Integrated Modeling of Observatories with Astronomical Coronagraphs using Hybrid Propagation Physics

Jaren Ashcraft

March 23, 2022

Abstract

High-contrast imaging systems require an accurate physical optics model to properly evaluate instrument performance. Commercial optical design packages and existing open-source physical optics tools achieve this by the propagation of a scalar optical field through the system via diffraction integrals. The scalar diffraction assumption obfuscates misalignment and polarization induced aberrations from the observatory, which have been shown to limit the sensitivity of coronagraphs to earth-like exoplanets. To compensate, the state of the art relies on separate models for the ray trace, polarization, and diffraction calculations. This approach complicates and decentralizes the design process, which can slow the development of astronomical coronagraphs. We propose to update POPPY (Physical Optics Propagation in PYthon) with Gaussian Beamlet Decomposition (GBD) to encapsulate the ray-based and wave-based behavior of an observatory equipped with a coronagraph simultaneously. GBD is a technique where the field at the entrance pupil is spatially decomposed into Gaussian beams, which can be propagated along ray paths for diffraction calculations without Fourier transformations. This approach unifies ray-based analyses with wave-based analyses, which can be a powerful design tool for astronomical observatories. We report on the development of an open-source GBD module tailored to the design of astronomical coronagraphs. The results are compared to traditional Fresnel diffraction models of a coronagraph, and opportunities for accelerated computing are explored.

1 Introduction

Traditional diffraction modeling regimes consider the optical system to be paraxial and the electromagnetic field to be essentially scalar.

Gaussian Beamlet Decomposition (GBD) is a ray-based method of diffraction calculation that approximates an optical field as a superposition of Gaussian beams. Gaussian beams are unique in that they can be propagated along ray

paths. In their seminal paper, Harvey et al[**?**] reviews the theory of complex ray tracing used to propagate Gaussian beams.

2 Gaussian Beamlet Decomposition

The idea is to decompose a field (in spatial or angle space) with a finite sum of gaussian beamlets. We do this so that we can do diffraction calculations from real ray data! This allows for computation of wavefront errors that arise from geometrical errors, diffraction on generally rough optical surfaces, and polarization from the optical elements.

Most of the math for beamlets is in Ashcraft and Douglas 2019, but let's start with the gaussian beamlet equation. The paraxial gaussian beam is defined by the following expression:

$$\phi(x, z) = A\sqrt{q(z)}e^{i\pi q(z)x^2} \tag{1}$$

Where ϕ is our elementary beam, A is the amplitude of the beamlet, x is a spatial coordinate in one dimension, and q is the complex beam parameter, given by

$$\frac{1}{q} = \frac{1}{R(z)} - \frac{i\lambda}{\pi w(z)^2} \tag{2}$$

Where $R(z)$ is the radius of curvature and $w(z)$ is the 1/e beamlet waist radius. We use these beamlets to populate the entrance pupil of an optical system to approximate a top-hat wavefront. This is done by just shifting the beams to fill out the pupil. We already reproduced the techniques' ability to recreate the PSF of an observatory with some arbitrary entrance pupil shape without phase error.

3 Differential Ray Tracing

$$\begin{pmatrix} x_1' & y_1' & u_1' & v_1' \\ x_2' & y_2' & u_2' & v_2' \\ x_3' & y_3' & u_3' & v_3' \\ x_4' & y_4' & u_4' & v_4' \end{pmatrix} = \begin{pmatrix} A_{xx} & A_{xy} & B_{xx} & B_{xy} \\ A_{yx} & A_{yy} & B_{yx} & B_{yy} \\ C_{xx} & C_{xy} & D_{xx} & D_{xy} \\ C_{yx} & C_{yy} & D_{yx} & D_{yy} \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ u_1 & u_2 & u_3 & u_4 \\ v_1 & v_2 & v_3 & v_4 \end{pmatrix} \tag{3}$$

To convert real ray data into a form that the ray transfer matrix is familiar with we need to do a bit of math. A given real ray is specified by

$$\overline{R_{out}} = \overline{O_{sys}R_{in}} \tag{4}$$

$$\vec{r}_{real} = \begin{pmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{pmatrix} \tag{5}$$

Where x, y, z are the positions of the ray intercept and α, β, γ are the direction cosines of the ray at the intercept. A ray for transfer matrices at a position z is given by

$$\vec{r}_{trans} = \begin{pmatrix} x \\ y \\ u \\ v \end{pmatrix} \tag{6}$$

Where u, v are the ray slopes in the x, y directions respectively. These slopes are computed in the x-z plane and y-z plane, so some angular conversion must be done. A given plane can be defined by a vector normal to the plane. $\vec{y}$ is normal to the x-z plane, and $\vec{x}$ is normal to the y-z plane. The angle of a real ray w.r.t. a given vector is simply the dot product

$$\vec{r} \cdot \vec{x} = |\vec{r}|cos(\theta_x) = |\vec{r}|\alpha \tag{7}$$

$$\theta_x = cos^{-1}(\frac{\vec{r} \cdot \vec{x}}{|\vec{r}|}) = cos^{-1}(\alpha) \tag{8}$$

Where the slope in a given plane can be found by determining the angle and subtracting π

$$v = tan(\theta_x - \pi) \tag{9}$$

$$u = tan(\theta_y - \pi) \tag{10}$$

4   ArbitraryWFE Ray Transfer Matrix

Jeong et al demonstrate a ray transfer matrix for some arbitrary wavefront error, but use it to derive expressions to model Zernike modes. The theory starts with a surface S at a position z with some wavefront deformation $W(x, y)$. The space of surface normals on this surface is defined by the gradient

$$\nabla S(x, y) = z - \frac{\partial W(x, y)}{\partial x} - \frac{\partial W(x, y)}{\partial y} \tag{11}$$

For an arbitary wavefront W_1 incident on S, and an exitant wavefront W_2, the wavefront aberration induced by the optic is $W_2 - W_1$. The angular deviation in each direction is given by

$$\alpha, \beta = tan^{-1}(\frac{\partial W(x,y)_2}{\partial x, y}) - tan^{-1}(\frac{\partial W(x,y)_1}{\partial x, y}) \tag{12}$$

In the paraxial limit the tangents disappear and we get the resulting expression:

$$\theta_2 = \theta_1 - \frac{\partial W(x,y)}{\partial x, y} \tag{13}$$

This can be formatted into a 4x4 matrix with

$$\begin{pmatrix} x_2 \\ y_2 \\ u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{-1}{x_1}\frac{\partial W}{\partial x} & 0 & 1 & 0 \\ 0 & \frac{-1}{y_1}\frac{\partial W}{\partial y} & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ u_1 \\ v_1 \end{pmatrix} \tag{14}$$

Which looks exactly like a ray transfer matrix, and can consequently be applied to paraxial raytraces of optical systems. Equation 14 is a generalization of expressions for wavefront error deformation. For example, if $W = \frac{(x^2+y^2)}{2f}$ for a thin lens model of wavefront deformation, the gradients (W_x, W_y) are:

$$W_x = \frac{x}{f}; W_y = \frac{y}{f} \tag{15}$$

Which result in the thin lens ray transfer matrix. For sources of arbitrary wavefront error (e.g. from polishing) the gradient must be computed numerically. We present a Python implementation of computing the wavefront error per-ray.

4