

ADDRESSING THE OPTICAL DESIGN AND ANALYSIS NEEDS OF CORPORATE, BUSINESS, EDUCATIONAL, GOVERNMENTAL AND INDIVIDUAL USERS.



An outside view of our corporate headquarters in Champaign, Illinois.

Optica Software News

Welcome to Ann Williamson!

Ann Williamson is our newest member of the Optica software team. She worked for many years as a scientific programmer on various science experiments. She previously worked at CERN (European Particle Physics Lab) on the LEP OPAL experiment, at the Stanford Synchrotron Radiation Laboratory, and at Lockheed Missiles and Space Company - Space Sciences Lab, where she worked on the UARS satellite project and Hubble Space Telescope Project. She has a physics and computer science background and has degrees from Tulane University, U.C. Berkeley, and Stanford University.

We recently asked for your assistance in completing our website survey. As our way of saying thank you, we are randomly selecting one completed survey each month and awarding a copy of our software to the person who completed it. Our recipient for September 2005 is Jennifer Shumaker-Parry, University of Utah. Please continue to fill out the survey for your chance to win.

We are currently seeking international resellers. Contact us at: support@opticasoftware.com to request a copy of our reseller application.

Events

We are scheduled to attend and exhibit at **Photonics West**, January 24-26, 2006 in San Jose, California. For more information: http://spie.org/app/exhibition/index.cfm?fuseaction=welcome&meeting_id=91. Please stay posted for details on a presentation by our developer Donald Barnhart.

CONTACT US:

Donald Barnhart donald@opticasoftware.com
Lorenzo Kindle lorenzo@opticasoftware.com
Ann Williamson annw@opticasoftware.com
Support support@opticasoftware.com
Website www.opticasoftware.com

Phone 217.328.9847

866.328.4298

Fax 217.328.9692

Wolfram Technology Conference

The software team recently attended the Wolfram Technology Conference October 6-8, 2005 in Champaign, Illinois. We met with the front-end developers of *Mathematica*[®] and attended some great presentations on what is new and upcoming. Needless to say it was very exciting! Stephan Wolfram gave an entertaining presentation on Wolfram Tones, an experimentation in a new kind of music, applying the mathematics of complexity theory to create music. We are happy to say that the development will reach a new level and as users of our software you will benefit as well. Please continue to support Wolfram Research and their products.

Creators of *Mathematica*[®]

USER TIPS

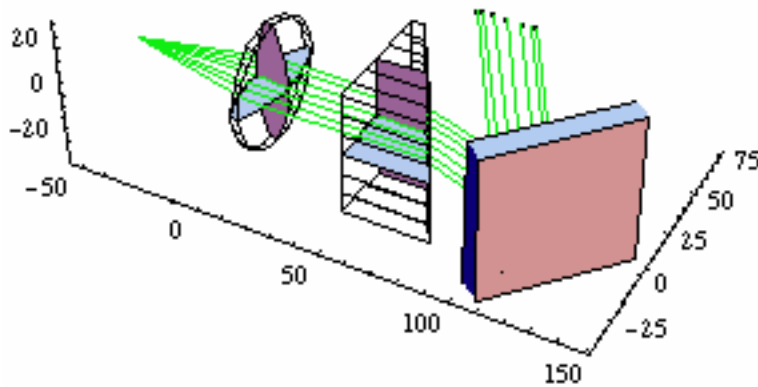
First Time Users - Sequential vs. Non-sequential Ray Tracing

You can use **Resonate** for performing non-sequential ray tracing between optical surfaces. **Resonate** takes a list of **Component** objects and creates a single **Component** object from the list. Once all of the surfaces are no longer in its trajectory, the ray is allowed to continue propagating to other components. This effect enables repeated non-sequential ray interactions with a number of surfaces to occur.

In this example the light source is a **WedgeOfRays** of spread angle 25 degrees. Optical components include a **BiConvexLens** centered at the origin, and a **WedgePrism** and **Mirror** positioned with **Move**. **Boundary** designates a rectangular box that absorbs rays intercepted by its walls. Note that with **Resonate**, we do not need to specify the components in the correct order.

```
source = WedgeOfRays[25, NumberOfRays ->7, BirthPoint -> {-50,0,0}];  
lens = BiConvexLens[60,50,14, "lens"];  
prism = Move[WedgePrism[30, {50,50},5,"prism"], 50];  
mirror = Move[Mirror[{60,60},10, "mirror", GraphicDesign -> Solid], {120, -10},-40];  
boundary = Boundary[150, GraphicDesign -> Off];
```

```
AnalyzeSystem[  
  { source, Resonate [{lens, mirror, prism}], boundary },  
  Axes -> True];
```



More details about sequential and non-sequential ray tracing can be found in the **Principles of Rayica guide**.

LED Optics

This example, which can be done in both *LensLab*[™] and *Rayica*[™], shows the ray trace for a LED lens. The Lens is modeled with a combination of **LensSurface** component functions: **LensSurface**, **CylindricalLensSurface**, and **SphericalLensSurface**. The **ConeOfRays** function models the light source.

Figures below

