foxsisim-gui Quickstart Guide

Robert Taylor
rlvtaylor@embarqmail.com

August 16, 2011

Introduction

The `foxsisim` python module is designed to simulate grazing incidence optics response to different light sources. The purpose of `foxsisim-gui` is to facilitate basic user operation with an easy to use GUI. Those familiar with the construction of optics modules such as the FOXSI telescope should find the user interface fairly intuitive. Nonetheless, this guide covers its basic use.

Module Settings

A complete optic is created with the `foxsisim.module.Module` class. The first tab in the GUI, named *Module*, facilitates module creation. There we see a table specifying *shell radii* and *angles* – shells being the individual 'tubes' within the module, each consisting of two mirrored *segments* (see Figure 1).

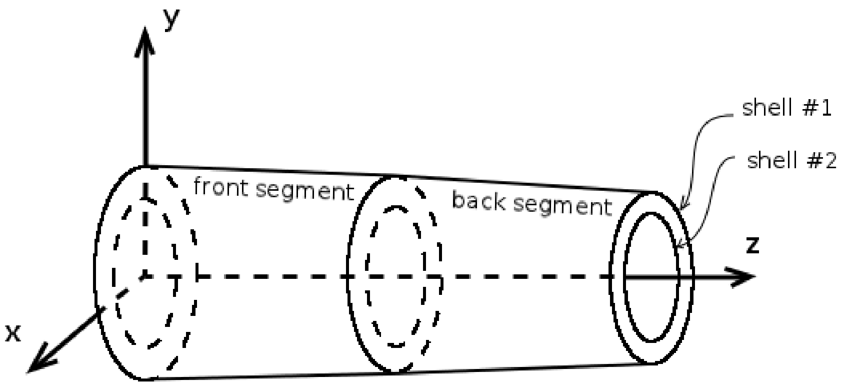

Figure 1: A diagram demonstrating how a multishell optics module is represented within the `foxsisim` python code. The module base (aperture center) is located at the origin and the module is aligned along the z axis. Rays must originate in the $-z$ region in order to pass through the module.

Shell radius refers to the distance from the z axis to the place where the shell's two segments meet. *Shell angle* refers to the angle between the front segment and the z axis.

There are additional important settings below the table. *Segment length* is self-explanatory, as are *Detector Width* and *Height*. *Focal length* is the distance from the center of the module to the focal point. *Detector Offset* is the distance along the z axis by which the detector is offset from the focal point. A positive offset value means that the detector is farther from the module; negative means closer. The *Detector Reso* is the pixel resolution of the detector.

Source Settings

Next comes the *Sources* tab. There we find a table that allows us to insert light sources of various *types* into the system:

- **atinf**: the source is rectangular (*Width* x *Height*) and all rays travel in exactly the same direction (*Normal*)
- **point**: all rays come from a single point (*Center*)
- **nonpoint**: the source is rectangular (*Width* x *Height*) and rays can leave it in any direction

Center is the location of the rectangular source. If the source is a point, then it is located at *Center*.

Normal determines how a rectangular source is oriented (does not apply to point sources). Furthermore, if the source is **atinf**, *Normal* is the direction in which rays will travel away from it.

Width and *Height* describe the dimension of the source's rectangular region. They do not apply to point sources.

Lastly, a source can be one of several *colors*. If the detector catches rays from a source, then its color will be used to calculate detector pixel values.

Simulation

The *Simulation* tab allows us to simulate our module and light sources and provides some basic plotting functions. Once all module, detector, and source settings have been entered, we can simulate the system by entering the number of rays and pressing the *Simulate* button. A simulation can easily be stopped and restarted without losing progress by using the *Stop* button. However, if we wish to enter new module/detector/source settings, the *Reset* button must be used.

A *Detector Pixel Plot* is simply the captured image of the detector. The detector creates the image by storing the average color of rays that hit each pixel.

A *Scatter Plot* displays the distribution of ray–detector collisions for one or more sources. Additionally, the *Color Bounce Count* option tells `foxsisim` to color each point in the plot based on how many times it bounces within the module. Usually this color will be blue (two bounces) – but under certain circumstances, points can be colored red (no bounces), green (one bounce), or black (more than two).