

Modeling a Simple Optical System

Numerically Modeling a One - Dimensional Optical System in Python 3

Landon Buell

PHYS 708.01 - Fall 2019

1 Abstract

2 Introduction

This project will see the creation of a python 3 program that can efficiently and accurately model a simple, one - dimensional optical system. For simplicity, the light source, or object that is *being* imaged, will always be placed at the origin of a coordinate system. All other optical components in the system will be confined to the positive *x-axis* between $x = 0$ and $x = 100$. Additionally, the entire system will be immersed in air where the index of refraction will be assumed as $n = 1$.

The program will revolve around a few basic optical principles as learned from class (PHYS 708 - Optics) as well as from various other references. For all computations, light will be assumed to be an perfectly idealized geometric ray. The exact wavelength of the ray will not change the index of refraction of the object as well.

3 General Methodology

The functionality of this project will rely on object-oriented programming.

3.1 Optical Components Object

The computational part of this program will involve the construction of a *class* object that will contain all attributes that are relevant to modeling the system. For a thick lens, this will involve a central position, two radii of curvature, an index of refraction as so forth. Each component of the system will be assumed to be perfectly ideal, containing no distortions of any sort.

3.2 Geometric Ray Object

The path that the ray takes through the optical system is modeled as a one dimensional *numpy array*. The floating point numbers contained within the arrays track the rays vertical position above the x-axis corresponding to evenly spaced x-value increments. All rays will be treated as ideal one-dimensional paths, and the exactly wavelength of frequency of the ray will not change any properties of the ray.

3.3 Visualization

Once the system has been tracked computationally, a visualization is produced. This visualization is done in Python's *matplotlib.pyplot* module and contains images of all optical components to scale, and the path that certain rays will take.

4 Example System #1

5 Example System #2

6 Additional Notes

- **Realistic Systems**

Due to the lack of detailed error handling, is it possible to attempt to model physically unrealistic systems. These may included (but not limited to): over lapping lenses, infinite or undefined values, etc.

- **Python Modules**

Required Python Modules: Numpy, Scipy, Matplotlib

7 Conclusion

8 References