Designing Optics Using Zemax OpticStudio®

Donald C. O'Shea
Julie L. Bentley

SPIE PRESS
Bellingham, Washington USA

Library of Congress Control Number: 2023946503

Published by
SPIE
P.O. Box 10
Bellingham, Washington 98227-0010 USA
Phone: +1 360.676.3290
Fax: +1 360.647.1445
Email: books@spie.org
Web: http://spie.org

Printed in the United States of America.
First Printing 2024.
For updates to this book, visit http://spie.org and type "PM367" in the search field.

SPIE.

Contents

Preface

The purpose of this text is to show you how to design an optical system using the optical design program Zemax OpticStudio®. The complete design process (from lens definition to tolerancing) will be developed and illustrated using the program. The text is organized so that a reader will be able to (1) reproduce each step of the process, including the plots for evaluating lens performance, and to (2) understand their significance in producing a final design.

This text is not a user's manual for Zemax OpticStudio (there are on-line reference guides for that). Rather, the text starts with a single lens to demonstrate the laws of geometrical optics and illustrate basic optical errors (aberrations) using Zemax OpticStudio. Then, through a series of examples and exercises, you can follow each step in the design process using Zemax OpticStudio to analyze and optimize the system to meet the required performance specifications. Once the nominal design meets these specifications, you can determine a set of tolerances that permits a large fraction of them to be manufactured with an acceptable as-built performance.

Although it is assumed that readers will follow the examples in the text and reproduce the results, you are encouraged to use them as jumping off points for an exploration of the designs. In addition to exercises with answers, we have added toward the end of the text what we call "Explorations"—open-ended problems with several possible directions in which to explore the design space. But this exploration needn't be confined to the final chapters. If there is a design feature or strategy that piques your curiosity and you want to find out what happens when you make a change in the design, go ahead and explore the consequences. You can't break anything. However, remember to save your lens before you begin to tinker with things.

One problem that will occur while exploring these various designs is maintaining a record of your work. Too often, a designer, trying out a new design or modifying a current design, can lose track of the performance and results of earlier attempts. Because Zemax OpticStudio does not currently contain a lab book or journal feature, it is highly recommended that during your design session a journal application be used to record your comments on progress, any data, and plots. This provides you with a record of your progress during a session. A journal application should capture anything typed or pasted into it and provide automatic backup so that users do not have to worry about saving the records. One of us (O'Shea), who runs Zemax OpticStudio on a Windows emulator, uses MacJournal as his journal app.

This text is written for a reader to continually interact with Zemax OpticStudio. Although any commercial optical design software can provide the tools to enter and modify designs, each program has its own interface, and it is not possible to demonstrate important optical principles with every program in a single text. For those who do not have immediate access to Zemax OpticStudio, there are two possible ways to use this text. If you're a student connected with a college or university, there are student licenses available for Zemax OpticStudio. For those who have access to other design programs, the operations and data entry may differ, but most of them will contain the same plotting, evaluation, and optimization functions as Zemax OpticStudio. So, with some translation, it should be possible to demonstrate the same operations as those used in this text.

We hope that this text will engage your curiosity and provide directions that will encourage you to work through all of our examples and then continue exploring optical design on your own. Designing optics is much like a game, where the rules are laid down by the laws of physics, where the pieces are surfaces, airspaces, and glass, where aberrations are obstacles to be overcome, and where the goals are set by the practical requirements of a design. Have fun playing!

Acknowledgments

We would like to express our sincere thanks to both Hsiu-An Lin and Kevin Scales from Zemax/Ansys's customer support team and to University of Rochester students Ankur Desai and Owen Lynch for reviewing the entire text and doing all of the examples. Their suggestions and feedback were invaluable in helping us provide accurate descriptions of the program and its uses.

Donald C. O'Shea
Atlanta, Georgia

Julie L. Bentley
Rochester, New York

December 2023

Introduction

Our approach is different from most texts on optical design because the operations and analyses in an optical design software package (Zemax OpticStudio) are described in great detail so that you can use the software to duplicate our examples. Zemax OpticStudio uses a Graphical User Interface (GUI) to input data, execute various analyses, and plot results. It is best to perform the operations as they are described in the text. From these results, you will be able to follow our discussion, familiarize yourself with program operations, and understand the information created there. "Humming along" by just reading the text won't do that.

To simplify the number of figures needed to show a particular GUI operation, we have chosen to use a shorthand notation for the GUI navigation. For example, to start a new design, File > New is used. This path format tells you to click on the File tab and select New from the available menu icons. The light blue background behind the text and the "greater than" character tells you that this is a Zemax OpticStudio operation, and this is the path you use to complete it. Additionally, many operations involve opening a new window, pushing a button, or selecting a drop-down menu option. These actions will be given a gray background, such as Select Preset.

Zemax OpticStudio uses an optical shop specification sheet format for its plots depicting lens performance. Most of these plots were intended for letter-size (8.5″ × 11″) paper in a landscape format and have a frame containing additional information such as the date that the plot was created, the name of the lens file used for the plot, and some significant analysis values for that plot. Simply reducing these plots to fit in a 5-inch-wide area for this text can't be done without a loss in detail and legibility. Because this text relies heavily on these plots to convey information, we have reformatted them. We eliminated the outside frame, extracted the figures in vector format, enlarged them, and added legible labels for placement within the figure. For those who have been using Zemax OpticStudio for some time, the switch between its native plots and our extracted versions may be disconcerting, but we believe our work ("It ain't easy!") adds to our descriptions and explanations on designing optics. Examples of modifications to the native OpticStudio graphic output are shown below for the lens data tables, lens cross-sections, transverse ray plots, and spot diagrams.

Modifications to the Lens Data Editor (LDE)

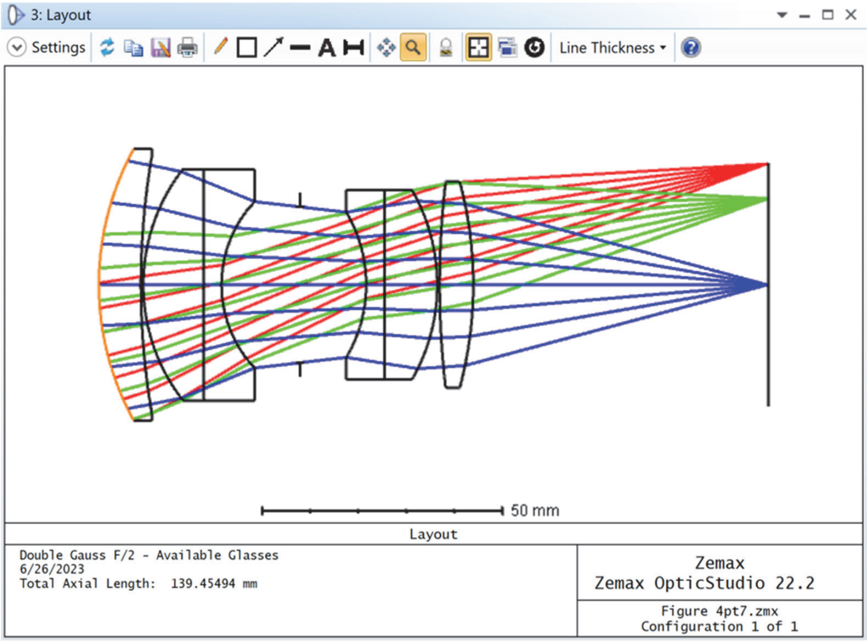
By default, the Lens Data Editor has a large number of columns, many of which only contain data for special types of lenses not used in this text. As shown below, we have hidden the lens parameters that are not needed for our discussion (e.g., Coating, Chip zone, Conic, and TCE), resulting in a much more compact display for the LDEs shown in this text. We've also added a title bar to the bottom of the LDE that lists the title of the lens and the settings for the aperture, field, and wavelength of the optical system that are entered in the System Explorer.

	Surface Type		Comment	Radius	Thickness	Material	Clear Semi-Dia
0	OBJECT	Standard ▾		Infinity	Infinity		0.000
1	STOP	Standard ▾		120.000	5.000	N-BK7	10.000
2		Standard ▾		-50.000	0.000		9.898
3	IMAGE	Standard ▾		Infinity	-		9.748

Title (OSlens); EPD (20 mm); Field (0°); Wavelength (d-line)

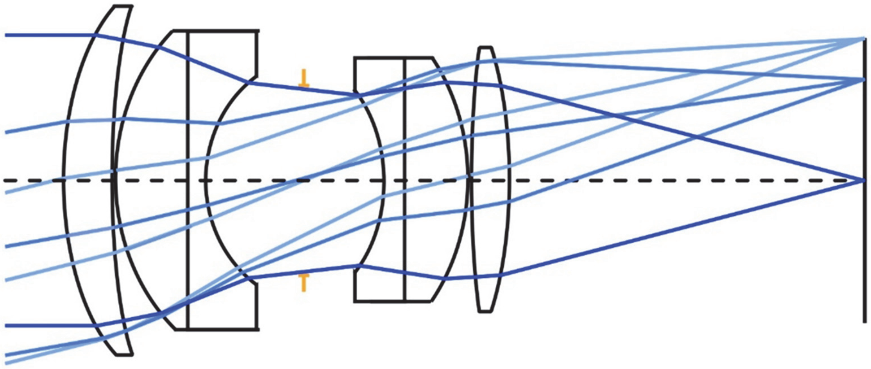
Modifications to Lens Cross-sections (Layouts)

The default lens cross-section for a double-Gauss lens is shown below.



For comparison, an example of our modified cross-section for that same lens is shown on the next page. Some obvious differences can be seen when comparing the two figures. First, the outside frame has been suppressed to save space.

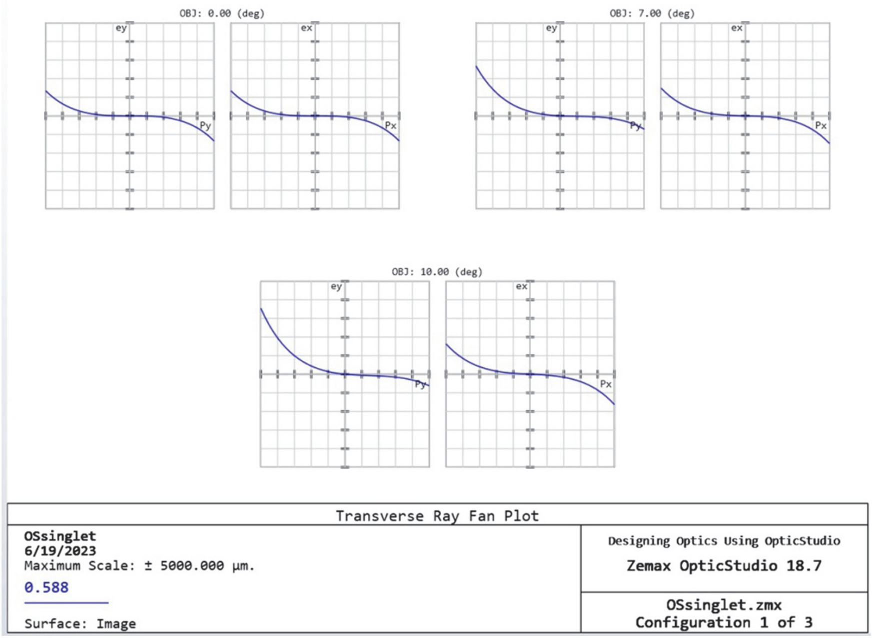
Second, to provide an immediately recognizable reference within the plot, the optical axis ray has been replaced by a black dashed line, designating the optical axis of the lens. Third, the default colors of the rays (blue, red, and green) for each field have been replaced by different shades of blue (the on-axis field is the darkest).



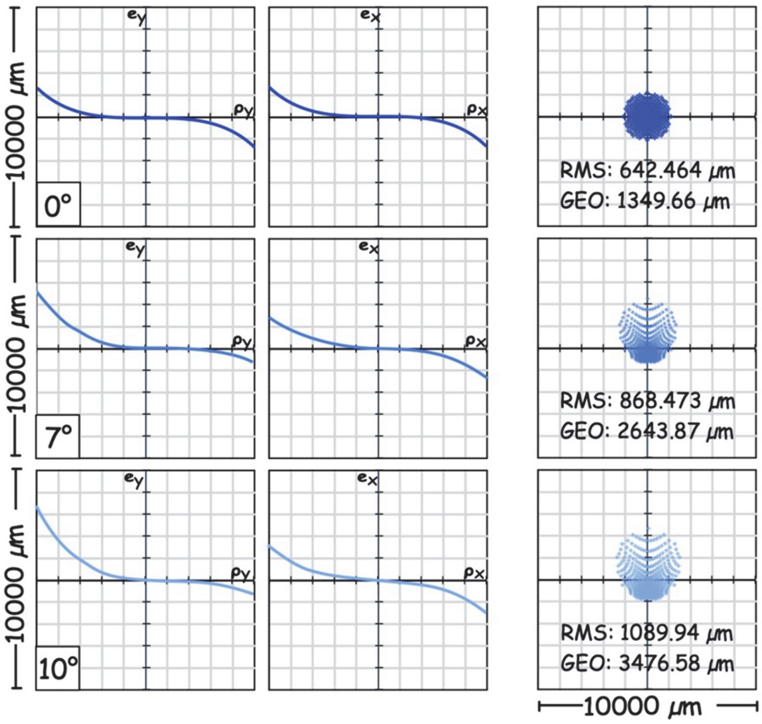
The other more subtle differences that you may notice when comparing the original figure to the modified figure is that in the default cross-section you can't see the rays coming into the lens (this occurs when the object is at infinity) and each lens is only drawn as big as the largest ray height at the lens. Many users find it useful to increase the viewed diameter of the lens elements and add an offset surface to observe the rays in object space when looking at these cross-section lens plots. Chapter 3 describes how to make these changes to your lens cross-sections. Note: The majority of the lens cross-sections in this text will be created with these two modifications in place even though we may not explicitly show the offset surface in the LDE and/or state that a specific aperture increment has been added to the lens diameters to create the figures.

Modifications to Other Graphic Output Plots

Other graphic output plots have been modified in a similar manner (primarily, to save space in the text). For example, the default plot of an array of ray fan diagrams for a lens with three field angles is shown on the next page. Note that there is a great deal of white space, and the labels for each field and the Maximum Scale are barely readable. The ray fans are shown in a landscape format with the on-axis field in the upper left corner and the full field at the bottom.



In this text, these plots are reformatted with the ray fan diagrams stacked vertically, with the on-axis field at the top. They are often placed next to the spot diagrams to save space, as shown below. This is Fig. 7.2 in the text.



The ray fan plots have been squared up, and the e_x and p_x, e_y and p_y axis labels that Zemax OpticStudio uses are retained. The vertical flat-ended arrow displays the range of error (units are included). In a similar manner, plots for field curvature, distortion, color aberration, and other performance plots have been extracted from their Zemax OpticStudio versions and replotted to increase detail and readable values.

A Note on Ray Aiming

Zemax OpticStudio uses an algorithm, called ray aiming, which determines the rays at the object that correctly fill the stop surface for each field for a given stop size. Generally, ray aiming is only required when the image of the stop as seen from object space is considerably aberrated, shifted, or tilted. Ray aiming can be turned on in the Ray Aiming tab of the system explorer SysEx > Ray Aiming > Paraxial by selecting Paraxial in the drop-down menu. This is discussed in detail in Chapter 5. Paraxial ray aiming is used for all examples and demonstrations from Chapter 5 to the end of the text. Since the default setting for ray aiming is ray aiming "off" when you first start Zemax OpticStudio, if you notice small differences between your results and ours, check your ray aiming setting—if it is off, turn it on!

Comments by Nicholas Herringer (Ansys, formerly Zemax)

Ansys and our optics-focused teams are excited to support Don O'Shea and Julie Bentley in the publication of this text. Through this book, it is our great pleasure to contribute to the education of both the next generation of optical designers as well as seasoned veterans. Over Zemax's thirty-year history, the field of optics has become an exciting scientific domain, fueled by innovation and simulation. We look forward to continuing that journey with you in the decades to come.

Note that this text was created with Zemax OpticStudio versions 22.2 and 23.1. One author (Bentley) ran it on a PC; the other author (O'Shea) ran it on a MacBook Pro in emulation with Parallels 16.5 to run first on Windows 7 and later on Windows 10. In the spirit of continual improvement, Ansys is regularly enhancing and updating OpticStudio; over time, some features and functionality may change, including compatibility with certain hardware configurations. If you are using a more recent version of Zemax OpticStudio than was used in this text and notice a difference in output from the examples, please contact Zemax.Support@ansys.com. Our support team will be happy to discuss with you any questions or concerns you may have.

Chapter 1
The Basics

Until recently, there was only one application for ray tracing, the modeling and analysis of an optical system. However, today's students have a different application in mind. They think that they are going to be taught how to use powerful computers to generate realistic scenes like those that animation studios use to create movies (see Fig. 1.1).

Both applications, optical system modeling and realistic scene generation, simulate rays traveling through space to create images. Some of these images may be very simple (such as a point or a line), while those in computer-generated images (CGI) are extremely complex.

In the latter case, thousands of rays are traced to build the image of a scene. To reduce the time to compute the scene, only those rays that will reach the eye are traced. The easiest way to do this is to trace the rays in reverse. Starting at an eye or a camera sensor, a ray is traced from a point on the sensor, through the lens, and out to the scene, where its intersections with the surfaces defined by the computer model of the scene reflect, refract, and scatter the light in the scene.

Figure 1.1 "Bosque Morning," by John Fowler. Reprinted from Wikimedia under the Creative Commons Attribution 20. Generic license.

Chapter 2
Rays and Ray Sketching

The first thing you do when you start a new lens in an optical design program is to enter the parameters that describe its design. But how can you tell if you entered the design correctly? How do you determine if the results are correct? And how can you gain some sense of how well your lens performs? These questions express the daily concerns of all optical designers. Some of the answers come from an understanding of how a system *should* operate, some from experience, and some from common sense. This chapter is intended to give you some tools to help answer these questions. When it comes to experience, that is something you will have to gain on your own.

2.1 Collimation

A lens (composed of a single element or multiple elements) separates the design space into **object space** and **image space**. Object space is also referred to as the **front** of the lens and image space as the **back** of the lens (Fig. 2.1). One of the reasons for doing so is that a lens has two focal points, and it is useful to be able to distinguish between them as the front and back focal points.

Both of the figures in Fig. 2.1 illustrate **collimated light**, a bundle of light whose rays are parallel to each other. Collimated light is sometimes referred to as parallel light. This concept is useful because it provides a simple way to represent light from a distant source, which is usually described as a source at infinity. (A laser beam that can travel long distances without much broadening (divergence) may also be modeled as a collimated beam but is usually simulated using physical optics concepts rather than geometrical optics.) In reality, collimation is only an approximation, but it simplifies the problem, and we will take advantage of that.

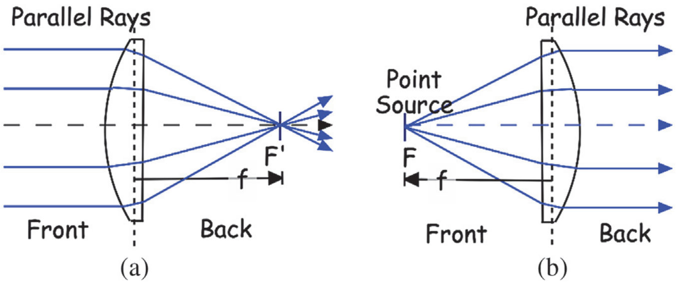


Figure 2.1 Collimation: (a) Collimated light focuses at the back focal point F′. (b) A point source at the front focal point F produces collimated light.

First Hiatus
Ledgers to Laptops

A hiatus is an interruption. The word is used to describe a break in the continuity of a work, a series, or some action. When a TV network says that your favorite TV series is "on hiatus," there is the prospect that at some future time the series will resume, and its fans will be pleased (which, of course, hardly ever happens).

In optics, the hiatus (Fig. H1.1) is the distance between two lens planes, the principal planes. (These will be discussed in Chapter 4.) When an axial ray enters a lens and hits the front principal plane P_F, it jumps across the hiatus at the same height to the back principal plane P_B and exits the lens.

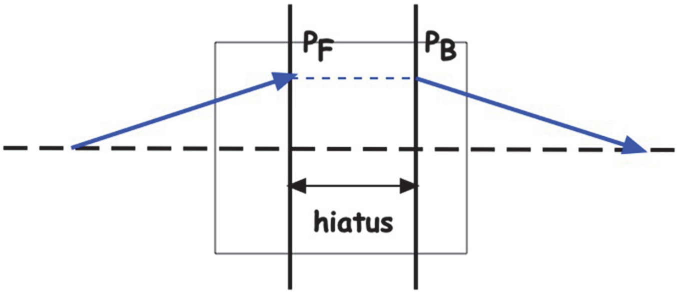


Figure H1.1 Illustration of "hiatus" in optics. Hiatus is the distance between the two principal planes of the lens, P_F and P_B.

So, this section of the text is an interruption. It is not directly related to what has just been discussed and what follows it. However, for some readers, the information provided here will give you a better sense of the development of optical design. It will also provide the tools and ideas needed to design and perfect an optical design. And it will give an appreciation of the work that others have done in this field.

H1.1 Simulations

The calculation of results began once humans learned how to quantify things. And once it was possible to describe some physical process using an equation, it was possible to model the process so that certain rules of thumb gained through much trial and error could be replaced by mathematical calculations. Thus began the practice of simulating these processes to be able to predict an outcome without performing the act itself.

Chapter 3
How to Put a Lens in a Computer

To trace rays with a computer, we start by entering the imaging system into the design program. The basic layout of OpticStudio's® graphic user interface (GUI) is shown in Fig. 3.1. The menu banner at the top of the main window has a series of menu tabs (e.g., File, Setup, Analyze, Optimize, and Tolerancing) used to enter, analyze, refine, and tolerance a design. Each menu tab has a row of icons to indicate the available functions and actions for that tab. Hovering your cursor over one of the icons opens a window showing the icon name and a short explanation of its function or action. In the next few chapters, we will focus on the File and Setup tabs. Analyze will be discussed in Chapters 6–9, Optimize in Chapters 10–12, and Tolerancing in Chapter 13. The last tab, the Help tab, provides links to OpticStudio resources.

The area below the menu banner is divided into the System Explorer on the left and a display space on the right for the Lens Data Editor (and any other editors or analysis windows). At the end of the Setup tab is a series of icons that help you control how these windows are displayed in the workspace (e.g., tiled, floating, or docked). At the very bottom of the program window is a status bar containing four-letter operands (discussed in Chapters 10–13) whose values update as the design is changed.

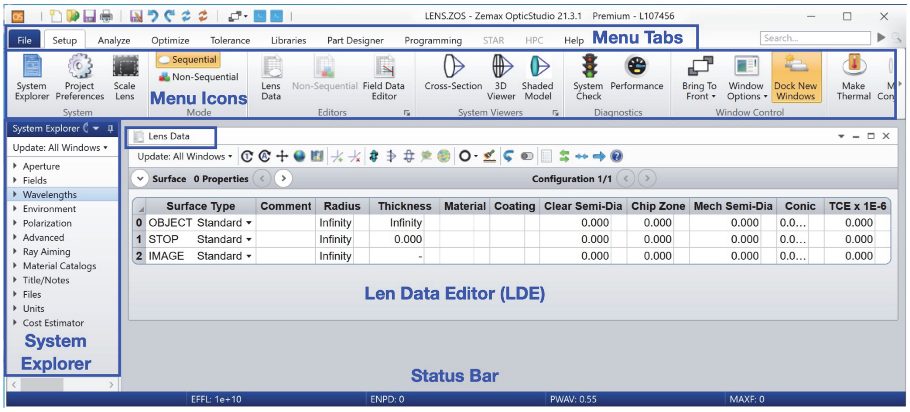


Figure 3.1 The basic layout of OpticStudio's graphic user interface.

3.2 Prescription Data

Now that the system data has been entered in the System Explorer, we can turn our attention to the spreadsheet titled Lens Data. There are a number of such spreadsheets in OpticStudio that are referred to as *Editors.* To distinguish the current spreadsheet from the others, we will refer to it as the Lens Data Editor or LDE for short. The initial LDE for a new lens consists of three rows and many columns. Each row represents a surface in the optical design with the data for that surface in the individual columns. For the purposes of this text, only six of the columns (Surface Type, Comment, Radius, Thickness, Material, and Clear Semi-Dia) will be shown in all LDE figures. The rest of the columns contain lens parameters for special types of lenses (e.g., aspheres), coatings, and mechanics that are not needed for our discussion, resulting in a much more compact display.

Figure 3.3 shows the current LDE with the three initial surfaces of any new lens: OBJECT, STOP, and IMAGE. The first row lists the values for the object surface. It is a planar surface (its radius of curvature is infinite) that is located at infinity (its thickness is infinity). The Material cell is blank, indicating that no glass has been assigned to object space. The default medium is air, and the refractive index is set to unity. The next term in the surface number column, STOP, may not be familiar. The designation of this surface as a stop will be the subject for considerable discussion later (Section 5.3). For now, consider it to be an important surface needed for a complete analysis of the lens. The last row in the LDE is always the image surface. In summary, there are three required surfaces that each design must have: object, stop, and image.

Typically, we need at least one more surface in a design (e.g., for a lens element we need a front and back surface). To insert a surface after the first (stop) surface, select the stop surface (S1) row and right-click on it. This opens a drop-down menu, where you can select Insert Surface After. When this action is taken, a new row with a "2" on the left side of the LDE will appear between the stop and the image surface, as shown in Fig. 3.4. New surfaces can also be added (or deleted) quickly by clicking on an existing surface row and using the insert and delete keys on your keyboard. The Insert key enters a surface **before** the current surface, while Ctrl-Insert inserts a surface **after** the current surface.

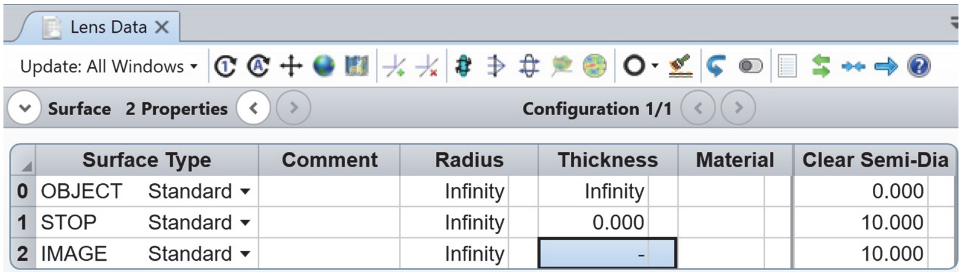


Figure 3.3 The lens "spreadsheet" (LDE) for a new lens with an EPD = 20 mm.

	Surface Type		Comment	Radius	Thickness	Material	Clear Semi-Dia
0	OBJECT	Standard ▾		Infinity	Infinity		0.000
1	STOP	Standard ▾		Infinity	0.000		10.000
2		Standard ▾		Infinity	0.000		10.000
3	IMAGE	Standard ▾		Infinity	-		10.000

Figure 3.4 Second surface inserted in the LDE.

This initial framework is used to build a complex design by adding surfaces and then entering the quantities that describe each of the surfaces and the spaces that separate them. Before entering a complete lens, we will first describe the inputs for each of the columns in the LDE, beginning with Surface Type. This column describes the form of the surface. The simplest surface, Standard, is based on a conic section whose vertex radius is given in the next column and by default has a zero conic constant (resulting in a simple spherical surface). For some surfaces, the shape is more elaborate, such as a complex asphere described by a polynomial. Expect to see Standard as the surface type for all designs in this text.

We can now specify the quantities that describe the makeup of the lens elements (radii of curvatures, thicknesses, and glasses) in the order that they are encountered by light from an object. The radius and thickness values are signed and determined by the coordinate system whose positive z axis is in the direction of initial light propagation (left to right, see Section 1.1). One odd thing about this coordinate system is that it is really a set of many coordinate systems with origins along the z axis. Each defined surface has an accompanying coordinate system whose origin is at the intersection of the surface and the optical axis.

The lens values, labeled with a surface number k, are thickness Tk, radius of curvature Rk, and glass Gk, starting with the object plane and continuing to the image plane. The values are entered into their corresponding cells in the LDE. References to the surfaces of the lens in the text will be labeled as Sk. In OpticStudio, the quantities related to the object and image planes have letter labels (O and I, respectively). The surfaces between them have numerical subscripts starting with 1, as shown in Fig. 3.5.

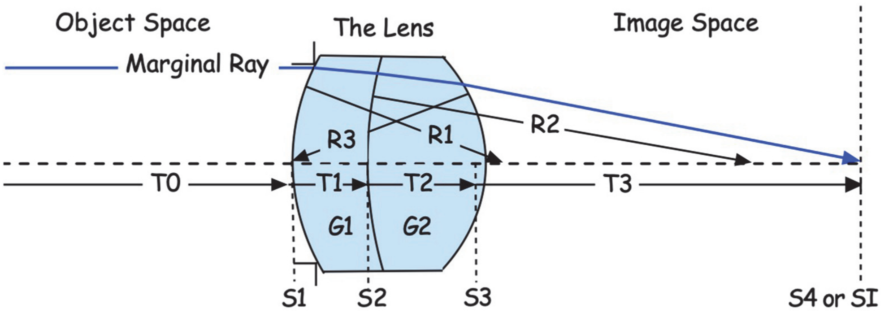


Figure 3.5 Describing a lens.

The R1 cell in the LDE contains the value of the first radius of curvature, R_1. Once the shape of the surface is defined, the distance to the next surface is entered. As was noted in Chapter 1, the sign of a thickness t to the left of a lens surface is negative and that to the right is positive. Unless a reflective surface is inserted, the lens thickness and the separating spaces are positive numbers entered in the Thickness column. So, typing 5 into T1 enters a thickness of 5 mm between the first and second surfaces.

Beyond the last lens surface is the image surface at a distance T3 or TI, where the image produced by the lens is located and analyzed. One other feature of this lens drawing is the blue ray entering at the edge of the lens. This is a **marginal ray** that describes the size of a bundle of light rays entering a lens. A full description and a discussion of its importance is given in Section 5.3.

To cause any ray bending in a lens, there must be changes in the refractive index across the lens surfaces. While the value of the refractive index could be entered, we would need to determine what glass is being used and then find the index of refraction at each design wavelength by looking it up or calculating it using a formula. Although it is possible to do this, in most lens design programs, a glass name is usually entered in the Material column, and the program calculates the refractive index for the wavelengths that have been chosen in the specification phase. For example, to assign a common optical glass, N-BK7, to the lens, its name, N-BK7, is entered in the G1 cell. An extensive discussion of glasses and wavelengths is given in Chapter 9 on chromatic aberration.

The last column, Clear Semi-Dia, displays the semi-diameters for each surface. (The term "semi-diameters" is used instead of "radius" to remove any confusion between the radius of a lens opening and the radius of curvature of a surface.) The semi-diameters are values that can be assigned by a user or computed by the program. For the present, all semi-diameters will be automatically calculated. User-entered apertures can be defined by changing the solve type (click on the small box next to the semi-diameter value to get to a drop-down menu) from automatic to fixed. A more detailed discussion of the handling of apertures and their effects on ray tracing is given in Chapter 5.

3.3 Entering a Single Lens

Start the entry of a single lens (often called a singlet by designers) by opening a new lens (File > New). We will call this lens "OSlens." Enter this name as its title in the System Explorer (System Explorer > Title/Notes > Title:). The "OS" label, standing for OpticStudio, is used to designate designs created, modified, and discussed throughout the text. As described in Section 3.1 System Data, you will give the lens a 20-mm-diameter entrance aperture and change the operating wavelength to the d-line. Insert a surface (see Section 3.2) after the stop surface.

Exercises

Because this chapter describes the entry of a lens into the design program, there were no exercises proposed until now. Instead, all of the exercises are given here with the answers following them. You may want to save the lenses in these exercises, but reserve the names of the files, OSlens and OSsinglet, for the design as they were saved in Sections 3.3 and 3.5, respectively. They will be used in future chapters of the text.

Exercise 3.1 Plano-convex lens
Restore the **OSlens** and replace its second surface with a flat surface. Determine the BFL and the paraxial image distance for this lens. If the object distance is changed to 500 mm, where is the paraxial image plane located? Did the BFL change?

Exercise 3.2 Meniscus lens
Replace the second surface of the **OSlens** with a concave surface of 200-mm radius of curvature. Determine the BFL and the paraxial image distance for this lens. If the object distance is changed to 1 m, where is the paraxial image plane located? What is the BFL?

Exercise 3.3 75-mm singlet
Restore the **OSlens** and use an angle solve on the R2 radius to change it to a 75-mm-EFL lens.

Exercise 3.4 Changing the singlet to a doublet
Restore the **OSsinglet** and make the following changes to the design:

(a) Insert a surface between S1 and S2 of the singlet.
(b) Give the new surface, S2, a radius of +200 mm and a thickness of 5 mm with SF1 glass.
(c) Determine the BFL and EFL of the new lens.

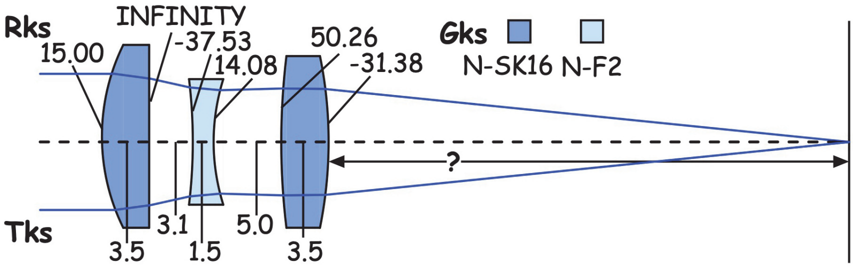
The following four exercises provide training in entering a system's prescription. The values for the radii, element thicknesses, airspaces, and materials are given in the figures. Assume that the object is at infinity, and use a thickness solve to find the image location. All lenses have EFLs at or near 50 mm at 587.56 nm (this text's default wavelength). For those lenses with multiple glasses, the glasses are given by the color-coded legend. When you have the correct configuration, save the file for future use, using the names given with each exercise.

In all of the previous exercises, the stop was located at S1. This will not be the case for these next four exercises. The stop will be elsewhere. After the necessary number of surfaces have been entered into the LDE, click on the surface where the stop should be. Then click on the drop-down arrow above

Surface Type to open the surface properties dialog box for that surface. Check the box in the upper right-hand corner of the Type tab, Make Surface Stop. The stop should now move from S1 to the correct surface.

Exercise 3.5 Triplet

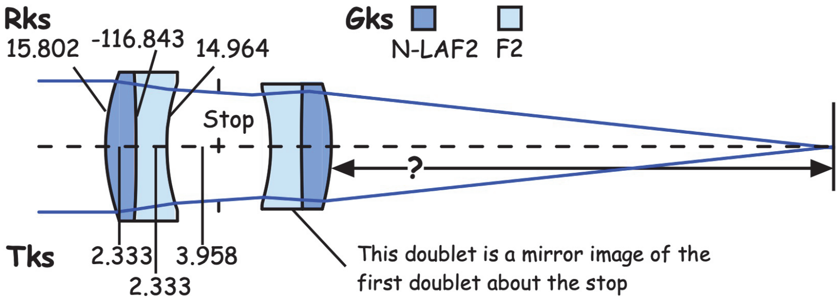
The EPD is 10 mm, and the stop is located at the first surface of the second element.



Use a thickness solve to find the image distance. What is the EFL? BFL? Save this lens as **OStriplet**.

Exercise 3.6 Rapid rectilinear

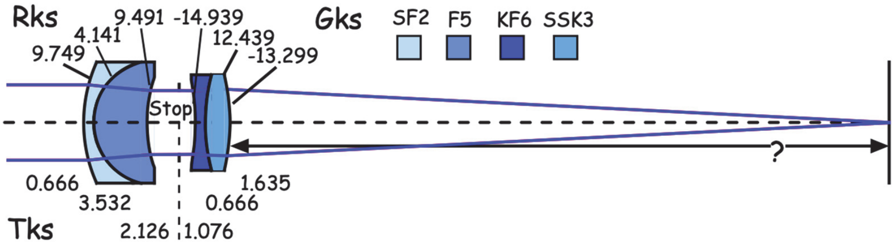
The EPD is 10 mm. The second doublet is the mirror image of the first.



Use a thickness solve to find the image distance. What is the EFL? BFL? Save this lens as **OSrapidrect**.

Exercise 3.7 Protar

The EPD is 5 mm. The stop is between the two doublets, 2.126 mm from the first.



Use a thickness solve to find the image distance. What is the EFL? BFL? Save this lens as **OSprotar**.

Chapter 4
To First Order. . .

In Chapter 3, we did a quick check to see if the initial entry of our OSlens was correct using the FIRST macro. The first-order listing for the OSlens is shown again in Table 4.1. The first three numbers may be puzzling because there are three separate focal lengths (effective focal length, back focal length, and front focal length) listed, and one of them is negative. Furthermore, none of these values exactly matches the thin lens focal length value of 68.294 mm calculated in Section 3.4. So, what is the correct value for the focal length?

Table 4.1 First-order properties of the OSlens using the FIRST macro.

```
Infinite Conjugates
   Effective Focal Length     68.9849
   Back Focal Length          68.0055
   Front Focal Length        -66.6344
   F/#                         3.4492
   Image Distance             68.0055
   Lens Length                 5.0000
   Paraxial Image
      Height                   0.0000
      Angle                    0.0000
   Entrance Pupil
      Diameter                20.0000
      Location                 0.0000
   Exit Pupil
      Diameter                20.7055
      Thickness               -3.4127
```

4.1 Principal Surfaces and Planes

To help explain why several focal lengths are computed and used for a single lens, we begin with an exaggerated lens example. Figure 4.1 shows a thick lens in air with a very short radius of curvature on its back surface focusing a fan of parallel axial rays. The solid blue lines show the standard trace of the rays through the lens. Paraxial rays, those close to the axis, focus at the **back focal point $\mathbf{F'}$**. A plane perpendicular to the axis at this point is defined as the **back focal plane (BFP)**.

But $N'F' = n \bullet \text{EFL}$, so $n' \bullet \text{EFL} = n \bullet \text{EFL} + \Delta\text{N}$. Solving for ΔN, we arrive at the equation for the nodal point shift, whose sign indicates the direction of the shift:

$$\Delta\text{N} = n' \bullet \text{EFL} - n \bullet \text{EFL} = (n' - n)\text{EFL}. \tag{4.2}$$

Some simple geometry will show that the ΔN values are equal, so that NN′ equals the hiatus PP′. Several important conclusions can be drawn from the figure and from Eq. (4.2):

1. There is no nodal point shift if the refractive index is the same in both spaces.
2. The shift ΔN is in the direction of the higher refractive index.
3. Because both nodal points are shifted by the same amount, the nodal points are separated by the hiatus H, as are the principal points.

From Eq. (1.7), we see that for a single surface $\text{EFL} = R/(n' - n)$. If we insert this into Eq. (4.2), we find that the displacement of the nodal point from the principal point ΔN equals R for the air–water interface problem. When we add the cardinal points to the air–water interface figure (Fig. 4.13), as shown in Fig. 4.18, you can see that the nodal points N and N′ are at the center of curvature of the surface, as expected.

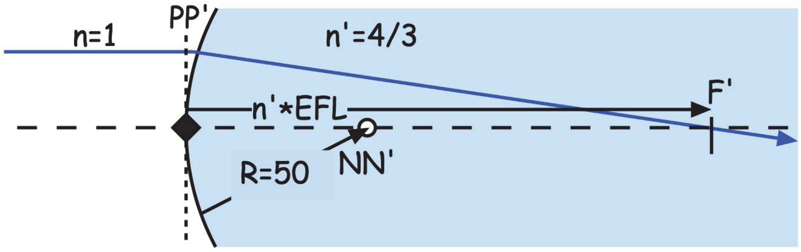


Figure 4.18 Air–water interface showing the principal, nodal, and focal points.

4.4.2 The human eye

Certainly, the most common immersed optical systems are our eyes. In Fig. 4.19, the optically important components are labeled on a model of the human eye. The outer surface of the eye (the cornea) is very thin (about 0.5 mm) and encapsulates the vitreous humor. The cornea provides the major contribution to the eye's optical power. Behind it is the crystalline lens whose power changes to focus light onto the retina as objects at different distances are observed. Behind the lens (and filling the rest of the eyeball) is the aqueous humor. Both humors are liquids with properties very close to water. Besides establishing the optical properties of your eye, the humors also maintain the rigidity of the eye.

Everyone's eyes are different, and they change as we age. There is no single optical prescription for the eye, but there are a number of models that represent a fairly realistic version of the eye. The Emsley model (Fig. 4.19) is one of the simplest models in that it ignores the thickness of the cornea. It also models liquids in the eye (humors) using water.

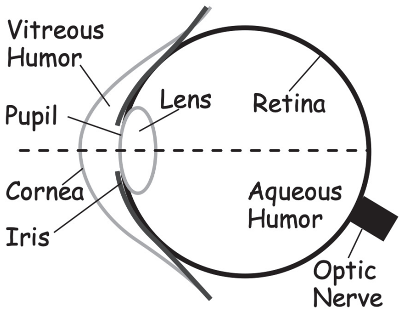


Figure 4.19 Cross-section of the human eye using the Emsley model.

Open up a new lens (File > New) and enter the prescription for the model eye from Figure 4.20. The stop is located at the iris at the front surface of the lens (S2). Change the EPD to 6 mm and the operating wavelength to the *d*-line. Because the material in the eye lens is not a catalog glass, OpticStudio permits the user to enter a model glass. For the present, the *V*-number is a property of glass that, along with its refractive index, identifies a specific glass. This will be covered in Chapter 9 on chromatic aberration. An extensive discussion of model glasses can be found in Section 12.6.1. In this simple model of the eye, the lens is assigned a refractive index of 1.415 with a *V*-number of 47. This is entered in the Material column of S2 by clicking on the box to the right of the cell and from the Solve Type menu by selecting Model, then entering 1.415 in the Index Nd space and 47.0 in the Abbe Vd space. Add a paraxial image thickness solve to the T3 thickness to locate the image surface at the paraxial image plane. The resulting LDE is shown in Fig. 4.21.

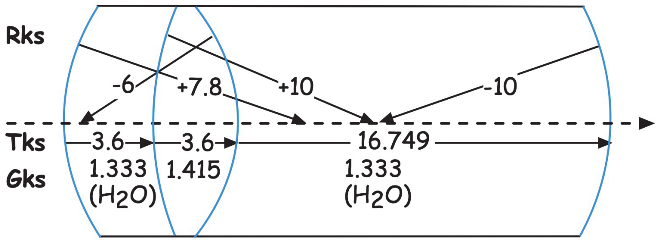


Figure 4.20 Simple model of the human eye for computer analysis.

	Surface Type	Comment	Radius	Thickness	Material	Clear Semi-Dia
0	OBJEC Standard ▾		Infinity	Infinity		0.000
1	Standard ▾	Cornea	7.800	3.600	WATER	3.150
2	STOP Standard ▾	Lens	10.000	3.600	1.42,47.0 M	2.657
3	Standard ▾	Aqu. Humor	-6.000	16.749 M	WATER	2.466
4	IMAGE Standard ▾		-10.000	-		0.241

Title (OSeye); EPD (6 mm); Field (0°); Wavelength (d-line)

Figure 4.21 LDE for a simple model of the human eye.

Chapter 5
Stops and Pupils and Windows, Oh My!

After a lens has been entered into OpticStudio® and the location of the image plane for an axial object point has been found using a paraxial image thickness solve, we need to determine how the lens performs at off-axis object points. We begin with a small number of points in the object plane, called **fields**, that are used to represent extended objects. Then we examine the passage of light through the lens by introducing the concepts of the aperture stop and field stop of the system, as well as its pupils and windows. These definitions may seem to be a needless complication ("C'mon, you've found the image, haven't you?"). But the purpose of a lens is to transfer radiation (light) from an object to an image. If the image is too dim or the object is not completely imaged, the design is a failure no matter how well resolved the image might be on axis. Students encountering the stops, pupils, and windows of a lens for the first time may find them bewildering. Thus, the title of this chapter. The objective for this chapter is to present stops, pupils, and windows as clearly as possible so that you feel comfortable using them in the construction and evaluation of optical systems.

5.1 Fields

Thus far, the object to be imaged has been a single point on the optical axis of the lens. We could add many points, covering the entire scene, so that the object would be a 2D picture. This type of object is available in OpticStudio, but the approach requires significant computing power and time. By taking advantage of the symmetry of the lens, we could determine the quality of the image by tracing rays from many points along a line in the object plane. But even that may be too complicated. To start, we simplify the evaluation by using three object points (see Fig. 5.1). One is the on-axis point; a second is at the edge of the picture; and the third is a point somewhere between the other two, shown in the figure as light dots on the arrow in the object plane.

Second Hiatus
Rays and Waves

Beginning in classical times, light patterns were investigated as occurring from light traveling in straight lines (e.g., objects casting shadows), and the operation of the simplest optics was described in terms of light rays (see Chapter 1). In the next few chapters, we will also examine the performance of lenses using the ray model, but there are situations where the evaluation of lens performance using rays fails to provide a realistic assessment of the design. This is because light consists of electromagnetic wavefronts moving through space, and its wave properties must be included in the evaluation of lens performance. Because light can be modeled as both a ray and a wave, this dual aspect of light must be addressed to correctly model an optical design. When a light wave encounters an aperture, it can bend around the corner of the aperture and enter the region of the geometrical shadow. While a ray-based model would argue that this is impossible, this wave-based phenomenon is known as diffraction and depends on the wavelength of the light and the size of the aperture in the optical design.

H2.1 Rayleigh Criterion

The first demonstration of the diffraction of light is usually that of light diffracted by a narrow slit. Instead of a narrow band of light beyond the slit, the light pattern is a bright central band flanked by dark and bright bands that diminish away from the central peak. The distribution of light [Fig. H2.1(a)] is described by the function $\sin^2\alpha / \alpha^2$. The variable α depends on the wavelength of the light and the width of the slit. For a lens with a circular aperture, the diffraction profile has circular symmetry and can be described with a circular function called a first-order Bessel function $J_1(x)$. Just as in the case of the narrow slit, the diffraction pattern for the circular aperture is the square of the function over its argument, $4J_1(x)^2 / x^2$. This function is known as an Airy function, and the resulting pattern is called an Airy diffraction pattern [Fig. H2.1(b)]. As in the case of the narrow slit, the dimensions of the pattern are determined by the diameter of the diffracting aperture D and the wavelength of the light λ. The angular separation θ between the peak of the pattern and the first dark ring is given by

$$\theta = 1.22\frac{\lambda}{D}. \tag{H2.1}$$

Chapter 6
Spherical Aberration

Once upon a time, the evaluation of an optical design required a great deal of ingenuity. With no high-speed computers to aid them, designers were forced to find shortcuts and clever approximations to assess the performance of a lens. They developed analytical tools to provide insight from limited input data in the shortest computational time. A paraxial ray trace, accurate for a small area about the optical axis, was used as a baseline measure of a perfect image. If the lens provided a perfect image, the areas on the object that were farther from the axis would also be imaged exactly with the correct magnification. To the extent that a lens fails to do this, the differences between the rays of a perfectly imaged object and those directed by the actual lens are a measure of the optical errors, or **aberrations**, in the lens.

6.1 Propagating Real Rays

Figure 6.1 illustrates the propagation of a ray through an optical system. It compares two rays. One ray is propagated by a lens with no aberrations (a "perfect" lens); the other is an aberrated skew ray propagated by a real lens. To keep the situation as general as possible, all quantities in the figure are normalized to their maximum values. Thus, at a point in the object plane, the fraction of the field between the full field and the axis is designated as *h*, a proper fraction between one and zero ($1 \geq h \geq 0$).

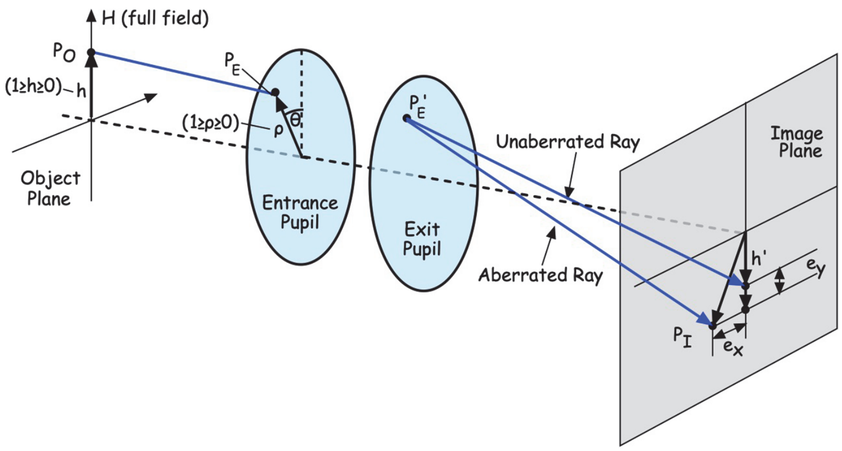


Figure 6.1 Propagation of an aberrated skew ray through an optical system compared to an unaberrated ray.

6.3 On-Axis Ray Errors for a Singlet Lens

For an on-axis object point ($h = 0$), the only third-order aberration for a rotationally symmetric lens will be spherical aberration. This is most easily demonstrated by launching a fan of rays from the axial object point that is spread across the breadth of the entrance pupil. For example, if we locate the object at infinity and the lens exhibited no spherical aberration, all of the rays in the fan would intersect on the axis in the focal plane of the lens. To see what happens in a real lens, we restore the **OSsinglet** (File > Open). If you can't retrieve it, its LDE is shown in Fig. 6.2.

	Surface Type		Comment	Radius	Thickness		Material	Clear Semi-Dia
0	OBJECT	Standard ▾		Infinity	Infinity			0.000
1	STOP	Standard ▾		120.000	5.000		N-BK7	10.000
2		Standard ▾		-32.464	49.290	M		9.914
3	IMAGE	Standard ▾		Infinity	-			1.350

Title (OSsinglet); EPD (20 mm); Field (0°); Wavelength (d-line)

Figure 6.2 LDE of the OSsinglet.

We can see what happens if we trace a fan of 15 on-axis rays through the lens by selecting Analyze > Cross-Section and changing the Number of Rays in the Layout > Settings to 15. As shown in Fig. 6.3, the rays do not all focus in the paraxial image plane. Instead, they cross the axis at different distances from the plane. Because of the large number of rays in a small region, the focal region is magnified in the inset box. Parallel rays close to the axis cross at the paraxial focal point (PF). The pair of marginal rays that enter at the edges of the entrance pupil cross at a point on the axis (MargF) much closer to the lens. There is a point on the axis where the focusing ray bundle is smallest (MinF). This is the location where all of the rays in the fan are contained in the smallest area.

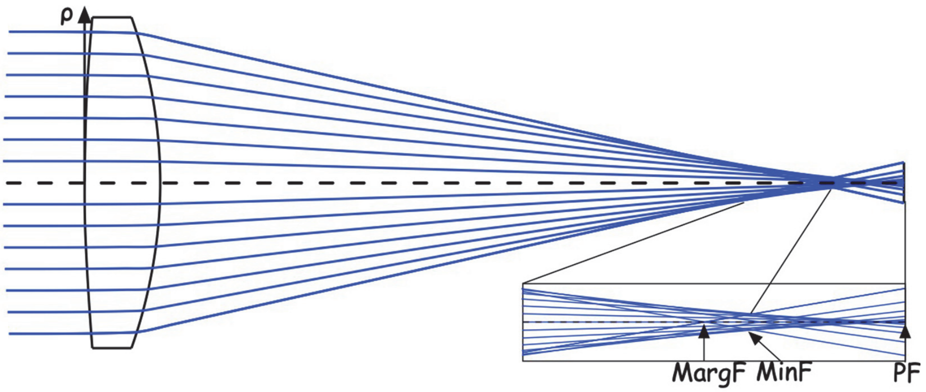


Figure 6.3 Spherical aberration of a singlet with a magnified focal region. MargF is the marginal focal plane; MinF is the plane with the smallest spot containing all the rays; and PF is the paraxial focal plane.

Chapter 7
Coma and Astigmatism

In the previous chapter, we demonstrated the properties of spherical aberration and explored its behavior along the optical axis using transverse ray curves and spot diagrams. If spherical aberration were the only aberration in the lens, the performance would be the same across the entire field. However, at field points away from the axis, additional errors add to spherical aberration. In this chapter, we examine the next two third-order transverse errors: coma ($\rho^2 h$) and astigmatism ($\rho_y h^2$ and $\rho_x h^2$).

7.1 Coma

Restore the $f/2.5$ **OSsinglet** with an object at infinity. In the FDE (Setup > Field Data Editor), enter our standard set of fields (0°, 7° and 10°) as described in Section 5.2.5. For more complex lenses, these angles are fairly modest, but for a singlet, they are large enough to demonstrate two off-axis errors, coma and astigmatism. As is indicated in Table 6.1, coma, or comatic error, is quadratic in ray height in the pupil ρ and linear in ray height in the field h. The maximum coma occurs for rays from the largest field value $h = 1$ and at the edge of the pupil $\rho = 1$.

To examine the off-axis errors in the singlet, create a lens cross-section (Analyze > Cross-Section) with an off-axis fan of 15 rays from the 7° field (F2) as shown in Fig. 7.1. This is generated by changing the Number of Rays to 15 and the Field to 2 in the Layout Settings window. For reference, we have thickened the 7° chief ray. If you want to see the area near the focal region in more detail, you can use the magnifying glass icon on the Layout toolbar to draw a box around the area you want to zoom in on. To reset the zoom of the layout, click the black icon with the counter-clockwise circular arrow at the end of the Layout toolbar.

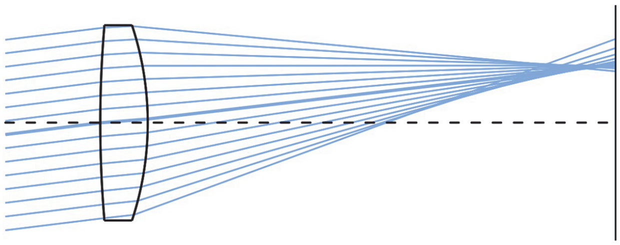


Figure 7.1 Meridional ray trace of the OSsinglet for a 7° object field.

Chapter 8
Aberrations of the Image Surface

In most cases, when you design an optical system such as camera lens, you require that the lens form an image of a scene on a flat surface. For example, Fig. 8.1 shows the imaging of an object at infinity onto a flat paraxial image plane by a perfect lens, i.e., one with no aberrations. However, the image surface for a real lens is not a plane but a curved surface whose *z*-axis departure from a flat paraxial image plane represents an image surface error for the lens called Petzval curvature. If the image plane is a curved surface (consider the shape of your retina), Petzval curvature may not cause any problem. However, because most modern sensors are flat, not curved, Petzval curvature must be managed when designing a lens.

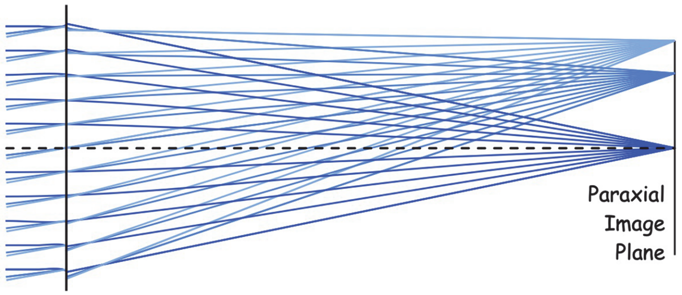


Figure 8.1 Imaging through a perfect lens.

This chapter explores Petzval curvature and a second image-surface aberration, distortion, in more detail. Distortion is an image-surface error that does not blur the image but causes errors in the chief ray locations in the *x*-*y* plane, resulting in a change in magnification with image height. This results in a stretched (or compressed) image of the object.

8.1 Field Curves

As was shown in the previous chapter, an aplanatic lens exhibits little spherical aberration or coma, and the focal images reside on curved surfaces. Restore the **OSaplanat**, shown in Fig. 7.6, either from the saved lens or from its prescription in Fig. 7.7. Trace a fan of 9 rays for each field in the Layout window (Analyze > Cross-Section). Figure 8.2 shows that the image points for

Chapter 9
Chromatic Aberration

When the early astronomers looked at the sky through their telescopes, the images of the stars were surrounded by colored halos caused by an optical error in the telescope, chromatic aberration. The error was so large that it severely limited their telescopes' resolution. During the Great Plague of 1666, Isaac Newton retreated to his family home at Woolsthorpe, where he began a series of experiments on the colors of light. To investigate the problem, Newton ground and polished a triangular prism. He directed a beam of light through the prism onto a wall of his room, where the color spectrum was displayed. And when a part of this spectrum was passed through a second prism, no additional colors were created. From this, he concluded that glass in the prism did not create the colors, but that white light consists of a spectrum of colors, and the prism spread them out. Faced with these results, Newton decided that it was not possible to construct a telescope from lenses without color error. He concluded that the only way he could make a telescope without chromatic aberration was to use a mirror to focus starlight because all wavelengths obey the same law of reflection. The result was the Newtonian reflecting telescope. It wasn't until more than 90 years later that John Dollond devised a method for correcting chromatic aberration in glass lenses.

In this chapter, we will discuss the values and notation used to describe the colors of light. Then we will use the **OSsingletRev** to demonstrate chromatic aberration and determine the nature and size of the color error in an image. Finally, we will describe how this color error can be corrected with a doublet.

9.1 Refraction and Dispersion

Optical materials, such as glass, crystals, and plastics, possess many different material properties (e.g., density and hardness). For optical design, the two principal material properties are the refractive index and dispersion. The first of these determines the amount of ray bending across a surface (Fig. 1.4) according to Snell's law:

$$n' \sin i' = n \sin i, \tag{1.2}$$

where i is the angle of incidence, and n is the refractive index on the one side of the interface; i' and n' are the same quantities on the other side. If the initial medium is air, then $n = 1$. But what refractive index should be used for the

Chapter 10
Reducing Aberrations

Once you learn how to model an optical system and analyze its optical errors, you can determine if the performance of the existing design is good enough or if it needs to be improved. In some cases, it's acceptable to change the specifications to improve the performance of a design (e.g., stopping down the lens). In other cases, the design can be modified to improve performance, a process called **optimization**. For example, in Section 6.4, we used Optimize > Quick Focus to find the best image plane location. Later in Chapter 6, the OSsinglet was modified by "bending" the lens (by hand) to reduce its spherical aberration. If needed, lenses can be added or a completely different design form can be sought. Up to this point in the text, most of the optimization of lens performance has been done manually. But, optimization can be done automatically in OpticStudio® using the various features found on the Optimize tab.

In this chapter, we will focus on OpticStudio's local optimization (Optimize > Optimize!) to reduce the third-order aberrations that we identified in Chapters 6–8. This is not a comprehensive approach to optimizing lens performance. In practice, just reducing third-order aberrations is not advised because real lenses can have large apertures and/or large fields that produce higher-order aberrations. Targeting the third-order aberrations to zero prevents them from balancing with higher-order aberrations (typically resulting in worse performance). However, this approach provides an introduction to optimization and therefore is presented before we tackle more advanced optimization (including global optimization) of lens systems in Chapter 12.

10.1 The Merit Function

Optimization requires a starting design, a set of variable parameters (e.g., curvatures, thicknesses, and airspaces), and a **merit function** (a means of accounting for the change in performance during optimization). Optimization is carried out in a series of steps or cycles. During each cycle, the variables in the design are allowed to change in small increments. At the end of each cycle, the merit function value of the new design is computed and compared to that of the previous cycle. For local optimization, OpticStudio uses an actively damped least-squares algorithm to find the minimum of the specified merit function that is nearest to the starting point design. Then the designer must determine if this is an acceptable design ("good enough!") or whether more substantial changes to the starting point must be made and another round of optimization started.

Chapter 11
Analyzing the Performance of a Lens

In the previous chapters, we've used a series of lenses to illustrate the properties and limitations of optical systems. What has not been addressed, until now, is what a lens will be used for and how good it needs to be. That is, if we want to design a lens to accomplish something in the real world, how do we go about it? While lenses can be used to focus a laser or to direct the output of a light source to illuminate a surface, the lens designs discussed in this text are targeted at imaging a real-world scene onto an appropriate sensor with the necessary resolution.

But what does it mean to "resolve" some detail of an image? In crime dramas on television or in the movies, there have been scenes in which a detective (or technical wizard) examines an image on a computer. The detective magnifies an area, clearly revealing a license plate number or a fuzzy portrait of someone near the scene of the crime. Sometimes screenwriters invoke the use of highly sophisticated software to "improve" the resolution from a small area of a single frame of a camera video. But how realistic is this? In this chapter, we will discuss different ways to measure the resolution of an optical system. We will show that it is a function of both the sensor and the lens, and that it can be limited by aberrations, diffraction, and/or the sensor itself.

11.1 Sensors

In an optical system, the image of an object is typically captured by a sensor. Ideally, there is a one-to-one mapping between each point in the object and each point in the image. The sensor stores the information as the difference (contrast) in light distribution between adjacent areas on the sensor. In your eye, for example, images are collected by your retina using photosensitive structures called rods and cones ($\sim$2 μm in size). Depending on the lens that the sensor is coupled to, the size of these light-gathering structures can impact (and ultimately limit) the resolution of the optical system.

11.5 Modulation Transfer Function

Another approach to evaluate the performance of a lens is to compute the response of the lens to a broad range of spatial frequencies. Like the audio response by a stereo system when reproducing music from a variety of musical sources, the response of an optical system is measured by the **optical transfer function (OTF)**. The OTF is the Fourier transform of the point spread function (the image of a point source). The **modulation transfer function (MTF)** is then the modulus of the complex OTF. Similar to measurements of a USAF 1951 chart, the MTF measures the image contrast as a function of spatial frequency. However, the analysis does not require a bar chart with element-by-element analysis. Instead, the wavefront emerging from the exit pupil of a lens from a single field point can be constructed in optical design software from the optical paths of traced rays and used to compute an MTF via a Fourier transform.

Figure 11.13 shows a basic MTF plot for one field point. The modulation (or contrast) is plotted for a series of increasing spatial frequencies for both sagittal (vertical) lines and tangential (horizontal) lines. For an optical system with no aberrations and no diffraction, the MTF would be unity at all frequencies. The object shown in the top right of the figure represents a sinusoidal wave of 30 cycles/mm. Note: A cycle is the equivalent of a line pair on the USAF target. The MTF curves on the left and the profiles on the right represent the responses of the lens. If the lens were a perfect lens having no aberrations, its MTF would only be due to diffraction, given by the solid black line. In a real lens, the contrast is degraded by both diffraction and aberrations. By default, the MTF is plotted to the optical cut-off frequency, which is the spatial frequency beyond which an optical system cannot transmit information and for incoherent light equals $2(\mathrm{NA})/\lambda$ (or $1/\lambda f/\#$). Here, the MTF is plotted out to 90 cycles/mm.

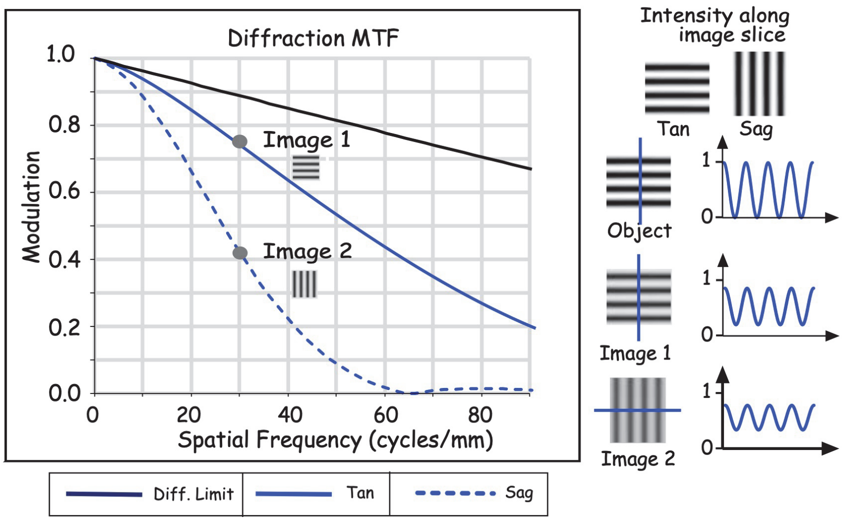


Figure 11.13 MTF chart for one field point showing both tangential and sagittal bars and the diffraction limit.

Restore the OStripletMod (Fig. 11.3) and change the image distance to 38.421 mm (for best focus). Like the PSF, there are a number of ways to produce an MTF plot in OpticStudio. For example, a fast Fourier transform (FFT) MTF plot can be generated by clicking on the MTF icon in the Analysis tab. Within the drop-down menu that appears, choose FFT MTF (Analyze > MTF > FFT MTF). At first, this feature creates an MTF plot out to the cutoff frequency of 410 cycles/mm, a frequency range that far exceeds the ability of this lens to reproduce such fine features. To display the contrast of this lens at a more appropriate frequency range, open Settings (Fig. 11.14) and change the Maximum Frequency to 100. Increase the Sampling to 128×128 and add a diffraction-limited (solid black) line to the plot by checking the Show Diffraction Limit box.

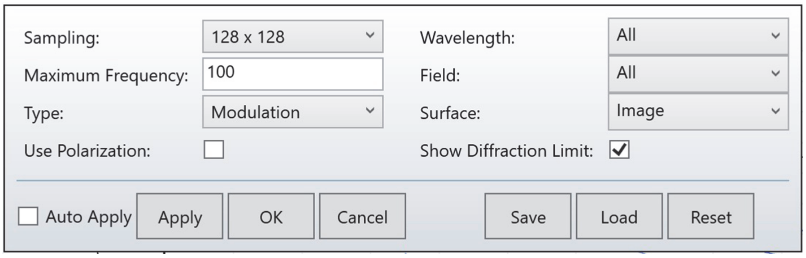


Figure 11.14 Settings for FFT MTF.

Click OK. The results are shown in Fig. 11.15. The program generates a plot of the MTF versus spatial frequency for the 0°, 7°, and 10° fields. For the axial field, the sagittal and tangential curves are the same and overlap so only one curve is shown. For the other two fields, the tangential responses are plotted as solid lines, while the sagittal responses are dotted lines. The MTF plot also shows the curve for a diffraction-limited lens (Diff. Limit) with the same *f*/# and operating wavelengths. It's quite clear that OStripletMod is not diffraction limited as the contrast for all fields is far below the diffraction limit. This agrees with our PSF analysis in Section 11.3.

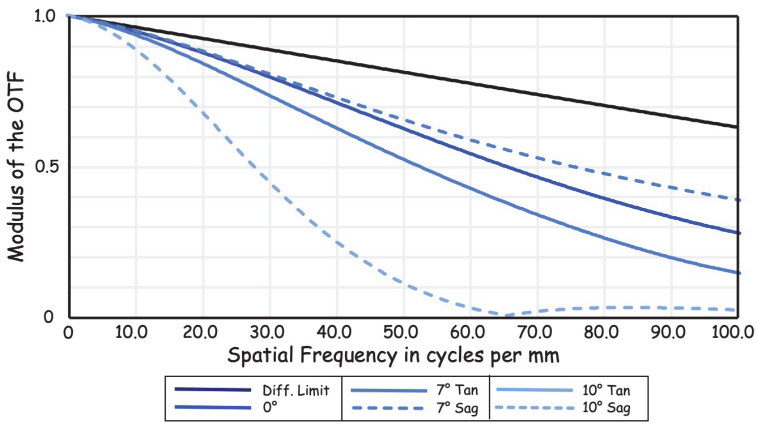


Figure 11.15 MTF plot of the OStripletMod lens.

Chapter 12
Designing a Lens

Thus far, we have shown how to enter a lens design into OpticStudio®, describe its faults, and then improve its performance. Our objective in this chapter is to design a lens to meet the requirements for a specific application. Starting with a simple lens system, it will be evaluated to see if it satisfies our requirements. If not, the design will be modified and re-evaluated through a series of steps or iterations until we arrive at a design that can perform to the specified conditions.

12.1 Defining the Problem

Now let's use the concepts that have been demonstrated in previous chapters to design a lens for a simple security camera. Modern security cameras can be quite complex, operating in both the visible and near-infrared wavelength bands with very large fields of view (FoVs) and mounted on a motorized stage to scan and zoom the scene. For now, we will focus on a much simpler security camera, whose purpose is to record objects and activity within a moderately sized (fixed) field and provide surveillance in the visible band only. In most instances, such cameras are mounted overhead or on ceilings (Fig. 12.1). Therefore, the objects of interest will be located far enough away from the lens that we may assume that the object is at infinity.

Figure 12.1 Ceiling-mounted security camera.

This type of optical system consists of an imaging lens and a CMOS sensor. But which lens and which sensor should we use? Do we need a custom design for the lens? A custom sensor? In the real world, the cost of materials, manufacturing, or assembly for an optical system must be considered before arriving at a final design solution. Although custom lenses are fairly common, the choice of a sensor is usually limited to one of the existing standard formats (see

Third Hiatus
Building a Lens

To properly tolerance a lens system (Chapter 13), you need to have a good understanding of how the lens will be built. This includes knowledge of how the lens elements will be fabricated, aligned into a mechanical housing, and tested. There are many ways to build a lens, ranging from traditional processes (completed primarily by hand) to modern, machine-driven [e.g., lens molding or computer numerical control (CNC) grinding and polishing] fabrication methods. While there are entire texts on this subject matter, we will briefly cover the more classical "by hand" methods here and leave the newer fabrication and testing techniques for you to explore on your own.

H3.1 Fabricating a Lens Element

Traditionally, a lens was made (often by hand) by grinding and polishing the lens on a rotating spindle. During the fabrication process, the lens parameters (e.g., radius and thickness) are tested to determine if they meet the design values. Ideally, each copy will be a replica of the optimized design.

Figure H3.1 illustrates the first steps in the fabrication of a singlet (OSsecureCam2). A disk of glass, appropriately called a **blank**, is attached to a mounting surface using a melted wax that hardens. A rotating metal tool with a convex surface is brought into contact with the blank [Fig. H3.1(a)]. This tool has been cut and polished to a high degree of accuracy to obtain a radius of curvature equal (but opposite in sign) to the radius of curvature of the second surface R_2 of the lens.

A slurry of abrasive metal oxides is applied to the surface, and the rotation of the tool against the blank grinds a rough surface. This process is called **generation**. Slurries with increasingly smaller grit sizes are applied to bring the surface from a "rough grind" to a "fine grind." The smaller the particle size in the slurry the less material is removed and the smoother the surface will be. Then the lens is demounted, flipped over, and reattached to the mount. A second set of grinding tools is used to generate R_1 [Fig. H3.1(b)]. In addition to generating R_1, this grinding stage must also obtain the correct thickness of the lens.

Chapter 13
Tolerancing

So, an optimized security camera has been designed. But can it be built? For the performance of a manufactured lens system to be as good as the nominal design performance, it must be fabricated and assembled as close as possible to the nominal optical design prescription. The question is how precisely must the various lens parameters be made? If you copy (and paste) any cell value in the LDE, you will see that each parameter is expressed to 15 places beyond the decimal point. Although this level of precision is not needed to specify the lens parameter, the lens parameter must be made (see the Third Hiatus) to an accuracy or **tolerance** that will be good enough to meet the performance specifications. This provokes a third question: How do you determine what tolerances are "good enough?" For example, how carefully must we make the radius of curvature of a lens element to meet a specified RMS spot diameter across all fields?

The answers to these questions can be provided through a general process called **tolerancing**. While it is relatively easy to model one specific lens error, it is much more difficult to think about how multiple lens errors, all occurring at the same time (with random values within their tolerance ranges), will affect the final lens performance. This is where we turn to software and statistical tolerancing algorithms to predict the probability of producing a set of lenses with the desired performance (at a desired cost). In this chapter we will use our two security camera design examples from Chapter 12 to demonstrate how to use OpticStudio® to find appropriate tolerance values for an optical system, predict as-built performance, and then communicate our tolerances to a manufacturer using an optical drawing.

13.1 Statistical Tolerancing

The first step in any tolerancing process is to define a set of starting tolerance values for the various manufacturing errors. These are the ranges, usually symmetrical about the design value, by which the lens parameters can vary during manufacturing. For example, a tolerance value of ± 25 μm on the thickness of a 0.5-mm-thick lens would require a lens to be fabricated with a thickness between 0.475 and 0.525 mm.

Once an initial set of tolerance values has been selected, we need to understand how sensitive each of the lens parameters is to change by evaluating how much that change would degrade the lens performance. Both positive and negative changes in lens parameters are evaluated (one tolerance at a time). After

Appendix
Macros: FIRST, THIRD

First, let's find out if the FIRST and THIRD macros described in Sections 3.4 and 6.6 are already installed in your copy of OpticStudio. Click on the Programming tab and then select the Macros List drop-down menu at the far left of the Programming ribbon. This will display a list of all preinstalled macros. Check to see if FIRST.ZPL and THIRD.ZPL are present, and if they are, click on them to use them.

If you are running a version of OpticStudio that does not come with the FIRST and THIRD macros preinstalled, you can create your own. First, open the Programming tab and click on the New Macro icon (sunburst) to open a New ZPL Macro window. Then copy the entire text of one of the macros below and paste it into the text window. Note that the text in OpticStudio is now displayed in a color-coded format. Click the Save As icon (second from left). This opens the Zemax > MACROS folder. Type the file name (FIRST or THIRD) and Save. You can then check that the file is now included in the Macros List in the Programming tab.

FIRST.zpl LISTING

```
! FIRST.zpl
! v0.5 (2022-03-22)
! Written by Don O'Shea 12-24-18
! Differences from prescription data output include:
!    F/# is calculated by EFL/EPD (standard textbook definition)
!    Exit pupil is measured from last surface vertex (not image
plane)

GETSYSTEMDATA 1

! get other data
bfl = OPEV(OCOD("CARD"), 1, NSUR(), PWAV(), 0, 3, 0) +
THIC(NSUR()-1)
ffl = OPEV(OCOD("CARD"), 1, NSUR(), PWAV(), 0, 2, 0)
m = OPEV(OCOD("PMAG"), 0, PWAV(), 0, 0, 0, 0)
LL = OPEV(OCOD("TTHI"), 1, NSUR() - 2, 0, 0, 0, 0)
TL = OPEV(OCOD("TTHI"), 0, NSUR(), 0, 0, 0, 0)
```

Donald C. O'Shea is Professor Emeritus for the School of Physics at the Georgia Institute of Technology. He is a Fellow of SPIE and of the Optical Society (OSA). He served as editor of SPIE's flagship journal, *Optical Engineering*, from 1998–1999 and 2001–2009. During 2000, he served as President of SPIE.

O'Shea received a B.S. in Physics from the University of Akron (1960), a M.S. in Physics from Ohio State (1963), and a Ph.D. in Physics from Johns Hopkins (1968). He did postdoctoral work on laser spectroscopy at the Gordon McKay Laboratory at Harvard from 1968–1970. In 1970, he joined the faculty at Georgia Tech, where he created an optics curriculum and published over 50 papers on optics and optics education. He was a Visiting Scholar at the Optical Sciences Center of the University of Arizona and at the University of Oulu, Finland.

He co-authored an undergraduate textbook on lasers, *An Introduction to Lasers and Their Applications*, with W. R. Callen and W. T. Rhodes (Addison-Wesley, 1977). He published a textbook on optical design, *Elements of Modern Optical Design* (Wiley, 1985), and a SPIE Tutorial Text on diffractive optics, *Diffractive Optics: Design, Fabrication, and Test*, with T. J. Suleski, A. D. Kathman, and D. W. Prather (SPIE Press, 2004). This current text is a companion to *Designing Optics Using CODE V®* (SPIE Press, 2018). He created the Optics Discovery Kit for OSA for use in precollege education. In 1996, he was awarded the Esther Hoffman Beller Award by OSA for "excellence in the field of optics education."

Julie L. Bentley is a Professor at the Institute of Optics, University of Rochester, and has taught courses in geometrical optics, optical design, and product design for over 25 years. She is a Fellow of SPIE and OSA and a former president of the Rochester Optical Society. She has served as an SPIE board member, an associate editor for *Optics Express*, and the chair for the International Optical Design Conference (IODC) and SPIE's optical fabrication conference (Optifab). In 2023, she was elected into the presidential chain for SPIE and will serve as its president in 2026.

Bentley received a B.S. in Optics (1990), a M.S. in Optics (1992), and a Ph.D. in Optics (1995) from the University of Rochester. She holds several U.S. patents and co-authored her first book, *Field Guide to Lens Design*, with S. Craig Olson (SPIE Press, 2012). Her second book, *Designing Optics Using CODE V®* (SPIE Press, 2018)), is a companion to this text, and was co-authored with Don O'Shea. In 2014, she received the University of Rochester's Goergen Excellence in Undergraduate Teaching award. In 2022, she received both the SPIE Director's Award and Optica's Esther Hoffman Beller Medal for Educational Excellence.

Her expertise is in the area of optical design and tolerancing of precision optical instruments. At the university, her research group is focused on three different aspects of optics: optical design with freeform GRIN (gradient index) materials, the design of next-generation adaptive optics (AO) ophthalmoscopes, and improved zoom lens design and optimization. Outside of teaching and research, she runs a successful optical design consulting business, Bentley Optical Design, where she designs lens systems for a wide variety of applications ranging from medical devices for cancer detection to visible and infrared military optics.