreadsheet. Then, of course, the lenses disappear in configuration 2 as well, but this can be accommodated using *bdi* data. BDI data are one form of drawing data that can be attached to a lens surface, and allow the construction of 3D drawings that cannot be produced by the internal routines in OSLO. For the present system, we will only need to make a plan view of the system, so we can represent each lens by a single facet, i.e a rectangular box. There is no special bdi editor in OSLO, so bdi data is normally entered using the internal OSLO text editor. Since the data are a series of commands, the internal OSLO editor is generally preferred over an external text editor, since the command lines can be selected and executed directly from the editor.

The data required for the present system is shown in the figure below. It consists of 8 vertices defining the two rectangles, attached to and located relative to surface 2. Note that surface 2 is outside the array channel. The *y* data are computed as the sum of the chief ray height in cfg 2, plus or minus the lens aperture, and the *z* data are computed relative to surface 2.



Once the bdi data is entered satisfactorily, the lenses should be drawn properly, but the problem still exists that only one configuration is shown at a time. To solve this, we need to overlay the drawings from all configurations. To do this, we turn off the Graphics Autoclear preference (shown on the default status bar), make a drawing in each configuration, and then turn the preference back on. It is easiest to do this using commands, viz.

```
Stp gacl off;
Cfg 1; pen 1; drl; pen 2; ddr;
Cfg 2; pen 1; drl; pen 3; ddr;
Cfg 3; pen 1; drl; pen 2; ddr;
Cfg 4; pen 1; drl; pen 2; ddr;
Cfg 5; pen 1; drl; pen 2; ddr;
Stp gacl on;
```

This should produce a drawing like the one at the beginning of this example. If you are nort able to reproduce this, it is possible that you have not defined your default drawing rays the same as here.

Non-sequential ray tracing

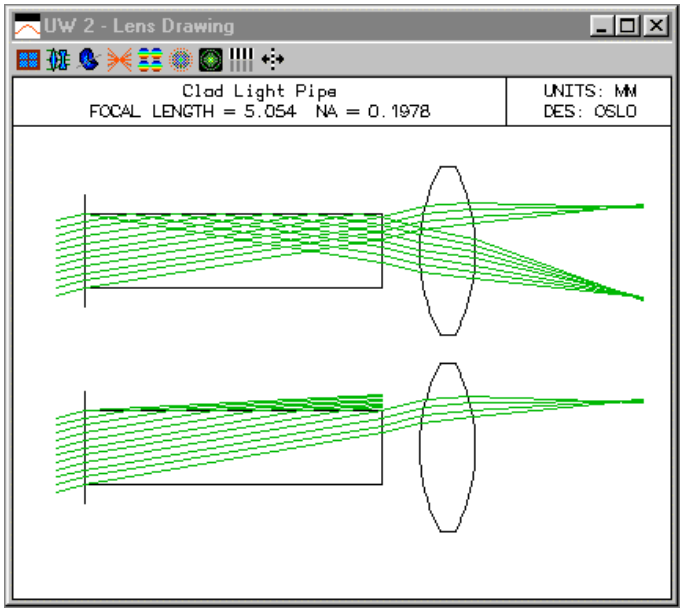
Light pipes/fibers

In general the term non-sequential ray tracing is applied to systems where the ray trace algorithm must decide not only where a ray strikes the next surface, but also which surface of several possibilities is the one actually intersected. A special case of non-sequential ray tracing occurs when a ray strikes the same surface an indeterminate number of times.

OSLO (all versions) contains a special surface for handling a *light pipe* or *fiber*, in which rays enter a tube and repeatedly reflect from the wall until they emerge from the far end. Only a straight elliptical or circular tube is permitted, and the ends of the tube must be perpendicular to its axis. A surface may be extruded to form a rod by entering the command **rod on** at the surface. To disable this feature, use the command **rod off.** Control of this surface property is also available from the Special >> Surface Control >> General spreadsheet.

An extruded surface consists of two parts: (a) the surface defined by the usual specifications, limited by the defined aperture boundary, and (b) the surface generated by *pulling* the aperture boundary in the *z*-direction by the distance specified by the thickness variable, **th**. The effect is to generate a rod shaped object whose cross section conforms to the aperture. If an elliptical special aperture is specified, this is used to define the cross-sectional shape of the rod. Otherwise, the circular aperture specified by the **ap** command is used.

The glass specification for the surface defines the medium of the rod. Normally, rays are confined to the interior of the rod by total internal reflection (which may take place several times before the ray encounters the next surface). If a ray exits the rod by refraction, it refracts into the original incident medium. The rod is terminated by the next surface in sequence.



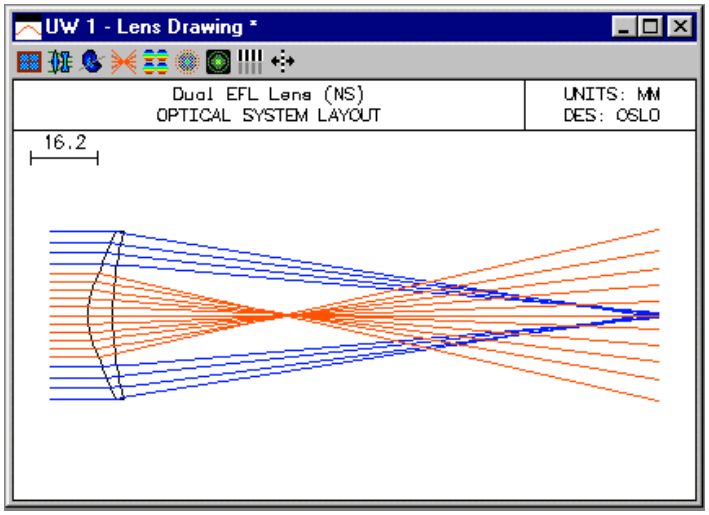
All of the usual ray trace commands are available for systems employing light pipes. However, for single ray tracing, only the ray data for the entrance and exit faces is displayed, and much of the other ray trace data (e.g., ray-intercept curves) is essentially meaningless because of the variable number of reflections in the light pipe.

Although the light pipe routine in OSLO is a form of non-sequential ray trace, it differs in that it does not take into account the fact that surfaces have two sides. There is no way, for example, in the above trace to have light re-enter the core from the cladding. Light that is lost into cladding is normally blocked at the far end of the pipe by using a checked aperture, as shown above.

Dual focal length lens

The following is intended as a simple example of non-sequential ray tracing. The system to be studied is a simple meniscus lens whose front surface has two regions having different curvature. Rays through the center of the lens encounter one curvature, rays through the edge another. The lens data are as follows.

```
*LENS DATA
Dual EFL lens (NS)
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE   NOTE
  0         --        1.0000e+20    1.0000e+14            AIR

  1         --             --         18.000000 AS         AIR   *
  2     20.000000          --         10.000000            BK7 C *
  3     40.000000          --         20.000000 X          BK7 C *
  4     80.000000          --         20.000000            AIR   *
  5         --        127.000000     20.000000            AIR   *

  6         --             --         20.000000

*TILT/DECENTER DATA
  2     DT    1          DCX       --       DCY       --       DCZ       --
                         TLA       --       TLB       --       TLC       --
  3     DT    1          DCX       --       DCY       --       DCZ     1.409325
                         TLA       --       TLB       --       TLC       --
  4     DT    1          DCX       --       DCY       --       DCZ     6.000000
                         TLA       --       TLB       --       TLC       --
  5     DT    1          DCX       --       DCY       --       DCZ    10.000000
                         TLA       --       TLB       --       TLC       --

*SURFACE TAG DATA
  1     LMO NSS  (5 surfaces)
  3     ELI   1
  4     ELI   1
  5     DRW OFF

*APERTURES
 SRF   TYPE APERTURE RADIUS
  0     SPC   1.0000e+14
  1     CMP    18.000000
  2     SPC    10.000000
  3     SPC    20.000000

     Special Aperture Group 0:
     A  ATP      Ellipse  AAC    Pass Thru  AAN       --
        AX1   -10.000000  AX2    10.000000  AY1   -10.000000  AY2    10.000000
```



The following comments may help you to understand the data:

Surface 2 is the central portion of the front surface. Surface 3 is the outer portion of the front surface, and surface 4 is the back surface.

The central zone of the first surface has a 10 mm aperture radius and a radius of curvature of 20 mm. The outer zone of the first surface has a 20 mm aperture radius and a 40 mm radius of curvature.

Surface 3 has a special aperture to create a hole in its center for surface 2. The “Pass Thru” designation makes the central portion of the surface equivalent to a hole.

The glass BK7 is put on surfaces 2 and 3, because the normal action is for rays to refract into these surfaces in the *to positive* direction. In the present example, rays will not encounter either of these surfaces from the back side, so no special actions are needed.

It is very important to ensure that the entry port and the exit port completely surround the non-sequential surfaces. For this lens, the center portion of the lens is placed in contact with the entry port, which is OK, because the sag of the surface is always positive. The axial thickness of the lens is 6mm, which determines the DCZ of surface 4. The exit port (surface 5) is placed at a distance of 10 mm from the entry port, to ensure that no portion of surface 4 falls outside the non-sequential region. The remaining DCZ (of surface 3) is determined by the requirement that the sags of surface 2 and 3 are identical at the edge of the central zone (10 mm height). The value can be found using the edge thickness (**eth 2 3 10 10**) command in OSLO, which produces the required value 1.409325.

To force the program to draw an aperture connecting surfaces 3 and 4, the element ID (ELI) is set to 1.

The DRW OFF designation on surface 5 keeps the program from drawing the exit port.

The data for this lens is entered as follows:

Select File >> New, enter the file name “bifocal.len”, then enter 5 for the number of surfaces. Click OK to dismiss the dialog box. The surface data spreadsheet will appear.

In the surface data spreadsheet, click the row button for surface 1, then the row button for surface 5 to select the surface range (entry and exit ports, plus 3 lens surfaces).

Set up a non-sequential group using Edit >> Non-sequential Group. Click the View Srf radio button at the top of the spreadsheet, so you can see the surface data. Click the Draw On radio button so you can see the effects of your data entries. Enter the value 18 for the Entrance Beam Radius.

Enter the data for the RADIUS of surfaces 2-4 as given in the listing above. Surface 2 is the central portion of the front surface. Surface 3 is the outer portion of the front surface, and surface 4 is the back surface.

Enter the data shown above for the apertures. The central zone is to have an aperture of 10 mm radius, and the overall lens is to have an aperture of 20 mm.

Surface 3 has a special aperture to create a hole in its center for surface 2. Click the button next to the aperture of surface 3, and select Special aperture data from the pop-up list. Enter “1” for the number of special apertures, then enter the data from the above listing in the special aperture spreadsheet that pops up. The “Pass thru” designation makes the central portion of the surface equivalent to a hole.

The glass BK7 is put on surfaces 2 and 3, because the normal action is for rays to refract into these surfaces in the *to positive* direction. In the present example, rays will not encounter either of these surfaces from the back side, so no special actions are needed.

Next you need to enter the data for the surface locations, relative to the Entry port (surface 1). It is very important to ensure that the entry port and the exit port completely surround the non-sequential surfaces. For this lens, the center portion of the lens is placed in contact with the entry port, which is OK, because the sag of the surface is always positive. The axial thickness of the lens is 6 mm, which determines the DCZ of surface 4. The exit port (surface 5) is placed at a distance of 10 mm from the entry port, to ensure that no portion of surface 4 falls outside the non-sequential region. The remaining DCZ (of surface 3) is determined by the requirement that the sags of surface 2 and 3 are identical at the edge of the central zone (10 mm height). The value can

be found using the edge thickness (**eth 2 3 10 10**) command in OSLO, which produces the required value 1.409325.

For each of the surfaces 3-5, click Special >> Local/Global Coordinates, and enter the appropriate DCZ data as discussed above.

To force the program to draw an aperture connecting surfaces 3 and 4, for both surfaces click on Special >> Non-sequential Data, then set the element ID to 1.

To keep the program from drawing the exit port, on surface 5, click Special >> Surface Control >> General, and set the Surface appearance to Not drawn.

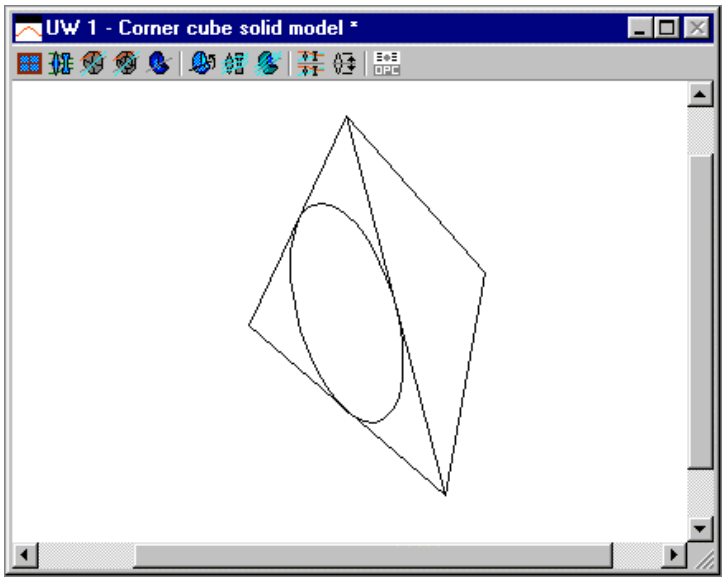
To show the ray trajectories, set the thickness of surface 5 to 127, as shown above, then click Update >> Operating Conditions >> Lens Drawings. In the spreadsheet, set the Image space rays button to Image Srf, set the number of field points to 1, set the number of rays to 15, and then click OK to dismiss the spreadsheet.

At this point, if all has gone well, your Autodraw window should have a picture similar to the one shown above. You should now experiment by changing all the data a little to see what happens.

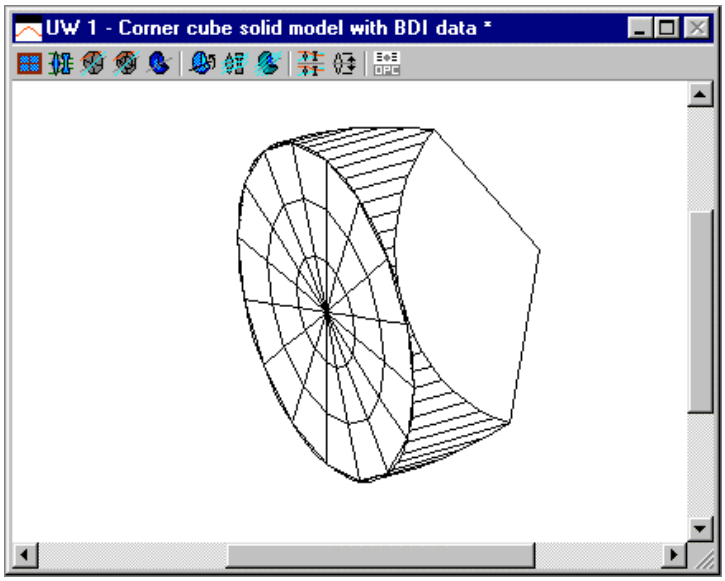
Corner-cube reflector

The corner-cube reflector, a combination of three plane mirrors at right angles to each other, has the property that rays entering the reflector exit from the reflector anti-parallel to the entering rays. A beam traversing a corner cube hits all three mirrors, but the order differs for different parts of the beam. Accordingly, corner cubes must be handled using non-sequential ray tracing. Generally, depending on the material and the conditions of use, a corner cube can function using total internal reflection, but there are reasons to "silver" the reflecting surface (discussed below), and the the standard corner cube prescription supplied in the OSLO demo library (cornercube.len) desginates the mirror surfaces as REFLECT.

The geometry of the corner cube reflector is such that rays entering the system perpendicular to the entrance face hit the first mirror at an angle of incidence of 54.73561 degrees. This makes the default drawing produced by OSLO look somewhat strange, as follows:



The portions of the entrance face outside the circular aperture are not optically useful, and typically when corner cubes are fabricated they are edged so that the fit into a cylindrical barrel. Accordingly, the cornercube.len model uses BDI data to produce 3D drawings (BDI data is ignored in plan-view drawings), as follows.



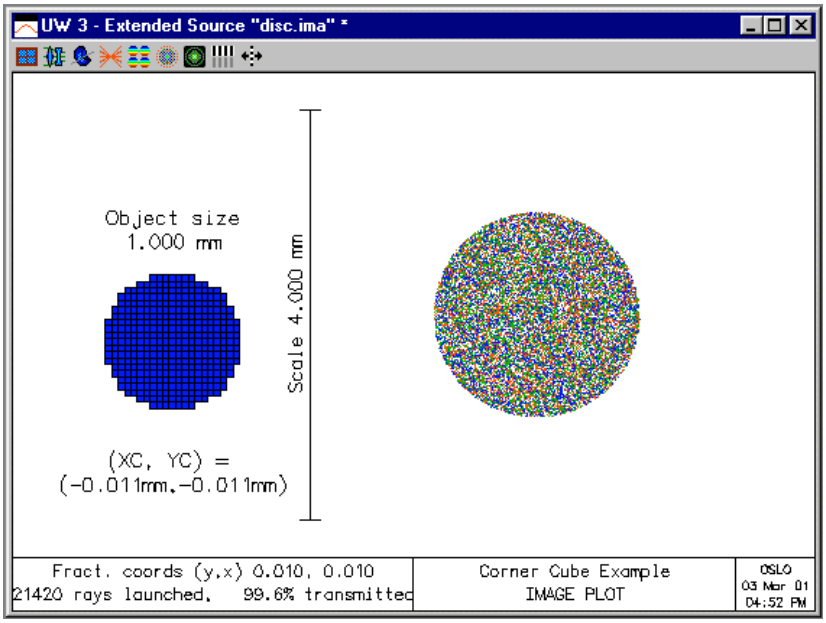
Unlike many BDI drawings, which use only a few vertices to define rectangular or triangular surfaces used in prisms, the data required for the corner cube model is quite extensive, because the cylindrical surface is simulated by facets. A few items of data are shown here to show that lines that define facets are drawn if the vertex number is positive, but supressed if it is negative. Thus in the following data for polygon facet 1, lines are drawn from vertex 4 to 1, and from vertex 1 to 2, but not from vertex 2 to 3. Regardless of whether the lines are drawn, however, the facet area is defined according to the PF specification, and hides the lines and facets behind it according to the viewing angle.

```
*BOUNDARY DRAWING DATA
 SRF 1:
 VX NBR          X             Y             Z          COORD SURF
    1            --            --          0.570000          1
    2         0.866025     -0.500000     -0.132893          1
    3         0.913545     -0.406737     -0.257040          1
    4         0.951057     -0.309017     -0.372080          1
    ...
 PF NBR         VX1           VX2           VX3           VX4
    1             1             2            -3            -1
    2            -1            -3            -4            -1
```

The specification of the non-sequential group for a corner-cube reflector is straightforward, and consists of the entrance port, the three reflecting surfaces, and the exit port. The other surfaces in the cornercube.len prescription are used to satisfy drawing requirements and are not optically active. Note that because of the complexity of the BDI data, it may be useful to insert the cornercube.len model in other systems as needed, using the Insert Lens file and Scale lens commands on the right-click spreadsheet menu.

Historically, the corner-cube was one of the first applications for non-sequential ray tracing. According to optics folklore, a problem with double images was encountered when a particular optical system was being fabricated in the shop. The question arose as to whther these double images could be simulated using optical design software, so that the design could be adjusted. It turns out that the only design adjustment needed is to silver the reflecting faces, but as a pedigogical exercise, it is interesting to see what goes on when light retroreflects from a corner cube. To do this requires both non-sequential and polarization ray tracing, so it is necessary to use OSLO Premium.

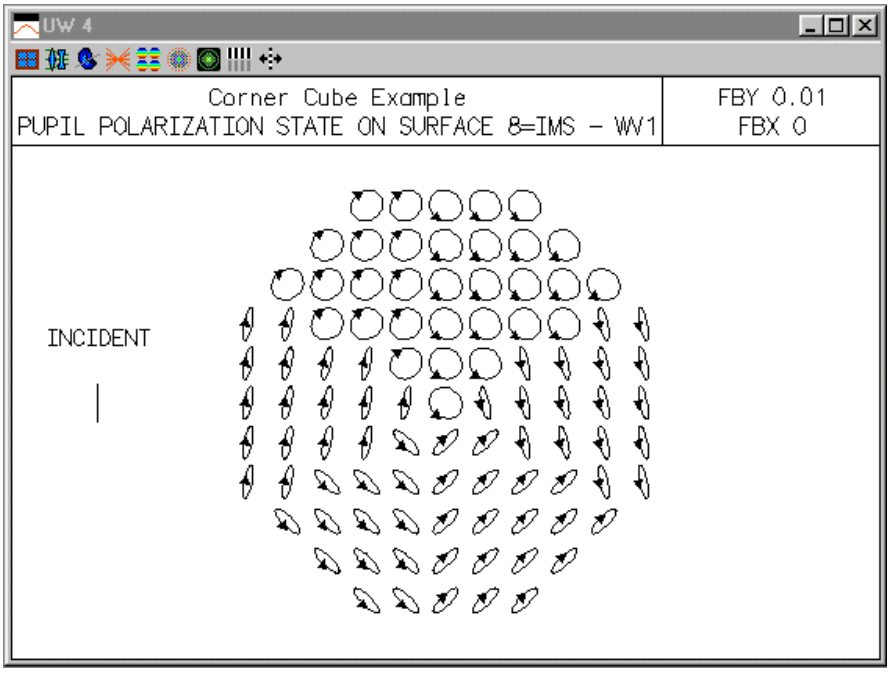
If you illuminate a corner-cube with a small but finite incoherent source, a spot diagram, even a random one, will show even illumination, as shown below.



This is to be expected, since a spot diagram shows only the intersection points of rays with the image plane, and gives no information concerning the irradiance. Thus spot diagrams are useful

for visualizing the shape of images formed by optical systems with aberration. In the case of a corner cube, however, there are no aberrations. On the other hand, there are polarization effects that can be readily observed using the polarization ray trace in OSLO.

To observe the polarization effects, you can use the cornercube.len model and turn on polarizatiion ray tracing in the General operating conditions. In addition, you must change the REFLECT designation on the mirror surfaces to AIR. If you do that, then you can use the Evaluate>>Polarization>>Pupil Polarization State command to produce the following plot.



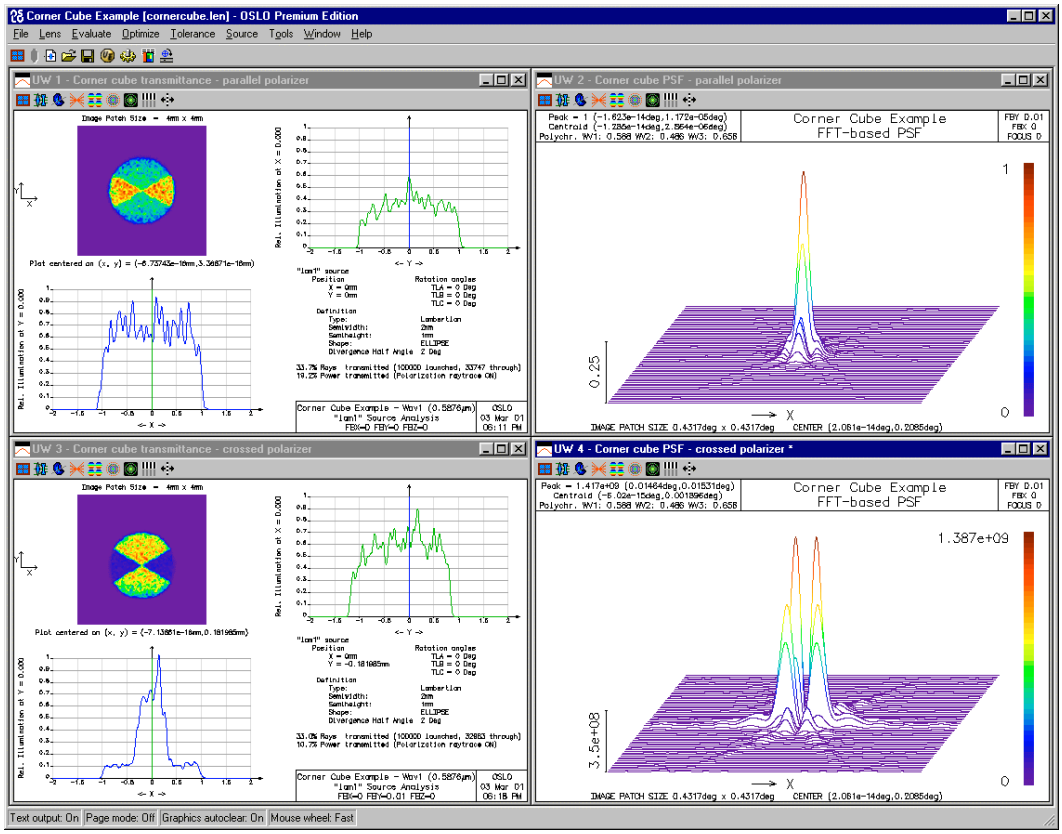
You see there are polarization effects that differ according to how light traverses the corner cube. Because of the unusual angle that rays intersect the mirror surfaces, different hexagonal sectors undergo different phase shifts that lead to the output polarization states shown. The arrowheads indicate the direction of rotation of the electric vector, showing that the output beam includes both right and left elliptically polarized components, when the incident beam is linearly polarized in the vertical direction.

You can analyze the system in more detail using the extended source and point spread function features in OSLO. The figure on the next page shows the results of such an analysis, produced by using a polarizing element on the output surface as follows, here shown as a crossed (i.e. *x*) polarizer. To set up a parallel polarizer, you use JA = 0.0 and JD = 1.0.

```
*POLARIZATION ELEMENT DATA
              AMPLITUDE     PHASE              AMPLITUDE     PHASE
  7      JA     1.000000      --          JB       --          --
         JC       --          --          JD       --          --
```

To produce the results on the following page, you need to set the object distance to 50mm for the illumination calculation, and 1e20 for the spread function calculation. In addition, you should set the object height slightly off axis (FBY = 0.01) to introduce a slight amount of aberration to prevent the crossed polarization case from blowing up during PSF normalization (the ideal PSF for this case is zero on axis).

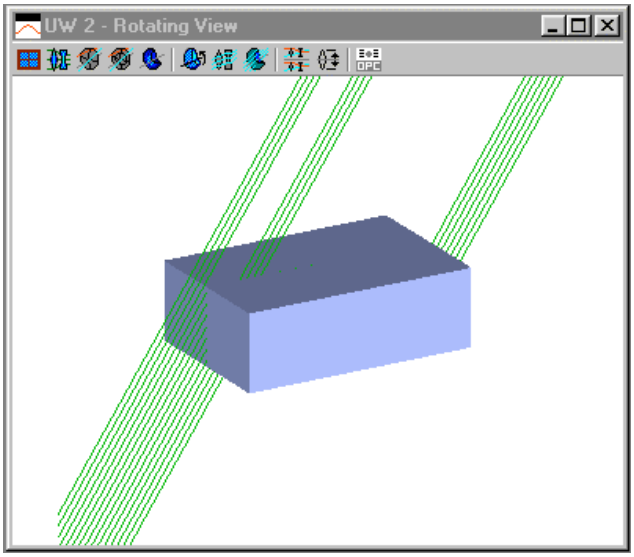
You see that there is intensity modulation introduced by the output polarizers, but this is not sufficient to cause the PSF effects, which are caused by the polarization itself.


Corner Cube Example [cornercube.len] - OSLO Premium Edition
File Lens Evaluate Optimize Tolerance Source Tools Window Help
UW 1 - Corner cube transmittance - parallel polarizer
UW 2 - Corner cube PSF - parallel polarizer
Corner Cube Example
FFT-based PSF
UW 3 - Corner cube transmittance - crossed polarizer
UW 4 - Corner cube PSF - crossed polarizer *
Corner Cube Example
FFT-based PSF
1.387e+09
Text output: On
Page mode: Off
Graphics autoclear: On
Mouse wheel: Fast

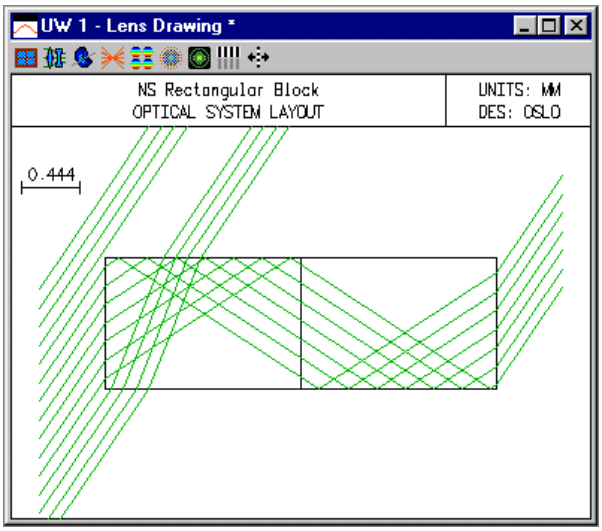
Nsobjects

Data entry for non-sequential groups can become tedious if the group contains more than a few surfaces, so it is generally worthwhile to set up commands that will create non-sequential groups more-or-less automatically. This permits you to enter a group quickly by entering only the key parameters for the group in a diaglog box. Examples of a few non-sequential objects can be seen using the *nsobjects* CCL command, which can be accessed from the Tools>>demo library. The nsobjects command supplied with OSLO contains basic objects, such as a lens, rod, and clad fiber.

In this example, we show how to make CCL commands for data entry. The method involves little or no programming, but instead uses OSLO spreadsheets to set up a prototype system, which then provides a template that can easily be converted to symbolic entry. The specific task here is to add a rectangular block object to the nsobjects collection. The starting point for the example is the rectblock.len file in the OSLO demo library, shown below.



To see the ray paths in the block, it is helpful to attach a slider to the field angle using the sw_callback with the angle attached to the object surface conic constant, as is done elsewhere in these examples. At an angle of 56 degrees, the system looks as follows.



Although the system here is a regular block of glass that might be used as a prism, the same basic geometry can be used for everything from windows to light guides, depending on the width,

height, and depth of the object. The task here is to create a command that generates a suitable object automatically. The approach is to edit the lens file with a text editor, changing the numerical values for these item into symbolic entries, and turning the lens file into a CCL command. The listing below shows the conversion that is required. Basically, an OSLO lens file is a series of CCL commands with literal arguments, so there is not too much to do.

Replace the initial comment line with a cmd definition, and put the whole lens file inside curly backets. In setting up the command definition, you need to decide what arguments to use. For the present case, we need the ones shown in the listing below: semiwidth, semiheight, length, glass, and title (optional).

Required: Put a semicolon at the end of each line.

Optional: Convert the entries into parenthesized form. This is not necessary, but is good form.

Optional: Define an additional variable semilength and set it equal to half the input length to simplify data entry.

Optional: Bracket the code with *stp outp off* and *stp outp on* commands to eliminate echoing in the text output window.

Go through the file, replacing the numeric entries by their symbolic equivalents. You can usually recognize the required conversion by the values, In the present case, the semiwidth = .5, the semiheight = 1.0, and the length = 3.0.

```
                                cmd nsblock(real Semiwidth, real Semiheight, real
                                Length, char Glass[], char Title[])
                                {
                                real semilength;

                                Stp outp off;
                                semilength = 0.5*length;
// OSLO 6.1 3447 0 32016
LEN NEW "Rect Block" 0 8        LEN NEW Title;
EBR  1.0                        EBR  1.0;
ANG  6.8922645369e-13           ANG  6.8922645369e-13;
DES  "OSLO"                     DES  "OSLO";
UNI  1.0                        UNI  1.0;
// SRF 0                        // SRF 0
AIR                             AIR;
TH   1.0e+20                    TH   1.0e+20;
AP  1.2029270909e+06            AP  1.2029270909e+06;
NXT  // SRF 1                   NXT;  // SRF 1
AIR                             AIR;
LMO NSS                         LMO NSS;
NXT  // SRF 2                   NXT;  // SRF 2
GLA BK7                         GLA Glass;
AP  1.0                         AP  1.0;
DT   1                          DT   1;
APN  1                          APN  1;
AY1 A -0.5                      AY1 A -Semiheight;
AY2 A 0.5                       AY2 A Semiheight;
AX1 A -1.0                      AX1 A -Semiwidth;
AX2 A 1.0                       AX2 A Semiwidth;
ATP A 2                         ATP A 2;
AAC A 4                         AAC A 4;
NXT  // SRF 3                   NXT;  // SRF 3
AIR                             AIR;
NAC ORD PK 2 NEG                NAC ORD PK 2 NEG;
AP  1.0                         AP  1.0;
DT   1                          DT   1;
DCY  0.5                        DCY  Semiheight;
DCZ  1.5                        DCZ  semilength;
TLA  90.0                       TLA  90.0;
APN  1                          APN  1;
AY1 A -1.5                      AY1 A -semilength;
AY2 A 1.5                       AY2 A semilength;
AX1 A -1.0                      AX1 A -Semiwidth;
AX2 A 1.0                       AX2 A Semiwidth;
ATP A 2                         ATP A 2;
AAC A 4                         AAC A 4;
NXT  // SRF 4                   NXT;  // SRF 4
AIR                             AIR;
NAC ORD PK 2 NEG                NAC ORD PK 2 NEG;
AP  1.0                         AP  1.0;
DT   1                          DT   1;
DCY  -0.5                       DCY  -Semiheight;
```

```
DCZ  1.5
TLA  -90.0
APN  1
AY1 A -1.5
AY2 A 1.5
AX1 A -1.0
AX2 A 1.0
ATP A 2
AAC A 4
NXT  // SRF 5
AIR
NAC ORD PK 2 NEG
AP  1.0
DT   1
DCX  1.0
DCZ  1.5
TLB  -90.0
APN  1
AY1 A -0.5
AY2 A 0.5
AX1 A -1.5
AX2 A 1.5
ATP A 2
AAC A 4
NXT  // SRF 6
AIR
NAC ORD PK 2 NEG
AP  1.0
DT   1
DCX  -1.0
DCZ  1.5
TLB  90.0
APN  1
AY1 A -0.5
AY2 A 0.5
AX1 A -1.5
AX2 A 1.5
ATP A 2
AAC A 4
NXT  // SRF 7
AIR
LME
TH   0.001
AP  1.0
DT   1
DCZ  3.0
APN  1
AY1 A -0.5
AY2 A 0.5
AX1 A -1.0
AX2 A 1.0
ATP A 2
AAC A 4
NXT  // SRF 8
AIR
TH   1.0
APCK Off
GPRT On
WV 0.58756 0.48613 0.65627
WW 1.0 1.0 1.0
END  8
DLVA 0
DLHA 256
DLAP 3
DLNF 1
DLNR  0 15
DLFP  0 1.0
```

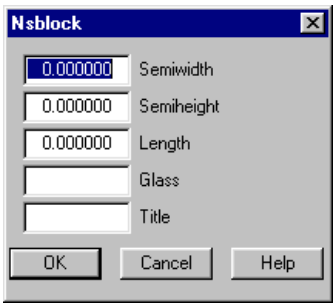
```
DCZ  semilength;
TLA  -90.0;
APN  1;
AY1 A -semilength;
AY2 A semilength;
AX1 A -Semiwidth;
AX2 A Semiwidth;
ATP A 2;
AAC A 4;
NXT;  // SRF 5
AIR;
NAC ORD PK 2 NEG;
AP  1.0;
DT   1;
DCX  Semiwidth;
DCZ  semilength;
TLB  -90.0;
APN  1;
AY1 A -Semiheight;
AY2 A Semiheight;
AX1 A -semilength;
AX2 A semilength;
ATP A 2;
AAC A 4;
NXT;  // SRF 6
AIR;
NAC ORD PK 2 NEG;
AP  1.0;
DT   1;
DCX  -Semiwidth;
DCZ  semilength;
TLB  90.0;
APN  1;
AY1 A -Semiheight;
AY2 A Semiheight;
AX1 A -semilength;
AX2 A semilength;
ATP A 2;
AAC A 4;
NXT;  // SRF 7
AIR;
LME;
TH   0.001;
AP  1.0;
DT   1;
DCZ  length;
APN  1;
AY1 A -Semiheight;
AY2 A Semiheight;
AX1 A -Semiwidth;
AX2 A Semiwidth;
ATP A 2;
AAC A 4;
NXT;  // SRF 8
AIR;
TH   1.0;
APCK Off;
GPRT On;
WV 0.58756 0.48613 0.65627;
WW 1.0 1.0 1.0;
END  8;
DLVA 0;
DLHA 256;
DLAP 3;
DLNF 1;
DLNR  0 15;
DLFP  0 1.0;
Stp outp on;
}
```

After completing the above steps, save the command in a CCL file (myobjects.ccl) in your private directory. Run OSLO, and compile your private CCL using the button in the main toolbar. Assuming that you receive the customary message

```
*CCL COMPILATION MESSAGES:
 No errors detected
```

The command should be ready to use. If you enter the command nsblock in the command line, you will be prompted for the arguments. You may find it more convenient to run the command by

entering "arg_entry=dialog_box;nsblock, in which case a dialog box will be generated automatically that prompts for input data.



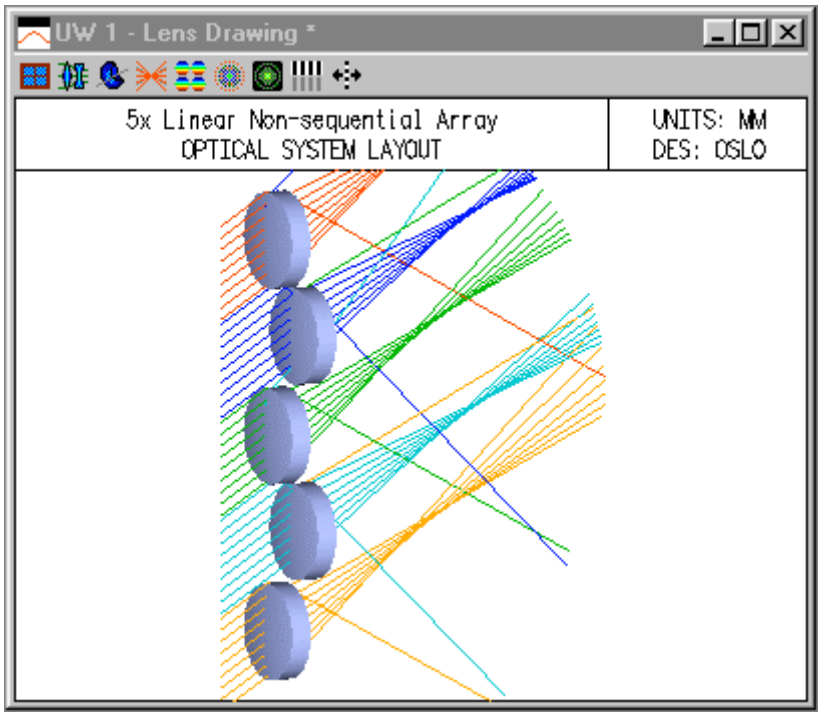
You have now added a command that you can use over and over, whenever you need a rectangular object for a non-sequential group. You may want to add it to the demo menu, or to a toolbar.

Clearly, the technique used here is not restricted to non-sequential groups, but can be used for any type of system that is sufficiently generally used to warrant the modest amount of setup time required to turn it into an object.

Non-sequential array

It is possible to set up entire arrays of lenses as non-sequential objects, rather than as arrays. For most design tasks, this is not a good way to proceed, but for evaluation it can be quite helpful, because it is possible to see exactly where light goes in passing through the system, which may not be possible with array ray tracing if the array has depth, as shown in the tabular array example. Here, we set up the tabular array as a non-sequential group to show the difference between non-sequential and array ray tracing.

We consider the same tabular array as is used for the array ray trace example. It consists of five elements in a line, with the odd elements displaced in the z-direction, as shown below.



The procedure for setting up such a system is to use the *nsobjects* CCL command to create a single non-sequential lens, then use copy-paste to duplicate it 4 times. Then the global coordinates of the vertices of each group are modified so the lenses are properly located. Including the entry and exit ports, it takes 22 surfaces to specify the array.

Gen | Setup | Wavelength | Field Points | Variables | Draw Off | Group | Notes

Lens: 5x Linear Non-sequential Array | Zoom 1 of 1 | Efl 2.9335e+60

Ent beam radius 5.000000 | Field angle 30.400000 | Primary wavln 0.587560

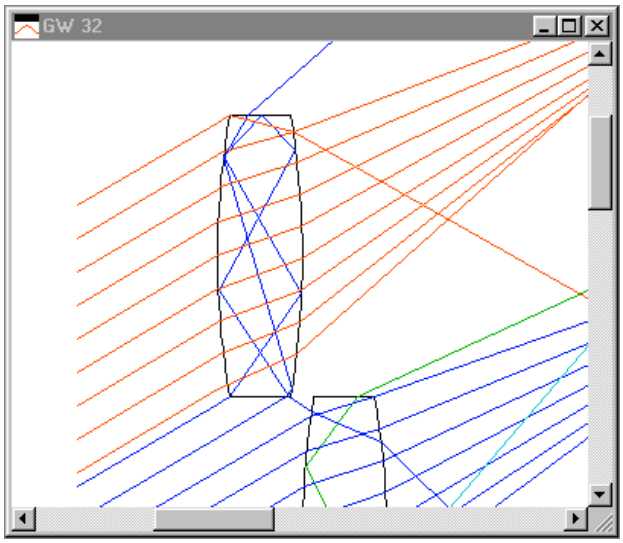
SRF	RADIUS	THICKNESS	APERTURE RADIUS	GLASS	SPECIAL
OBJ	0.000000	1.0000e+20	5.8670e+19	AIR	A
AST	NON SEQ GRP		10.000000 A	AIR	N
22	Exit Port		0.000000	AIR	NFC
IMS	-40.000000	0.000000	30.000000		C

Note that the field angle for the system has been specified as 30.4 degrees; this is to illustrate the ray trace behavior shown in the top figure. In addition, the exit port for this system is a total sphere with a radius of 40, centered on the vertex of the middle element. The reason for this is to ensure that rays exiting the system hit the exit port and are allowed to escape.

To identify the surfaces is a complicated non-sequential group such as this one, it is helpful to label the surfaces, which is done here using surface notes, as shown below.

```
*LENS DATA
5x Linear Non-sequential Array
 SRF      RADIUS      THICKNESS   APERTURE RADIUS       GLASS  SPE  NOTE
 OBJ        --        1.0000e+20    5.8670e+19            AIR   *

 AST        --             --         10.000000 A          AIR   * Entry Port
  2     30.000000         --          5.000000            BK7 C * mid-front
  3      5.000000         --          5.000000 X          AIR   * mid-topedg
  4     -5.000000 P       --          5.000000 PX         AIR   * mid-botedg
  5    -30.000000         --          5.000000            AIR   * mid-back
  6     30.000000         --          5.000000            BK7 C * upmid-front
  7      5.000000         --          5.000000 X          AIR   * upmid-topedg
  8     -5.000000 P       --          5.000000 PX         AIR   * upmid-botedg
  9    -30.000000         --          5.000000            AIR   * upmid-back
 10     30.000000         --          5.000000            BK7 C * botmid-front
 11      5.000000         --          5.000000 X          AIR   * botmid-topedg
 12     -5.000000 P       --          5.000000 PX         AIR   * botmid-botedg
 13    -30.000000         --          5.000000            AIR   * botmid-back
 14     30.000000         --          5.000000            BK7 C * top-front
 15      5.000000         --          5.000000 X          AIR   * top-topedg
 16     -5.000000 P       --          5.000000 PX         AIR   * top-botedg
 17    -30.000000         --          5.000000            AIR   * top-back
 18     30.000000         --          5.000000            BK7 C * bot-front
 19      5.000000         --          5.000000 X          AIR   * bot-topedg
 20     -5.000000 P       --          5.000000 PX         AIR   * bot-botedg
 21    -30.000000         --          5.000000            AIR   * bot-back
 22    -40.000000         --             --               AIR   * Exit Port

 IMS   -40.000000         --          30.000000                  *
```

In order to identify the anomalous rays in the previous lens drawing, it is helpful to try to identify the ray entering the system, and trace it through using either graphical or text output. Consider the spurious blue rays on the drawing. A plan view, combined with zooming ther window, shows the detail.



The top two blue rays, intended to go through the upper-middle lens, actually enter the top lens. (The lens has been set up so that the edges are polished and refract rather than scatter rays incident on them; other non-sequential actions could be imposed to absorb or reflect the rays). From the drawing, it appears that the rays strike the corners of the top element. In order to see exactly what happens, it is helpful to trace the actual rays and look at the numerical data. From the lens drawing operating conditions, we see that the top blue ray has a fractional aperture coordinate of 2.95 (based on the 30.4 degree field angle).

```
*CONDITIONS: LENS DRAWING
   Initial distance:            --        Final distance:              --
   Horizontal view angle:          240    Vertical view angle:              8
   First surface to draw:            0    Last surface to draw:             0
   X shift of drawing:          --        Y shift of drawing:          --
   Drawn apertures (solid):       Full    Image space rays:         Image srf
   Rings in aperture (solid):        3    Spokes in aperture (solid):       4
   Number of field points (rays):    5    DXF/IGES file view:     Unconverted
```

```
   Draw aperture stop location:    Off    Hatch back side of reflectors:   On
   Red value for shaded solid:     175    Green value for shaded solid:   185
   Blue value for shaded solid:    250    Points for aspheric profile:     41
   Fpt Frac Y Obj Frac X Obj Rays Min Pupil  Max Pupil    Offset  Fan Wvn Cfg
    1     1.00000      --      9   -0.95000    0.95000      --     Y   1   0
    2     1.00000      --      9    1.05000    2.95000      --     Y   1   0
    3     1.00000      --      9    3.05000    4.95000      --     Y   1   0
    4     1.00000      --      9   -2.95000   -1.05000      --     Y   1   0
    5     1.00000      --      9   -4.95000   -3.05000      --     Y   1   0
```

We can trace this ray by setting the field point to one and using either the Tra button in the text window toolbare, or the command:

`tra std glo all usr 2.95 0.0 n 1`

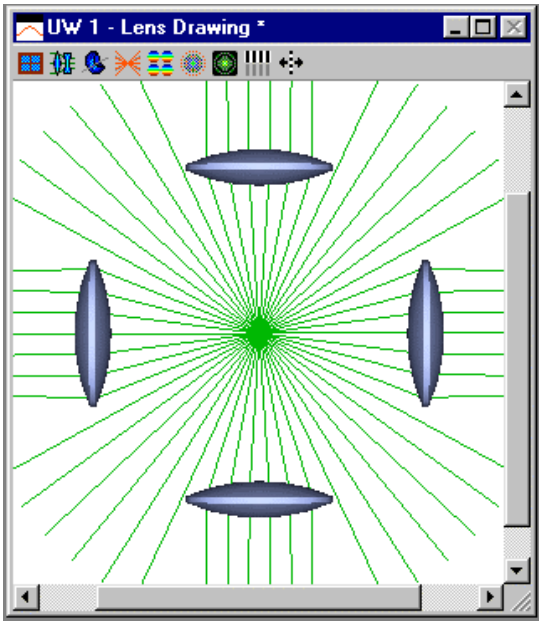
Note that we specify that ray output be provided in global coordinates. It is possible but not practical to use local coordinates inside a non-sequential group.

```
*TRACE RAY - GLOBAL COORDS REL TO SURF 1 - FBY  1.00, FBX  0.00, FBZ  0.00
 SRF        Y           X           Z           YANG        XANG         D
  1      14.750000      --          --       30.400000      --        7.463998
 16      15.000000      --        0.426115   55.344536 -6.4792e-16    0.494038
 17      18.681403 -2.8778e-17     2.971008  119.617171 -180.000000    4.475400
 14      23.539159 -5.9007e-17     0.209492   46.832682 -2.5760e-16    5.587823
 15      25.000000 -6.5168e-17     1.579746  -46.832682  1.4540e-15    2.002911
 17      23.734649 -3.5049e-17     2.766632 -118.864905  180.000000    1.734881
 14      18.761999  7.1330e-17     0.025555  -56.404938  1.6847e-15    5.678094
 16      15.000000  1.4481e-16     2.524554  -32.936309  8.7812e-16    4.516374
  6      14.474596  1.5724e-16     3.335577  -24.356236  4.1411e-16    0.966337
  9      13.353533  1.7514e-16     5.811974  -44.485322  4.8673e-16    2.718332
 22     -16.657159  4.3470e-16    36.366730  -44.485322  4.8673e-16   42.827967

 23      -9.321999  3.7126e-16    28.898590  -44.485322  4.8673e-16  -10.467936
 PUPIL      FY          FX          RAY AIMING                          OPD
           2.950000      --       CENTRAL REF RAY                   -6.1101e+04
```

The numeric output provides the information we seek. By comparing the surface numbers in the output listing with the surface data listing above, we see that after passing through the entry port, the ray strikes the bottom edge of the top element (surface 16), 0.426mm behind the vertex of the element. Since the sag of the surface at that point (according to Lens>>Show Auxiliary Data>>Surface Sag) is .4196, we conclude that the ray strikes the surface 6.4 microns from the corner.

Wheel of fortune

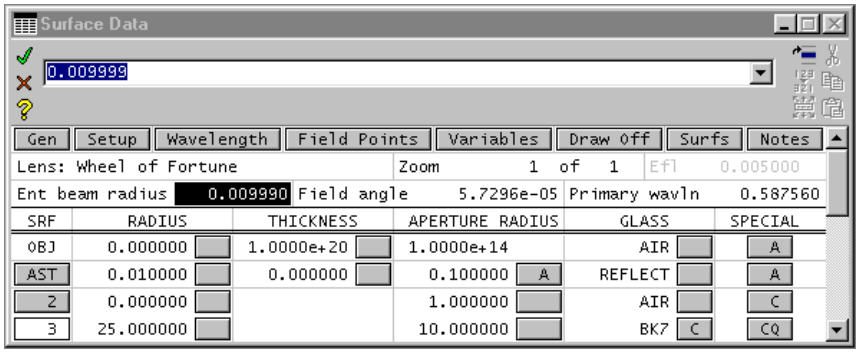
The Wheel of Fortune system is included as an example to illustrate that the entry port to a non-sequential group does not need to be physically outside the group. The system consists of four lenses surrounding a point source, as shown below.



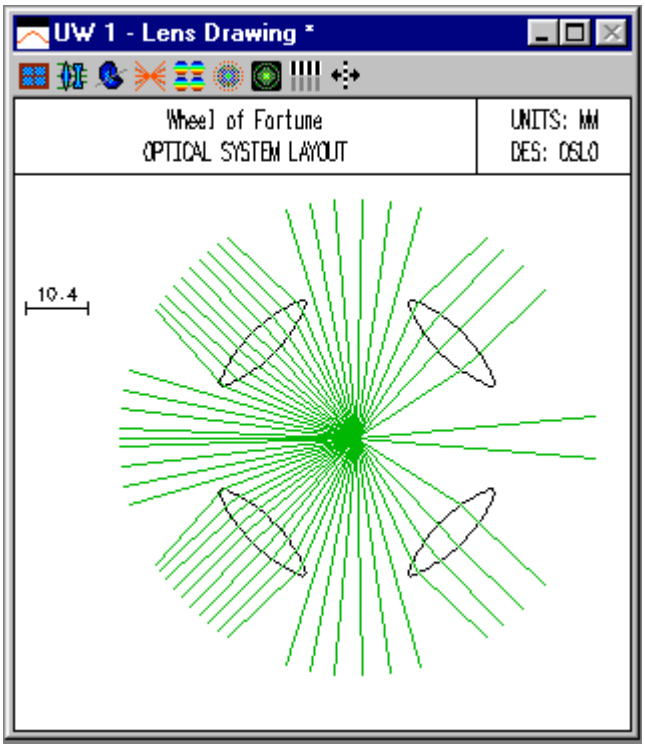
The coordinates data for the system is as follows. The right-hand lens is given by surfaces 2-3, the top lens by surface 4-5, the left-hand by 6-7, and the bottom by 8-9. The object surface is in the center of the group. The ray-aiming mode is set to Extended Aperture Mode in the Setup spreadsheet, with the beam half angle set to 180 degrees, so the source radiates into a full sphere. The exit port is a 40 mm radius complete sphere (designated by an aperture radius of 0.0) so that rays exiting at any angle will leave the system.

```
*TILT/DECENTER DATA
  2     DT    1          DCX        --        DCY        --        DCZ     20.000000
                         TLA        --        TLB        --        TLC        --
  3     DT    1          DCX        --        DCY        --        DCZ     25.000000
                         TLA        --        TLB        --        TLC        --
  4     DT    1          DCX        --        DCY     20.000000    DCZ        --
                         TLA     90.000000    TLB        --        TLC        --
  5     DT    1          DCX        --        DCY     25.000000    DCZ        --
                         TLA     90.000000    TLB        --        TLC        --
  6     DT    1          DCX        --        DCY        --        DCZ    -20.000000
                         TLA        --        TLB    180.000000    TLC        --
  7     DT    1          DCX        --        DCY        --        DCZ    -25.000000
                         TLA        --        TLB    180.000000    TLC        --
  8     DT    1          DCX        --        DCY    -20.000000    DCZ        --
                         TLA    -90.000000    TLB        --        TLC        --
  9     DT    1          DCX        --        DCY    -25.000000    DCZ        --
                         TLA    -90.000000    TLB        --        TLC        --
 10     DT    1          DCX        --        DCY        --        DCZ     40.000000
                         TLA        --        TLB        --        TLC        --
```

In this system, the object distance is zero (1e-20, actually), so the source is contained within the group. Another way to implement this sort of arrangement is to put the source outside, but place a mirror inside the group. In the system below, the entry port has been rotated by 45 degrees, the source is placed at infinity, and a 10-micron radius spherical mirror is placed at the center of the system. Ray aiming is restored to normal (Central reference ray).



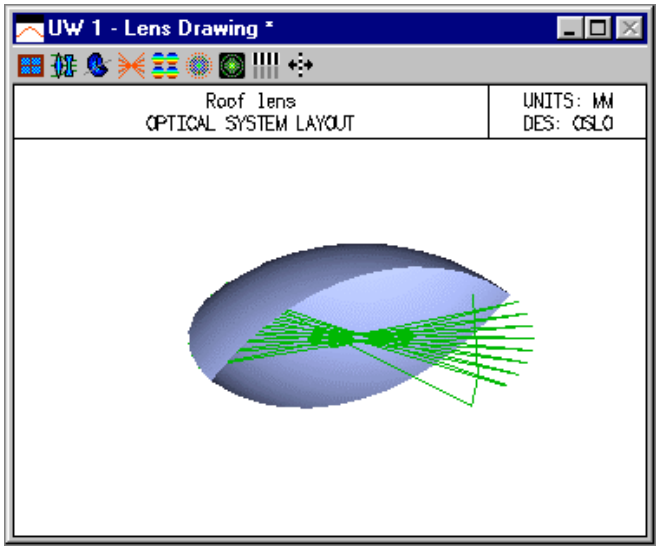
Light enters the system from the left and strikes the mirror in the center, where it is reflected and popagated in the non-sqeuential group until it escapes through the exit port. It is not possible to reflect rays completely in the forward direction because they reflect from the mirror. By making the entrance beam radius only slightly smaller than the mirror radius, however, we can reflect rays at almost grazing incidence so they are only slightly deviated. It is helpful to attach a slider to the entrance beam radius to experimentally determine the proper value. This can be accomplished by using a short sw_callback routine attaching the ebr to cc[0], similar to the technique used elsewhere in these examples to manipulate the field angle. The resulting ebr is somewhere around .00999, as shown above. With this value, the plan view of the system looks like the following.



In connection with this example, note that the forward-propagating rays coming from the tiny mirror are strange rays, in the sense of requiring an alternate surface intersection specification for the normal ray trace. However, there is no such thing as an alternate surface intersection point in the non-sequential trace; rays find the correct intersection point based on the shortest optical path.

Roof Lens

The roof lens is a test lens made from two convex surfaces that intersect at previsely 90 degrees at the aperture. The lens was cited by M. Hayford at the 1998 Optical Design Conference as one that tested the accuracy of non-sequential ray trace routines, because after several reflections from the roof, rays eventually wander off the correct trajectory due to round-off error in the ray trace calculations, and the roof fails to retro-reflect as it should.



Thin-film coatings

OSLO coating library

Optical thin films have numerous scientific, technological, and commercial applications over a wavelength range that extends from the x-ray to the submillimeter regions. They can be used to shape the spectral response of the light transmitted and reflected by the surfaces to which they are applied. Some of the generic spectral filters that are made of multilayer coatings include antireflection coatings, neutral beam splitters, reflectors, short- and long wavelength cut-off filters, narrow band transmittance filters, and polarizers. To achieve these effects, from one to many tens of layers may be required. From the point of view of a lens designer the most important filter types are antireflection coatings and reflecting coatings. However, there may be instances where it may be of interest for lens designers to perform preliminary order of magnitude calculations with some of the other types of multilayer coatings mentioned above. For this purpose a number of filter designs have been added to OSLO.

The multilayers can be constructed of dielectric layers only, or they can consist of a combination of both metallic and dielectric layers. With all-dielectric layer systems the incident light is either transmitted or reflected—there are no absorption losses. Furthermore, because the dispersion of the refractive indices of dielectric materials is relatively small, the spectral features can be tuned over a wide range of wave-lengths simply by scaling the thicknesses of all the layers by the same amount. This is true only if the new position of the spectral feature of interest lies within the transparency range of the coating materials. Furthermore, for a given angle of incidence and plane of polarization, the transmittance T and the reflectance R of an all-dielectric layer system are independent of the direction of the incident light and they obey the relation $T + R = 1.0$.

If there are absorbing films in the multilayer system, some of the light incident on the multilayer will be absorbed . The condition $T + R + A = 1.0$ holds, where A represents the absorbtance. The scaling of the layer thicknesses in order to shift the features in the spectrum is no longer so simple. First, the dispersion of the optical constants of metals are much more pronounced than that of dielectric coating materials. Second, because the extinction coefficients of metals are large, the changes in the thicknesses of the metal layers must be much smaller. Lastly, whilst the transmittance of a multilayer coating with absorbing layers is independent of the direction of the incident light, in general this is not true for the reflectance and absorbtance.

The properties of both all-dielectric and metal-dielectric layer systems depend on the angle of incidence and on the state of polarization of the incident light. In general, the spectral features of multilayer coatings shift towards shorter wavelengths with an increasing angle of incidence. In addition, for angles of incidence greater than 10 or 15 degrees, a marked polarization splitting can usually be observed. For unpolarized light this results in a general broadening of the spectral features. This has important implications for lens designers. For best results, any filter in which there are sharp transitions from high to low transmittance or reflectance should be located within that part of a lens system in which the angles of incidence and the convergence angles are as small as possible.

Coating material considerations

The optical constants of thin films can depend on the actual process used for their deposition. Films produced by thermal evaporation and by conventional electron beam gun evaporation can be quite porous. Although heating of the substrate can result in a reduction of the porosity, it rarely results in completely dense films. The spectral features of coatings produced in this way frequently shift towards longer wavelengths as water vapor is adsorbed by the pores. For some applications it is possible to predict sufficiently accurately the changes that will occur on exposure of the multilayer to the atmosphere. For other, more stringent applications, it is necessary to produce very stable multilayer systems that do not age at all with time and exposure to moisture. Higher energy deposition processes, such as ion assisted electron beam gun evaporation, ion plating, ion sputtering or magnetron sputtering, yield dense coatings that meet these requirements. However, either the equipment is more expensive, or the deposition process is slower and so, as a rule, a premium has to be paid for such coatings.

Dielectrics can be classified into soft and hard coating materials. The former can be deposited by thermal evaporation at relatively low temperatures and are frequently protected by a cemented cover glass from damage due to abrasion. Hard coating materials are deposited by electron beam gun evaporation, or by sputtering. They are much harder and are quite suitable for front surface mirrors.

The optical properties of metals are even more sensitive to the deposition process than those of dielectric layers. This is especially true for partially transparent metal layers used in beam splitters and in certain advanced multilayer coatings.

The conclusion from the above is that, until a deposition process is decided upon, it is difficult to predict what optical constants should be used for the design of the multilayer. It is customary to use approximate optical constants for preliminary designs. Although it is frequently sufficient for this purpose to use non-dispersive refractive indices for the dielectric layers, the dispersion of the optical constants of metals must be taken into account. A good source of information on the optical constants of metals is the “Handbook of Optical Materials” vols. I, II edited by Palik [1,2]. Any systems designed in this way will require only slight modifications of the thicknesses of the layers to allow for the discrepancy between calculated and experimental optical constants.

Some sample multilayer systems

Some of the multilayer systems presented below are based on multiples of quarter wave layers which are easy to monitor by optical means. These systems are best represented by a notation in which H and L correspond to high and low refractive index layers of quarter wave optical thickness, respectively. Thus, for example, (HL)^2 H is the same as HLHLH, which represents a five layer quarter wave stack. (HL)^3 (LH)^3 is the same as HLHLH2LHLHLH, which represents an eleven layer narrow band filter in which the central layer is a half wave layer of low refractive index, etc.

Other systems consist of layers with thicknesses that depart from quarter wave thicknesses significantly. The additional degrees of freedom available in such refined systems can be used to optimize the performance of the multilayer. The thicknesses of such layers are frequently monitored using quartz crystal monitors.

The all-dielectric multilayer systems listed in the table below, with the exception of two systems, are constructed out of two coating materials only with non-dispersive refractive indices 1.45 and 2.35. These values are not too far removed from the refractive indices of the soft coating material pair MgF2 and ZnS or from the hard coating material pair SiO2 and Nb2O5. The optical constants of aluminum were taken from Palik. Inconel alloy constants were measured at NRCC. The substrate and incident medium materials in all systems are BK7 glass or air. M in system 2 stands for a quarter wave layer of medium refractive index of 1.7.

No.	Name	Type	Layers	Description
1	AR_1	single layer MgF2 AR coating	1	glass/L/air
2	AR_2	quarter-half-quarter AR coating	3	glass/M2HL/air
3	AR_3	narrow band AR coating	2	glass/optimized/air
4	AR_4	wide band AR coating	7	glass/optimized/air
5	R_1	quarter wave stack reflector	13	glass/(HL)^6 H/air
6	R_2	opaque Al layer reflector	1	opaque Al /air
7	SP_1	short wavelength pass filter	17	optimized, between glass
8	LP_1	long wavelength pass filter	17	optimized, between glass
9	BS_1	Inconel layer beam splitter	1	0.0125 μm of Ag cemented between two 45 deg prisms
10	BS_2	multilayer beam splitter	7	optimized, between two 45 deg prisms of BK7 glass

11	NB_1	narrow band filter (1 cavity)	15	glass/(HL)^4 (LH)^4 /glass
12	NB_2	narrow band filter (2 cavities)	31	glass/optimized/glass

A more detailed explanation of the theory of optical thin films will be found in the excellent book by Macleod [3]. For more information on classical and less usual applications of optical thin film coatings the interested reader is referred to references [4,5].

References

1. E. D. Palik, Handbook of Optical Constants of Solids I (Academic Press Inc., Orlando, 1985).

2. E. D. Palik, Handbook of Optical Constants of Solids II (Academic Press Inc., Boston, 1991).

3. H. A. Macleod, Thin Film Optical Filters (McGraw Hill, New York, 1986).

4. J. A. Dobrowolski, “Optical Properties of Films and Coatings,” in Handbook of Optics (Editor-in-Chief,. M. Bass) (McGraw-Hill, New York, 1995), pp. 42.1-42.130.

5. J. A. Dobrowolski, “Usual and Unusual Applications of Optical Thin Films-An Introduction,” in Thin Films for Optical Coatings (Eds. R. F. Hummel and K. H. Guenther) (CRC Press, Inc., Boca Raton, Florida, 1995), pp. 5-35.

Chapter 11 Glossary

The lens is the document

Writing a document consists of expressing information in words, pictures, etc. that describe the subject matter. Designing a lens consists of expressing information in numbers, words, drawings, etc. that describe the lens. The essence of both is the creation of information that is expressed in language. A book isn't done until the information is expressed; a lens design isn't either. A word processor helps the author create his document; OSLO helps the optical designer create his lens. In the new metaphor, the *lens is the document*.

In the early days of lens design, the prescription gave the curvatures, thicknesses, and glasses, plus the aperture, field, and wavelengths, just enough information to trace rays through it. Now, an overall design contains a great deal more information, including tolerances, coating specifications, and manufacturing details such as polishing grades, bevels, and cosmetic requirements. Although OSLO does not yet deal with all these aspects of a design, its data structure is able to contain much more than the simple lens prescription.

To make best use of OSLO you should understand the basic nature of the data used and produced by the program. OSLO data can be divided into four major groups: surface data, operating conditions, preferences, and the Spreadsheet Buffer. A fifth group would include miscellaneous items such as command arguments, fixed defaults, and various temporary data used internally.

Surface data

Surface data include the usual lens prescription items such as radii of curvature, thicknesses, glasses, aspheric constants, etc. Surfaces are numbered, starting with 0 for the object surface. In a normal, sequential lens, the surface numbers increase monotonically in the order that rays strike them.

There are four ways to enter and modify surface data. The first, and most commonly used, is the Surface data spreadsheet, which can be invoked from the Update menu, or from the F5 toolbar icon. The second is called global editing, in which the data item, surface number, and value are entered on the command line. The third is the internal lens editor, in which the program enters a mode that accepts data for a particular surface. Finally, surface data can be entered as a text file prepared “off-line” using any editor. Although the spreadsheet is very convenient for typical design sessions, the other methods each have a *raison d’etre*.

Operating conditions

Operating conditions include items that pertain to the whole lens, or describe its conditions of use. The stereotype operating conditions are the ones used to specify the aperture and field of view. OSLO includes many other operating conditions, some of which (e.g. wavelengths, error function) have traditionally been included as operating conditions in all optical design programs. Other OSLO operating conditions are new, and have been included to more completely describe the complete design (e.g. element drawing data), to enable consistent analysis (e.g. spot diagram operating conditions), or to eliminate extensive data re-entry (e.g. lens drawing operating conditions).

An important thing to remember about operating conditions is that they are attached to the lens data, and are saved in lens files with the surface data. This simplifies the design process. If you set up, for example, a certain set of rays to be shown on a lens drawing, then whenever you make a drawing that includes the default rays, your choices for that particular lens will be used.

Preferences

Preferences are to be distinguished from operating conditions. Preferences are attached to the program, and remain the same for all lenses. Examples include items that affect the appearance of

graphics windows, number formatting, and the like. Most preferences take effect as soon as they are set, but some require that the program be restarted. These include the maximum number of surfaces, spot diagram rays, and wavelengths that the program can use, as well as the selection of fonts. Once a preference has been set, it is saved in a configuration file (*.ini) that is read on startup, so the settings are preserved from one session to another.

The Spreadsheet Buffer

The Spreadsheet Buffer is an important part of the OSLO data structure. So far as OSLO is concerned, the spreadsheet buffer is an output-only data structure, but it serves as the principal source of communication between the program and user macro commands (called star commands, see below). A spreadsheet buffer is attached to each text window. Whenever OSLO writes floating-point numeric output to the window, a copy of the output, having full internal precision, is placed in the spreadsheet buffer. The various buffer elements can be addressed by the SCP and CCL macro languages, used in data entry, or displayed in the message area by clicking on the number in the text window.

Star commands

Although OSLO appears to the casual user as an ordinary windows application, it is actually a combination of an application and a macro language. Some of the commands supplied with the program are part of the core program, while others are written in the macro language. It is not normally observable to the user which is which. There are two macro languages in OSLO. OSLO Light contains SCP, an interpreted language. OSLO PRO and OSLO SIX also contain CCL, a compiled language. Both languages use C syntax.

Macro commands written in SCP are executed by preceding the command name with an asterisk, hence the name “Star command”.

Advanced users can write their own star commands and integrate them with ones supplied by Sinclair Optics, because the menus supplied with OSLO can be changed by the user. Since these commands are often required only to support special needs, they do not need to have the scope that internal commands do, so it is very easy to add features to the program.

Click/Command Interface

As mentioned above, it is possible to add star commands to OSLO. These commands can either be added to the menu system, or executed directly by typing the command name into the command line. Although many Windows programs boast of “no commands to learn”, all programs are based on commands. The question is whether the commands can be directly accessed by the user. OSLO retains the traditional command interface as an option for the user. This provides a highly efficient way to use the program, and the name Click/Command indicates that at any point, the user can choose either to enter a command, or click on an icon or menu with the mouse.

It is possible to enter commands while a spreadsheet is open. OSLO distinguishes between commands and data, and carries out the appropriate action. For this reason, the cells on OSLO spreadsheets are called SmartCells.

Interactive operation

OSLO is fundamentally an interactive program. After each command, the requisite action is performed and the display is updated. This means, for example, that the surface data spreadsheet is always up to date. There are a few exceptions, mostly relating to entering optimization data, but for the most part OSLO is based on an interactive model. Resulting from this model are the concepts of a current wavelength, configuration, object point, and spot diagram. Once these quantities are set, they remain set until changed by the user, or by some command that resets them to a different value. This increases the speed and efficiency of the program. In the case of the spot diagram, for example, once a spot diagram has been computed, a wide range of analyses can be carried out without retracing the rays.

Prominent examples of the interactive organization of the program are the so-called interactive design windows used in OSLO. In the interactive design windows, lens parameters are attached to graphic sliders. Whenever a slider is changed by the user, the lens parameter changes, and OSLO

updates a particular evaluation function (ray trace, spot diagram, etc.). The user senses that the program is responding in “real time”.

Index

*

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z