What Is *LensLab*?

Introduction

Welcome to *LensLab*.

LensLab brings optics to *Mathematica*. Both simple and complex optical systems can be defined and analyzed with the *LensLab* package. In addition to *LensLab*'s built-in library of lenses, mirrors, prisms, and gratings, new components can be created and existing ones modified with "generic building blocks", *LensLab*'s component-structuring language.

LensLab sets a new landmark for user-designed component and diagnostic capabilities. Written in the *Mathematica* programming language, the open-code architecture of *LensLab* enables users to make additions and programming changes on any level that is comfortable for them. Because *LensLab* uses *Mathematica*, it works with most computer platforms.

A Design Tool

Scientists and engineers who are developing equipment and experimental setups involving optics will find *LensLab* a useful tool for determining component specifications and system layouts. *LensLab* enables evaluation and optimization of alternative system designs before capital expenditures on equipment are made. The high-quality graphics generated by *LensLab* can be used directly in blueprints, grant proposals, and research reports.

An Educational Tool

LensLab is also useful for educational purposes, providing a foundation for *Mathematica*-based courseware in optics. *LensLab* enables computer-based laboratory experiments, in some cases replacing physical experiments and circumventing exposure to hazardous laser emissions and use of fragile or expensive laboratory equipment. By using *LensLab*, students can conduct experiments in optics with greater flexibility, precision, and in less time than is required using physical hardware. In other cases, *LensLab* can be used to supplement the laboratory experience to help students gain greater insights into physical experiments.

LensLab's Components

LensLab is a language extension of *Mathematica* for ray tracing and rendering of optical systems. By using *LensLab,* you can define lenses, mirrors, prisms, cavities, ring cavities, fresnel mirrors and lenses, lens doublets and triplets, pin holes, gratings, pipes, liquid-air interfaces, slits, optical fibers, lasers, amplifiers, beam splitters, screens, baffles, and paraxial components. Some paraxial components include thin lenses, thick lenses, ABCD matrix components, and paraxial-modeled graded index components. Regardless of their form, all components can have rectangular, circular, elliptical, or arbitrary polygonal edges.

In addition to using *LensLab*'s standard library of on-axis and off-axis spherical, cylindrical, and parabolic curved components, you can easily create new components having curved surfaces of your own design.

LensLab makes it easy for you to add new types of refractive materials to *LensLab*'s existing catalog of commonly used glasses, crystals, liquids, and gases. With *LensLab*'s built-in functions for creating new refractive index function models, you can use wavelength-dependent index measurement tables to generate new refractive material definitions. Once defined, these new materials can be used immediately in components.

LensLab's Component-Structuring Language

You can create hybrid optical components using "generic building blocks", *LensLab*'s component-structuring language. In this way the standard lenses and mirrors can have holes inserted in them or gratings placed on top of them. You can easily define new components from scratch with generic building blocks, just as *LensLab* does for its built-in component definitions. Because of *LensLab*'s open code, you can check how *LensLab* defines its stock components as a first step to designing your own custom components.

Features in *LensLab*

LensLab supports:

User-directed and nonsequential ray tracing

Off-axis surface curvatures

Arbitrary user-defined surfaces

High-level component-structuring language

User-defined refractive materials

User-defined intensity amplification models

Components with off-axis rectangular, elliptical, and polygonal holes

Full three-dimensional placement and evaluation of optics

Ray tracing for both imaging and nonimaging applications

Complete quantitative information about all optical surfaces

Two-dimensional and three-dimensional rendering of the same system

Wide range of component-rendering options including solid or wire-framed surfaces

Magnified presentations of selected system subgroupings

"Spot" diagrams of any optical surface

Optical systems can be defined in modular segments that can be assigned distinct variable names, independently moved in three-dimensional space, and chained together for modeling a total system.

LensLab has built-in functions for measuring focal lengths of real and virtual imaging systems. By adding your own *Mathematica* routines, *LensLab* can be used for creating phase plots and determining modulation transfer functions of optical systems.

It is not possible for *LensLab* to include built-in solutions for every project. Instead, coupled with the existing capabilities of *Mathematica*, *LensLab* is more like a well-stocked workshop ready for any project. If a component or analysis function doesn't exist in the *LensLab* package, you have the necessary tools to build it!

About This Manual

The Scope of This Manual

This manual provides a comprehensive view of *LensLab*'s functions and capabilities. However, it is not the purpose of this manual to teach about the general *Mathematica* system. There are many helpful books written for that purpose.

In most uses of *LensLab*, you will need to know a small part of *LensLab*'s functions. This manual is organized to make it easy for you to learn the specific information you need for a particular calculation. In many cases, for example, you may be able to set up your calculation simply by adapting some appropriate examples from this manual. You should understand, however, that the examples in this manual are chosen primarily for their simplicity rather than to correspond to realistic optical modeling situations.

LensLab is a system built on a fairly small set of very powerful principles. This manual describes these principles, but by no means spells out all of their implications. In particular, while the manual describes the elements that go into modeling optical systems, it does not give detailed examples of optical system design.

The Parts of This Manual

This manual contains ten chapters and an appendix. The four initial chapters provide essential information about using the *LensLab* system. Chapter 1 introduces you to *LensLab*'s basic features. By learning the functions presented in the first chapter, you will have a foothold for using *LensLab*. The second chapter shows you how *LensLab* does ray tracing. Understanding the ray-tracing process is essential for effective use of *LensLab*. In addition to explaining *LensLab*'s ray-tracing process, Chapter 2 introduces you to the built-in functions for generating rays. Next, Chapter 3 gives you a tour of *LensLab*'s built-in component functions. Then, Chapter 4 shows you how to use the different built-in **`Move`** functions to position components and rays in both two-dimensional and three-dimensional space.

Chapters 5 - 7 contain advanced information about the *LensLab* system. Chapter 5 introduces features of *LensLab* important to the advanced user. In Chapter 6, you learn how to perform some advanced experiments using *LensLab*. Then, Chapter 7 gives you a single example that demonstrates the basic design and layout techniques employed by the author for modeling more complex optical systems in *LensLab*.

The three final chapters in the manual provide reference information about particular elements of the system. Chapter 8 serves as a reference about the various parameters of the **`Ray`** object. Next, Chapter 9 serves as a reference about the **`Component`** object. Finally, the Appendix shows you a summary of *LensLab*'s built-in object and function names, followed by a complete reference guide of definitions for *LensLab*'s built-in objects and functions.

How to Read This Manual

If at all possible, you should read this manual in conjunction with using the actual *LensLab* package. When you see examples in this manual, you should try them out on your computer. For your convenience, this manual is provided in both printed and on-line notebook formats.

The first eight chapters in this manual are intended to be pedagogical, and can meaningfully be read in a sequential fashion. The last two chapters and appendix, however, are intended solely for reference purposes. Once you are familiar with *LensLab*, you will probably find the list of functions in the Appendix the best place to look up the details you need.

Copyright Statement

Optica Software makes no representations, express or implied, with respect to this Product, including without limitations, any implied warranties of merchantability or fitness for a particular purpose, all of which are expressly disclaimed. Users should be aware that included in the terms and conditions under which Optica Software is willing to license the Product is a provision that the author, Optica Software and distribution licensees, distributors and dealers shall in no event be liable for any indirect, incidental or consequential damages, and that liability for direct damages shall be limited to the amount of the purchase price paid for the Product.

In addition to the foregoing, users should recognize that all complex software systems and their documentation contain errors and omissions. Optica Software shall not be responsible under any circumstances for providing information on or corrections to errors and omissions discovered at any time in this document or the package software it describes, whether or not they are aware of the errors or omissions. Optica Software does not recommend the use of the software described in this document for applications in which errors or omissions could threaten life, injury or significant loss.