POLARIZATION ABERRATIONS IN CORONAGRAPHS

by

Jeffrey M. Davis

A Dissertation Submitted to the Faculty of the

COLLEGE OF OPTICAL SCIENCES

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2019

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by *Jeffrey Davis*, titled *Polarization Aberrations in Coronagraphs* and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Russell Chipman
Russell Chipman Date: 6/11/2019

James Schwiegerling Date: 6/11/2019

James B. Breckinridge
James Breckinridge Date: 11 June 2019

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Russell Chipman
Russell Chipman Date: 6/11/2019
Dissertation Committee Chair
Optical Sciences

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Russell Chipman, for pushing me to my limits and giving me advice so I could continue to think deep thoughts about polarization. This dissertation would not have been completed without him. I would also like to thank him for his love of Mathematica, which has rubbed off on me. I'm so sorry Matlab, there's a new love in my life now. I would also like to thank Dr. Meredith Kupinski for all her help and suggestions on the technical details of my simulations. Without her help, this project would have taken so much longer.

To my lab-mates Lisa and kira, thank you for being around so that I had someone to kvetch to. Having others who knew the struggle really helped keep my spirits up. You guys are the bees knees.

To my friends Alex, Emily, Jon, and Justin, our board game nights were the greatest escape from the pressures of getting this dissertation done. I likely would have gone insane without you guys.

A huge shoutout also goes to my committee. My conversations with Dr. Breckinridge are always enlightening and jovial. And though I didn't end up working for him, I'm deeply thankful to Dr. Schwiegerling for giving me my first opportunity to work in a lab in the College of Optical Sciences.

DEDICATION

For Zach Senft, the best friend a man could have.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In the search for habitable extrasolar planets, the ability to separate and detect the dim planet's light from its much brighter host star is paramount. The inherent polarization properties of optical systems can lead to small but significant deviations from ideal imaging behavior, possibly hindering the ability of that system to detect exoplanets. In this work, the polarization aberrations in a telescope/coronagraph optical system are modeled with polarization ray tracing software. The effect of these polarization aberrations on the coronagraph's ability to suppress on-axis starlight is analyzed, and an algorithm for mitigation of some of the polarization aberrations is provided.

This dissertation lays the foundation for modeling and analyzing polarization aberrations in telescope/coronagraph systems and shows that, in the absence of adaptive optics, both isotropic and anisotropic polarization aberrations degrade a coronagraph's ability to suppress starlight. The contrast is a measure of on-axis starlight suppression. With isotropic polarization aberrations, the contrast is degraded by two orders of magnitude relative to the ideal, polarization aberration-free case. Anisotropy modeled on the primary mirror further degraded the contrast by two orders of magnitude relative to the isotropic case. Modeling polarization aberrations shows how coatings negatively affect the possible contrast of a coronagraph.

CHAPTER 1

Introduction

NASA astrophysics asks three big questions[1]: **how does the universe work? How did we get here? Are we alone?** In an effort to find answers these questions, four large mission concepts studies[2] (LUVOIR, HabEx, OST, and Lynx) have been undertaken for the 2020 Decadal Survey. These mission concepts will strive to explore neighboring planetary systems around sun-like stars, explore the growth of black holes and galaxies, and probe the origins and underlying physics of the cosmos across a broad range of wavelengths of light. One of the common characteristics of each of these mission concepts is the assessment of habitable conditions of planets orbiting distant stars (exoplanets).

To date, nearly 4,000 exoplanets have been discovered[3]. These planets range from large, gas giants to small, rocky worlds. As telescope technology advances, so too does our ability to detect fainter exoplanets. In the search for extra-terrestrial life, this ability to detect fainter earth-like planets in the habitable zone around distant stars is indispensable. Current techniques for detecting exoplanets include radial velocity, astrometry, transit, gravitational microlensing, and direct imaging[4, 5, 6, 7, 8, 9, 10, 11].

The radial velocity method uses Doppler measurements of a star to calculate the period, distance, and shape of orbit and the mass of the orbiting planet.

Astrometry uses plane of sky variations in a star's position to yield the true inclination and orientation of a planetary orbit. Transit detection is possible when an exoplanet transits across our line of sight of a star when making its orbit. In this case, the star will exhibit brief, periodic fluctuations in intensity which signals the presence of an orbiting planet. Gravitational microlensing detects planets by the direct gravitational perturbation of a background source of light by a foreground planet. Direct imaging of exoplanets seeks to separate and directly detect the light coming from an exoplanet from its parent star. As opposed to the indirect imaging techniques, direct imaging of exoplanets allows for the acquisition of the spectra of the exoplanets. Spectroscopy of an exoplanet provides one of the only feasible ways to determine the habitability of an exoplanet in a star's habitable zone. Thus finding earth-like exoplanets and characterizing their atmospheres in the search for extra-terrestrial life depends on accurate and precise direct imaging techniques.

There are two main techniques to disentangle a parent star's photons from an exoplanet's photons: a coronagraph and a starshade. Coronagraphs[12, 13] act to remove the parent star's light inside of a telescope while starshades[14] fly outside of the telescope and externally block the parent star's light[15]. These instrument have differing capabilities and limitations, and some current telescope designs utilize a hybrid system that uses both a coronagraph and starshade[16]. Whether through a starshade, coronagraph, or combination thereof, the ultimate goal is to directly image a distinct image of the exoplanet orbiting a distant star and examine its spectra for signs of the ability to support life[17].

1.1 Motivation and Research Objectives

Current coronagraph designs are planned to allow future space based telescopes to directly image faint earth-like exoplanets, whose reflected light is around 10 billion times fainter than the parent star[11]. With such extreme starlight suppression, any small source of error in the telescope can potentially limit the telescope/coronagraph system's ability to detect an exoplanet. One such source of error that tends to be overlooked is the inherent polarization properties of the surfaces that make up an optical system. According to Breckinridge et. al[18], "[t]he point spread function for astronomical telescopes and instruments depends not only on geometric aberrations and scalar wave diffraction but also on those wavefront errors introduced by the physical optics and the polarization properties of reflecting and transmitting surfaces within the optical system." The research objective of this dissertation is to seek answers to these questions: do polarization effects limit a telescope's ability to detect exoplanets? And what can be done about polarization aberrations to compensate for or mitigate their effects?

1.2 Methodology

To address these issues, a way to model the effects of polarization in optical systems and coronagraph designs is needed. Additionally, the design of a current or multiple current telescope and coronagraph designs are essential.

The modeling and analysis software this dissertation utilizes is Polaris-M[19, 20, 21], polarization ray tracing software that incorporates a suite of optical analysis functions written in Wolfram's Mathematica. Polaris-M was

developed from the ground up at the University of Arizona to incorporate many advanced polarization analysis features. It has the added advantage of residing in Mathematica, giving Polaris-M access to the vast functionality of Mathematica.

While Polaris-M is fully capable of modeling the polarization effects of an optical system via polarization ray tracing, some coronagraph effects require diffraction algorithms. So in conjunction with Polaris-M, coronagraph modeling code is implemented in MATLAB[22, 23].

For current telescope and coronagraph designs, the Jet Propulsion Laboratory provided the optical layout for their habitable exoplanet observatory and the designs for the coronagraph it will use. Additionally, Goddard Space Flight Center provided the optical layout for their large UV, optical, and infrared surveyor mission.

By utilizing Polaris-M's polarization ray tracing, coronagraph modeling code, and the provided telescope designs, this dissertation analyzes the effect of polarization in coronagraph systems. The steps are: (1) import the optical systems provided into Polaris-M, (2) use Polaris-M's polarization ray tracing functions to determine polarization effects in the optical system without the coronagraph, (3) use diffraction algorithms developed in MATLAB with Polaris-M inputs to determine polarization effects with the coronagraph, and finally (4) compare the results from the diffraction algorithms with polarization considerations versus the outputs with no polarization considerations. The analysis presented in this dissertation shows how coatings negatively affect the contrast of a coronagraph. Therefore, polarization should be taken into account when

designing adaptive optics and wavefront control lest the faint exoplanet's signal be masked by polarization effects.

1.3 Dissertation Outline

This dissertation is broken into seven parts. Chapter 1 (this chapter) introduces the topic of this dissertation and provides some general background information. Chapter 2 provides the theoretical groundwork necessary to understand polarized light, polarization ray tracing, and polarization aberrations. It also defines two polarization properties, diattenuation and retardance, and details how Polaris-M is used. In chapter 3, the basic foundation of image formation in an imaging system is covered, an explanation of how coronagraphs operate is supplied, and a detailed example using a specific type of coronagraph, a vector vortex, is given. Chapter 4 analyzes the optical layouts for the two real optical systems, HabEx and LUVOIR, provided for this dissertation. The analysis of these systems in Polaris-M is explained, and calculation of the contrast from the coronagraphs (with and without polarization aberrations) is provided. Chapter 5 talks about mitigation techniques for polarization aberrations and presents an alternate optical layout to minimize the polarization aberrations in the HabEx system. In chapter 6, Polaris-M is utilized to calculate longitudinal chromatic aberration from dielectric coated reflecting surfaces. Lastly, chapter 7 is the concluding chapter.

CHAPTER 2

Polarization and Polaris-M

This chapter gives a brief introduction to the formalism used to describe polarization, defines a few key parameters associated with polarization, and describes a few of the key functionalities of the polarization ray tracing program, Polaris-M.

2.1 Polarized Light

Electromagnetic radiation travels as a wave, where the electric and magnetic portions of the field each oscillate transverse to both each other and the direction of propagation of the light [24]. The simplest light wave is the plane wave, which has an electric field at a point in space, $\mathbf{r}$, and time, t, given by[25]:

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}_0 e^{i\frac{2\pi}{\lambda}\mathbf{k}\cdot\mathbf{r}-\omega t} \tag{2.1}$$

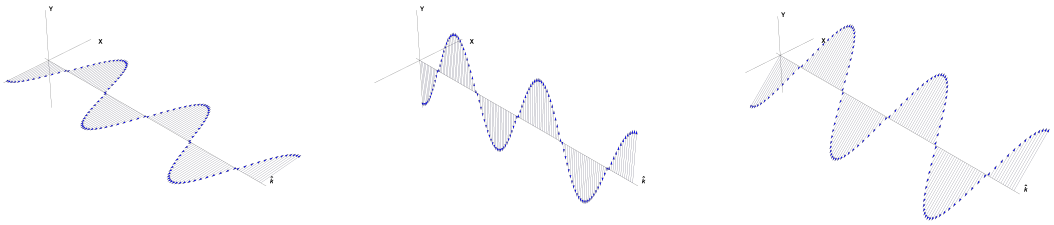
λ is the free space wavelength of the light, $\mathbf{k}$ is the direction of propagation of the wave, and ω is the frequency of the wave given by the relationship $c = \omega\lambda/(2\pi)$. c is the speed of light in vacuum. The polarization of the light is the direction of oscillation of the electric field, given by $\mathbf{E}_0$. For light traveling along the z-axis, polarized light can be decomposed into two orthogonal components along the x- and y-directions.

$$E_x(z,t) = E_{x0}e^{i\frac{2\pi}{\lambda}z-\omega t} \tag{2.2}$$

$$E_y(z,t) = E_{y0}e^{i\frac{2\pi}{\lambda}z-\omega t+\phi} \tag{2.3}$$

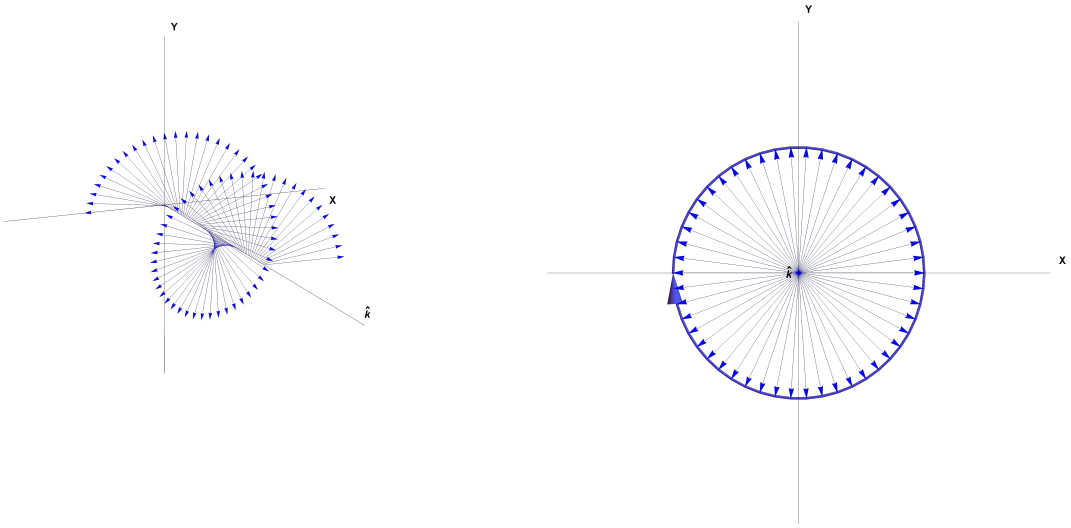
ϕ is the relative phase between the x- and y-electric fields. When $\phi = 0$ or $\phi = \pi$, the two fields will trace out a line in the x-y plane as z or t varies. Such a case is deemed linear polarization. When $\phi = \pi/2$ and $E_{x0} = E_{y0}$, the two electric fields will trace out a circle as z or t varies. This is the circular polarization case. For all other cases, the two electric fields will trace out an ellipse, creating elliptical polarization.

For example, figure 2.1 shows three cases of linearly polarized light: horizontally polarized light will oscillate along the x-axis, vertically polarized light will oscillate along the y-axis, and 45° linearly polarized light will oscillate at an angle of 45° counterclockwise relative to the x-axis. Circularly polarized light, shown in figure 2.2, will rotate around in the x-y plane, with the tip of the electric field direction tracing out a circle as the wave goes through a full cycle of oscillation. When the amplitude of the two orthogonal components of the polarization are unequal and the relative phase between the two components is not $0, \pi$, or $\pi/2$, the resulting polarization state is elliptical, as seen in figure 2.3.



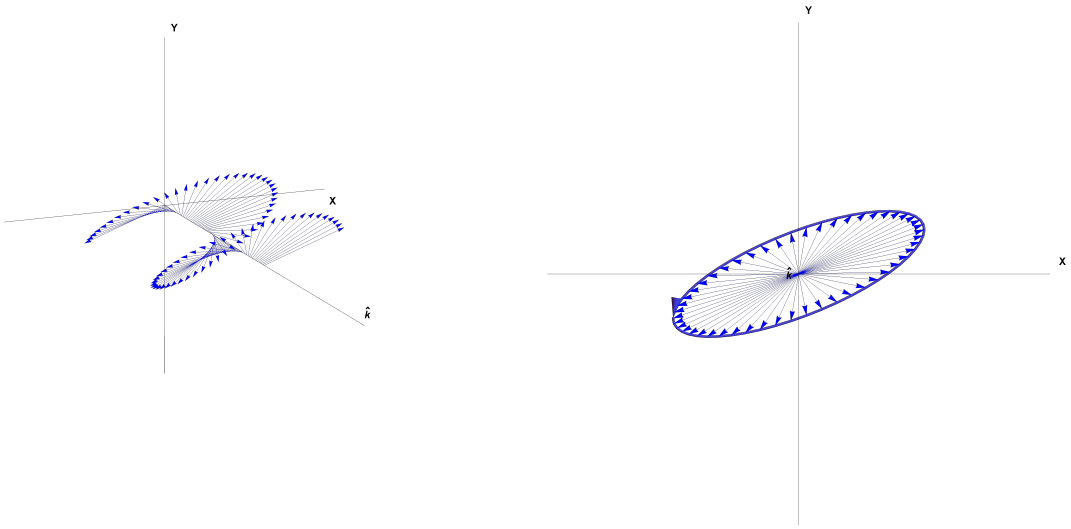
(a) (b) (c)

Figure 2.1: Light propagates as an EM wave. Linearly polarized light has an electric field that oscillates in a preferential linear direction. Different linear polarization states include (a) x-polarized, (b) y-polarized, and (c) 45° polarized



(a) (b)

Figure 2.2: Circularly polarized light has an electric field that rotates in the transverse plane as the light goes through a full cycle. (a) shows circular light as viewed from an arbitrary angle with respect to the propagation direction. (b) shows circular light as viewed down the axis of propagation with a superimposed circle showing the path the circular polarization traces out as it propagates down the axis.



(a) (b)

Figure 2.3: Elliptical polarized light occurs when the two orthogonal components of the polarized light have unequal amplitudes and the relative phase between the two is not $0, \pi$, or $\pi/2$. (a) shows elliptical light as viewed from an arbitrary angle with respect to the propagation direction. (b) shows elliptical light as viewed down the axis of propagation with a superimposed ellipse showing the path the polarization traces out as it propagates down the axis.

2.1.1 Jones Calculus

A common mathematical description of polarized light and polarization properties of optical elements is the Jones formalism[26, 27, 28, 29, 30, 31, 32, 33]. By making the assumption that the plane wave's direction of propagation is along the z-axis, Jones calculus uses two element Jones vectors, **E**, to describe polarized light, and two by two Jones matrices, **J**, to describe polarizing elements.

$$\mathbf{E} = \begin{bmatrix} E_x \\ \\ E_y \end{bmatrix} \tag{2.4}$$

$$\mathbf{J} = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix} \tag{2.5}$$

Equation 2.4 shows that light traveling along the z-axis can be decomposed into two orthogonal components; a component along the x-axis, E_x, and a component along the y-axis, E_y. If the light is *not* traveling along the z-axis, then the two components of the Jones vector instead represent local coordinates associated with a plane transverse to the direction of propagation [34]. When an incident Jones vector, $\mathbf{E_1}$, interacts with a polarizing element, the resulting electric field after the interaction, $\mathbf{E_2}$, is given by the multiplication of the Jones matrix with the input Jones vector:

$$\mathbf{E}_2 = \mathbf{J} \cdot \mathbf{E}_1 = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix} \cdot \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} J_{xx}E_x + J_{xy}E_y \\ J_{yx}E_x + J_{yy}E_y \end{bmatrix} \tag{2.6}$$

As can be seen from equation 2.6, the elements of the Jones matrix specify how much of the original polarization states exits in the same orientation (*e.g.* J_{xx} maps input x-polarized light into output x-polarized) and how much exits in the orthogonal orientation (*e.g.* J_{xy} maps input y-polarized light into output x-polarized light.) Each polarizing elements in a system has its own Jones matrix, and the total system Jones matrix can be represented by the matrix multiplication of the sequence of Jones matrices the light passes through. It is important to note that in equation 2.7, the first Jones matrix the light interacts with is the rightmost Jones matrix while the last Jones matrix is the leftmost.

$$\mathbf{J}_{total} = \mathbf{J}_Q \cdot \mathbf{J}_{Q-1} \ldots \mathbf{J}_2 \cdot \mathbf{J}_1 = \prod_{q=Q}^{1} \mathbf{J}_q. \tag{2.7}$$

2.1.2 Polarization Properties: Diattenuation and Retardance

The diattenuation of a polarizing element is a measure of how strongly it polarizes the light passing through. It can also be defined as the exiting degree of polarization when unpolarized light is incident upon a polarizing element. A retarder is an optical element that causes a polarization dependent optical path length (phase) change in the light that interacts with the retarder. The calculation of diattenuation and retardance from a Jones matrix depends on the properties of the Jones matrix.

Jones matrices generally fall into one of two categories: homogeneous or inhomogeneous[35]. Which category a Jones matrix falls under depends on the eigenvectors of the Jones matrix. The eigenvectors of a Jones matrix (also known as eigenpolarizations) are the two input Jones vectors that are the same as the output Jones vector for that Jones matrix (scaled by an eigenvalue). The two eigenpolarizations, $\mathbf{E}_q$ and $\mathbf{E}_r$, of the Jones matrix $\mathbf{J}$ have the property:

$$\mathbf{J} \cdot \mathbf{E}_q = \xi_q \mathbf{E}_q = A_q e^{-i\phi_q} \mathbf{E}_q$$

$$\mathbf{J} \cdot \mathbf{E}_r = \xi_r \mathbf{E}_r = A_r e^{-i\phi_r} \mathbf{E}_r \tag{2.8}$$

ξ_q and ξ_r are the complex eigenvalues corresponding to the eigenpolarizations, A_q and A_r are the amplitudes of the eigenvalues, and ϕ_q and ϕ_r are the phases of the eigenvalues. When $\mathbf{E}_q$ and $\mathbf{E}_r$ are orthogonal ($\mathbf{E}_q^\dagger \cdot \mathbf{E}_r = 0$), the corresponding Jones matrix is homogeneous. When $\mathbf{E}_q$ and $\mathbf{E}_r$ are not orthogonal

($\mathbf{E}_q^\dagger \cdot \mathbf{E}_r \neq 0$), the corresponding Jones matrix is inhomogeneous.

2.1.2.1 Homogeneous Jones Matrices

In the case of a homogeneous Jones matrix, the two polarization properties diattenuation and retardance are straightforward to calculate. Diattenuation and retardance of a homogeneous Jones matrix are defined as:

$$\mathbf{D} = \frac{A_q^2 - A_r^2}{A_q^2 + A_r^2}, \quad 0 \leq \mathbf{D} \leq 1 \tag{2.9}$$

$$\delta = |\phi_r - \phi_q|, \quad 0 \leq \delta \leq \pi \tag{2.10}$$

A diattenuation of zero indicates that the element transmits all polarization equally, while a diattenuation of 1 indicates that the element is a perfect polarizer, allowing only one polarization state to be transmitted. The retardance measures the phase difference between eigenpolarizations after interaction with a retarder. Common retarding elements are quarter-wave ($\delta = \pi/2$) and half-wave ($\delta = \pi$) retarders.

The classification of a homogeneous Jones matrix depends on its diattenuation, retardance, and eigenpolarizations. Jones matrices with eigenpolarizations that are linear are classified as linear diattenuators or linear retarders. Similarly, circular eigenpolarizations lead to circular diattenuators and circular retarders while elliptical eigenpolarizations lead to elliptical diattenuators and elliptical retarders.

2.1.2.2 Inhomogeneous Jones Matrices

For inhomogeneous Jones matrices, calculating the diattenuation and retardance requires a few extra steps. Using polar decomposition[36], a Jones matrix can be expressed as the product of a purely retarding unitary matrix (**U**) and purely diattenuating Hermitian matrix (**H**)

$$\mathbf{J} = \mathbf{UH} \tag{2.11}$$

These unitary and Hermitian components are calculated using singular value decomposition[37]

$$\mathbf{J} = \mathbf{W\Sigma V}^{\dagger} = \begin{pmatrix} w_{x,1} & w_{x,2} \\ w_{y,1} & w_{y,2} \end{pmatrix} \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \begin{pmatrix} v^*_{x,1} & v^*_{y,1} \\ v^*_{x,2} & v^*_{y,2} \end{pmatrix} \tag{2.12}$$

W and **V** are unitary matrices and $\mathbf{\Sigma}$ is a diagonal matrix of singular values. The maximum amplitude transmission for **J** is Λ_1, corresponding to input state $(v_{x,1}, v_{y,1})$ and output state $(w_{x,1}, w_{y,1})$. Similarly, the minimum transmission is is Λ_2, corresponding to input state $(v_{x,2}, v_{y,2})$ and output state $(w_{x,2}, w_{y,2})$. The diattenuation of **J** is then

$$\mathbf{D} = \frac{\Lambda_1^2 - \Lambda_2^2}{\Lambda_1^2 + \Lambda_2^2} \tag{2.13}$$

The unitary and Hermitian components are thus calculated as

$$\mathbf{U} = \mathbf{WV}^{\dagger} \tag{2.14}$$

$$\mathbf{H} = \mathbf{V\Sigma V}^{\dagger} \tag{2.15}$$

The retardance of the pure retarder is found by calculating the eigenvalues of **U** and taking the difference of the phase of those eigenvalues.

2.2 Polarization Ray Tracing Calculus

The caveat for using Jones calculus is that the light under consideration must be propagating in only one direction, with local coordinates transverse to the direction of propagation defining the two orthogonal components of the electric field. To use Jones calculus in optical systems which have changing directions of propagation of the light (either by reflection or refraction of a single ray or a curved wavefront with different propagation directions for each ray), a local coordinate system is required for each ray segment. These local coordinate systems lead to complications due to the intrinsic singularities of local coordinates[38]. To avoid carrying the local coordinates along with each ray segment, a method for performing polarization ray tracing in three dimensions, polarization ray tracing calculus, is used in lieu of Jones calculus.

2.2.1 PRT Matrices

A polarization ray tracing (PRT) matrix, **P**, is a three by three matrix that describes the interaction of a three dimensional electric field with a polarizing element in global coordinates[39]. Additionally, the **P** matrix describes how the direction of propagation will change after interacting with a polarizing element.

$$\mathbf{E}_{out} = \mathbf{P} \cdot \mathbf{E}_{in} \tag{2.16}$$

$$\hat{\mathbf{k}}_{out} = \mathbf{P} \cdot \hat{\mathbf{k}}_{in} \tag{2.17}$$

Similar to Jones calculus, the total effects of an optical system can be condensed to the matrix multiplication of each ray's $\mathbf{P}$ at each surface.

$$\mathbf{P}_{total} = \mathbf{P}_Q \cdot \mathbf{P}_{Q-1} \ldots \mathbf{P}_2 \cdot \mathbf{P}_1 = \prod_{q=Q}^{1} \mathbf{P}_q. \tag{2.18}$$

The transformation from a Jones matrix to a PRT matrix, or vice versa, is a straightforward sequence of rotations into and out of local coordinates for each surface interaction. Since Jones matrices represent the electric field in a plane transverse to the direction of propagation, the s- and p-polarization states and the propagation vector ($\hat{\mathbf{k}}$) form a natural basis set for local coordinates. Given the incident and exiting propagation vectors of a ray from a surface, $\hat{\mathbf{k}}_{in}$ and $\hat{\mathbf{k}}_{out}$, the incident and exiting s- and p- directions are given by:

$$\hat{\mathbf{s}}_{in} = \frac{\hat{\mathbf{k}}_{in} \times \hat{\mathbf{k}}_{out}}{|\hat{\mathbf{k}}_{in} \times \hat{\mathbf{k}}_{out}|}, \quad \hat{\mathbf{p}}_{in} = \hat{\mathbf{k}}_{in} \times \hat{\mathbf{s}}_{in} \tag{2.19}$$

$$\hat{\mathbf{s}}_{out} = \hat{\mathbf{s}}_{in}, \quad \hat{\mathbf{p}}_{out} = \hat{\mathbf{k}}_{out} \times \hat{\mathbf{s}}_{out} \tag{2.20}$$

Equation 2.19 breaks down at normal incidence. In this special case, the s- and p- coordinates are degenerate and can be chosen arbitrarily (as long as $(\hat{\mathbf{s}}, \hat{\mathbf{p}}, \hat{\mathbf{k}})$ form a right hand system). Two orthogonal rotation matrices are constructed from the incident and exiting $(\hat{\mathbf{s}}, \hat{\mathbf{p}}, \hat{\mathbf{k}})$ coordinates to rotate between global and local coordinates. These orthogonal matrices are

$$\mathbf{O}_{in}^{-1} = \begin{pmatrix} \hat{s}_{in,x} & \hat{s}_{in,y} & \hat{s}_{in,z} \\ \hat{p}_{in,x} & \hat{p}_{in,y} & \hat{p}_{in,z} \\ \hat{k}_{in,x} & \hat{k}_{in,y} & \hat{k}_{in,z} \end{pmatrix} \tag{2.21}$$

$$\mathbf{O}_{out} = \begin{pmatrix} \hat{s}_{out,x} & \hat{p}_{out,x} & \hat{k}_{out,x} \\ \hat{s}_{out,y} & \hat{p}_{out,y} & \hat{k}_{out,y} \\ \hat{s}_{out,z} & \hat{p}_{out,z} & \hat{k}_{out,z} \end{pmatrix} \tag{2.22}$$

$\mathbf{O}_{in}^{-1}$ operates on the incident electric field defined in global coordinates and projects it into the local coordinates of the surface. $\mathbf{O}_{out}$ does the opposite, operating on the electric field in local coordinates to project it into global space. A PRT matrix is formed by

$$\mathbf{P} = \mathbf{O}_{out} \cdot \mathbf{J} \cdot \mathbf{O}_{in}^{-1} \tag{2.23}$$

where $\mathbf{J}$ is the Jones matrix of the surface interaction. For reflecting or refracting surfaces, the Jones matrix is defined in terms of the Fresnel reflection or transmission coefficients

$$\mathbf{J}_t = \begin{pmatrix} t_s & 0 & 0 \\ 0 & t_p & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{J}_r = \begin{pmatrix} r_s & 0 & 0 \\ 0 & r_p & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.24}$$

t_s, t_p, r_s, and r_p are the (complex) Fresnel transmission and reflection coefficients for the surface. Conversely, the Jones matrix of a PRT matrix can be determined by rearranging equation 2.23.

$$\mathbf{J} = \mathbf{O}_{out}^{-1} \cdot \mathbf{P} \cdot \mathbf{O}_{in} \tag{2.25}$$

This Jones matrix is a three by three quantity due to the three dimensional nature of the PRT. However, the third row and column don't carry any polarization information and can be dropped. The actual Jones matrix is simply the upper left two by two portion of the three by three Jones matrix.

2.2.2 Converting PRTs to a Jones Pupil

While the three dimensional description of polarization using PRT matrices provides a great workaround for the problem of tracking local coordinates, the three dimensional nature of the results can be hard to communicate. Visualization of the three dimensional nature of the PRT matrices can be difficult, and two dimensional descriptions of polarization information are much more easily communicable. The most common way to present polarization aberrations is as a set of spatially dependent Jones matrices, the Jones pupil [40]. The Jones pupil describes the linear mapping of the polarization state of rays at the entrance pupil to the polarization state at the exit pupil and it contains all of the polarization effects of the system as a function of position in the exit pupil.

To calculate the Jones pupil, the polarization ray tracing program Polaris-M traces a grid of rays to the exit pupil of an optical system and extracts the cumulative PRT matrix (equation 2.18) for each ray, creating a grid of cumulative PRTs. The program then transforms the PRT matrix into a Jones matrix for each ray in the grid by using equation 2.25. When the cumulative

PRT is used, $\mathbf{O}_{in}$ and $\mathbf{O}_{out}$ in equations 2.21 and 2.22 are defined by the local coordinate systems entering and exiting the system rather than the local coordinates entering and exiting a surface. Two common sets of local coordinates for spherical surfaces, dipole and double pole coordinates, are shown in Appendix A. The result of this transformation is a grid of Jones matrices, where each individual Jones matrix is associated with a particular pupil coordinate. As an example, a simple parabolic reflecting surface's Jones pupil is calculated utilizing Polaris-M. The parabolic mirror, shown in figure 2.4, has a focal length of 12.5 mm, a diameter of 20 mm, and is coated with a thick layer of aluminum. The index of refraction for this aluminum is $n = 0.666 + 5.58i$. To sample the properties of the mirror, a collimated rectangular grid of rays (51 ray across by 51 rays wide, total of 2601 rays) at a wavelength of 500 nm is traced from a distance of 10 mm before the mirror until the focal point of the mirror. The grid of rays is spaced such that it overfills the mirror, so some of the rays don't make it to focus. In this simple system, the entrance pupil, stop, and exit pupil are all the mirror surface, so the Jones pupil will describe the polarization properties of the rays after they interact with the mirror.

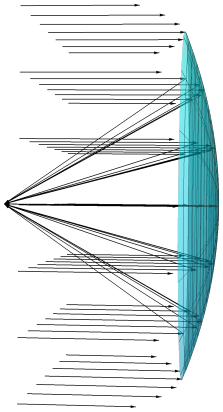
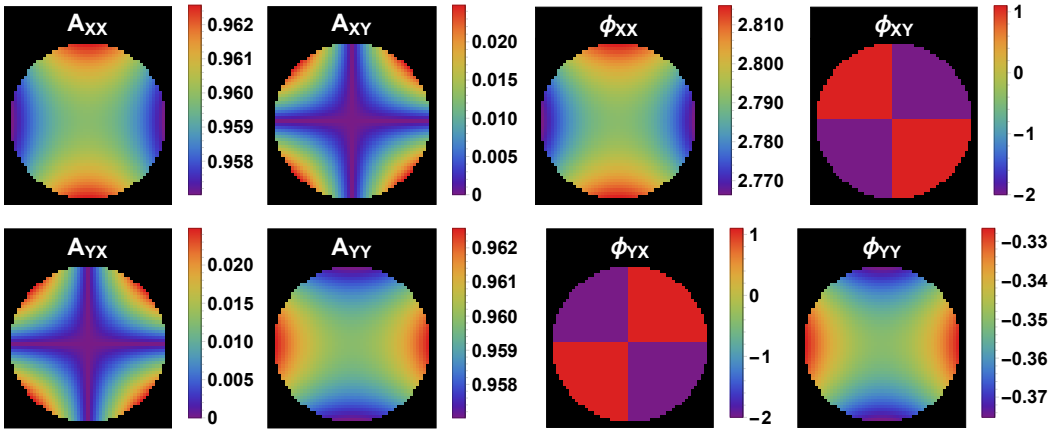


Figure 2.4: A simple parabolic reflecting surface modeled in Polaris-M. The focal length is 12.5 mm and the diameter is 20 mm. A grid of rays enters from the left, reflects off of the parabolic mirror, then comes to focus.

Once the grid of rays has been traced through the system, Polaris-M takes the rays at the exit pupil and converts their PRTs to Jones matrices. The Jones matrices are then ordered into a Jones pupil, show in figure 2.5. The Jones pupil has a circular shape because the mirror's surface is circular, cutting off any rays that fell outside of the mirror's circular support. The amplitude of the on-diagonal elements are compliments of one another, with the "xx" amplitude having minimums along the horizontal and maximums along the vertical while the "yy" amplitude has minimums along the vertical and maximums along the horizontal. Similar behavior can be seen in the on-diagonal phase of the Jones pupil. The off-diagonal amplitudes are quite small, indicating that there is not much polarization mixing for this system. The off-diagonal phase maps have jump discontinuities, going through a π phase shift at points in the pupil where the amplitude goes through a zero.



(a) (b)

Figure 2.5: Amplitude (a) and phase in radians (b) of the Jones pupil for a simple parabolic mirror. The difference in the on-diagonal (XX and YY) amplitudes across the pupil indicates that this system will have pupil position dependent diattenuation, while the difference in the on-diagonal (XX and YY) phases indicates that this system will have pupil position dependent retardance.

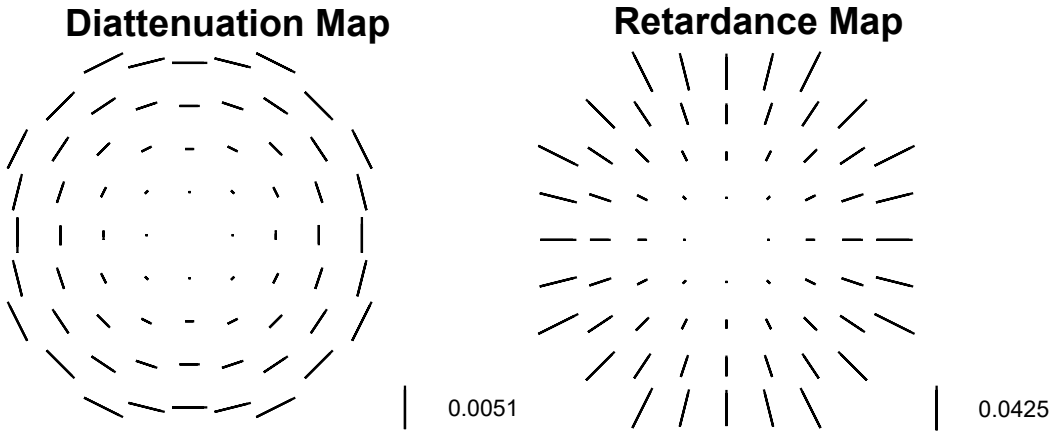
2.2.2.1 Polarization Aberrations

The Jones pupil is a particularly powerful tool for visualizing polarization aberrations because it allows for quick comparison of the Jones matrix elements across the pupil. Polarization aberrations are deviations from ideal behavior, and the ideal wavefront is transmitted with uniform polarization state for arbitrary inputs which is represented by a Jones matrix equal to the identity matrix. Thus, polarization aberrations are represented by deviations of a ray's Jones matrix from an identity matrix. The polarization aberrations of a system can be divided into diattenuation aberration, polarization dependent transmission, and retardance aberration, polarization dependent optical path (phase) difference.

Diattenuation aberration describes how different rays preferentially transmit one polarization state over another. For this parabolic mirror, the differences in the amplitude of the on-diagonal elements of the Jones pupil are indicative

of the diattenuation. Similarly, retardance aberration is present in a system that imparts a different amount of phase on one polarization state relative to another. The differences in the on-diagonal phases maps are indicative of the retardance in this system.

Another helpful visualization technique to highlight the diattenuation and retardance of a surface is the polarization aberration map. Polarization aberration maps are two dimensional vector maps that represent the pupil position dependent diattenuation or retardance of a particular surface. The length of the line in the map is indicative of the magnitude of the diattenuation or retardance while the direction of the line shows the axis of maximum transmission for diattenuation and the fast axis for retardance. The scale bars in the lower right of each plot show the maximum value in the vector map.



(a) (b)

Figure 2.6: Polarization vector maps for a simple parabolic mirror. The diattenuation map (a) shows that the diattenuation magnitude increases quadratically from the center of the pupil outward and the orientation of the maximum transmitting polarization is tangentially oriented. The retardance map (b) shows that, like the diattenuation, the retardance magnitude increases quadratically from the center of the pupil outward and the orientation of the fast axis is radially oriented. The diattenuation and retardance are zero at the center of the pupil because the angle of incidence is zero for the center of the pupil and thus there is no difference in the Fresnel reflection coefficients. It should be noted that the diattenuation and retardance orientations are orthogonal for every point in this pupil.

2.2.2.2 Sources of Polarization Aberrations

As light propagates through an optical system, it will interact with optical surfaces and polarizing elements throughout the system which alter its amplitude, phase, and polarization orientation. When the polarizing element alters the amplitude or phase of the light based on the polarization state, that element will contribute to the polarization aberrations of the system. For example, when reflecting off of a metal mirror, the Fresnel reflection coefficients are different for s- and p-polarized light[41]. The Jones matrix for such an interaction takes on the form of the upper left two by two shown in equation 2.24. Since r_s and r_p are only the same at normal incidence, the Jones matrix will not be an identity matrix for this interaction at non-normal incidences, leading to diattenuation and retardance aberration. Another source of polarization aberrations is form birefringence in thin film coatings on optical elements [42]. Form birefringence is refractive anisotropy due to the micro-structure formed in the process of depositing metal thin films [43]. The structure of the thin film as it is evaporated onto a substrate can become amorphous in some areas and columnar in others, leading to birefringence. The effects of form birefringence can be seen in section 4.2.4.1.

2.2.3 Amplitude Response Matrix and Point Spread Function

The amplitude response function of an image forming system using scalar image formation theory is the Fourier transform of the exit pupil function. The square magnitude of the amplitude response function yields the point spread function (**PSF**)[44]. In order to express the effects of polarization aberrations on image

formation, we use the Jones pupil instead of the scalar exit pupil function. The Fourier transform of each component of the Jones pupil yields the amplitude response matrix (**ARM**)[45].

$$\mathbf{ARM} = \begin{pmatrix} \mathscr{F}\{J_{xx}\} & \mathscr{F}\{J_{xy}\} \\ \mathscr{F}\{J_{yx}\} & \mathscr{F}\{J_{yy}\} \end{pmatrix} \tag{2.26}$$

Here, $\mathscr{F}$ is the spatial Fourier transform of the pupil. For a plane wave incident on the system with Jones vector **E**, the **ARM** details the electric field distribution (amplitude and phase) at the image plane, given by $\mathbf{ARM} \cdot \mathbf{E}$. Figure 2.7 shows the magnitude of the amplitude response matrix calculated from the Jones pupil in figure 2.5. Since the on-diagonals are very similar to a unit circular aperture (only small amplitude and phase variations across them), the on-diagonals of the **ARM** are close do a diffraction limited circular aperture's amplitude response. The off-diagonals of the **ARM** are 2-3 orders of magnitude smaller than the on-diagonal elements and exhibit an even splitting into four parts. This splitting is due to the structure in the off-diagonals of the Jones pupil.

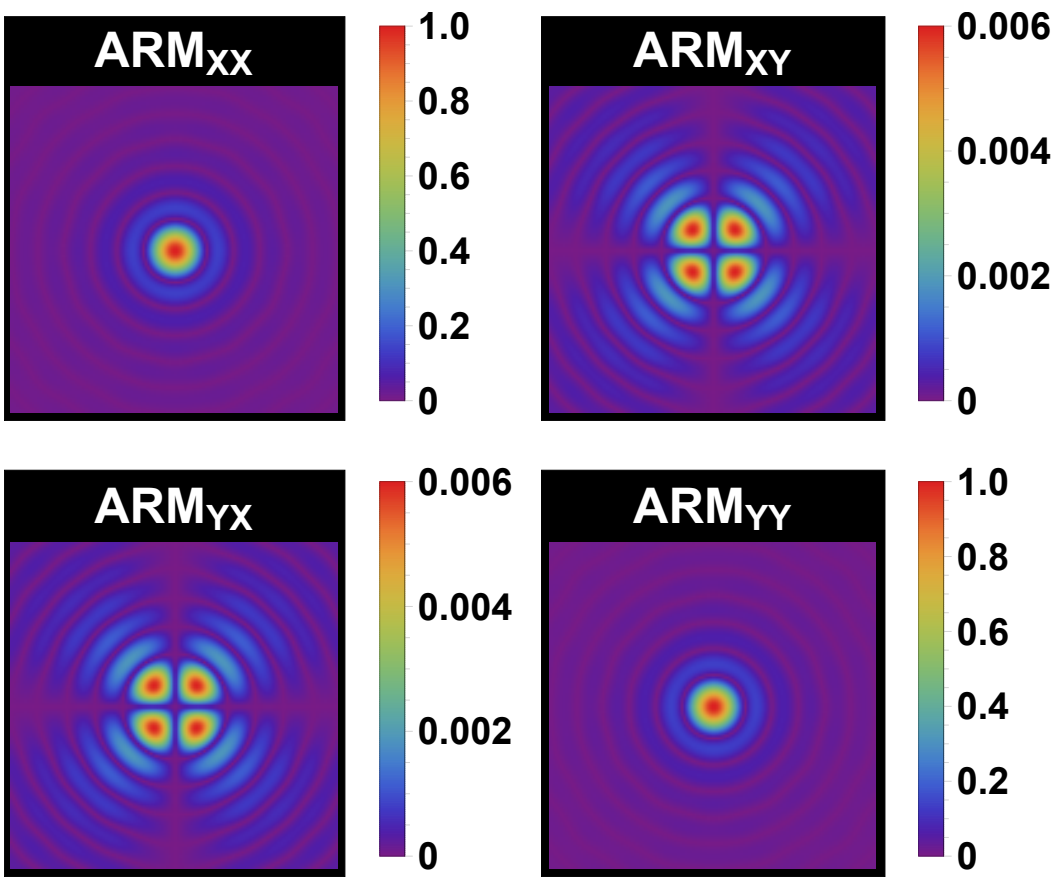


Figure 2.7: Magnitude of the amplitude response matrix of parabolic mirror surface, normalized by the maximum of the XX component. The on-diagonals are similar to a diffraction limited circular pupil's amplitude response. The off-diagonals have structure due to the shape of the off-diagonals of the Jones pupil.

For unpolarized input illumination, the incident X and Y polarized light are incoherent with respect to one another. This means that the field distribution represented by each component of the amplitude response matrix will add incoherently in intensity. Thus the PSF for unpolarized light has four components[18]:

$$\mathbf{PSF} = |\mathbf{ARM}_{xx}|^2 + |\mathbf{ARM}_{xy}|^2 + |\mathbf{ARM}_{yx}|^2 + |\mathbf{ARM}_{yy}|^2 \tag{2.27}$$

Figure 2.8 shows the PSF of this system, plotted on a log scale for easier visu-

alization. The off-diagonals of the ARM are about three orders of magnitude smaller than the on-diagonals, so their contribution to the system PSF will be six orders of magnitude smaller than the on-diagonals. Due to these extremely small off-diagonals, the PSF of this system is very close to an Airy disk.

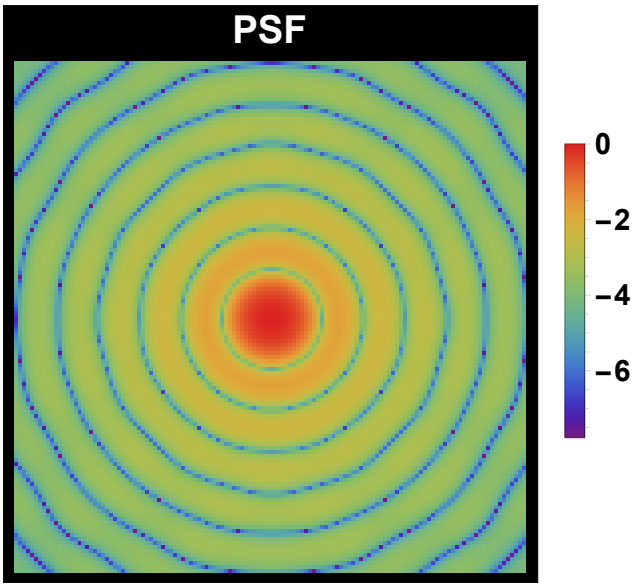


Figure 2.8: PSF plot of a simple parabolic mirror system. This PSF has been normalized and plotted on a log scale for easier visualization. Since the on-diagonals of the ARM are much larger in amplitude than the off-diagonals and there are only small amounts of diattenuation and retardance, the PSF looks like an Airy disk.

2.3 Polarization Ray Tracing With Polaris-M

To conclude this section, the steps for a typical Polaris-M ray trace and analysis are outlined. A step by step outline can be seen in table 2.1. More in-depth information about Polaris-M and polarization ray tracing calculus can be found in [19, 39, 46].

The first step to polarization ray tracing an optical system is to define the surfaces of the system. Each surface in Polaris-M has eleven properties, including a vertex, surface normal, aperture size, and coating specification. Unlike many

other commercial ray tracing programs, Polaris-M utilizes global Cartesian coordinates and defines surfaces vertices and surface normals in that global context. There is no need for dummy surfaces to account for tilts and decenters. Once the optical system has been defined, a polarization ray trace needs a ray or set of rays to trace through the system. Each ray has 29 different parameters, some of which include ray position, propagation vector, wavelength, optical path length, and PRT matrix. Rays traced through a system in Polaris-M have each of their 29 parameters updated after every surface interaction, and new ray segments are added to that ray set, making the ray data at any optical surface extremely easy to extract.

Polaris-M's ray tracing algorithm utilizes well-established procedures to geometrically trace each ray to determine the physical path that ray will take[47], calculating the propagation vector, position, and optical path length at each step. At each interface, Polaris-M determines the ray's angle of incidence at the surface and calculates the Fresnel reflection or transmission coefficients given the surface's material and/or thin film specification for that ray's angle of incidence and wavelength. Then, using equations 2.23 and 2.24, these Fresnel coefficients are converted into a PRT for that ray. The ray trace continues until either the ray reaches an absorbing surface or fails to find a surface interaction and is terminated. If multiple rays are being traced, a single ray is traced until it is terminated or absorbed, and then the next ray in the ray set is traced until all of the rays have been traced.

Polarization analysis in Polaris-M is easily conducted through a suite of functions that calculate everything discussed in this chapter: the Jones pupil, the

amplitude response matrix and PSF, polarization vector maps, and much more. To create smoothly varying pupil maps, an appropriate number of rays should be traced through the system. In the example shown in figure 2.5, a square grid of 51x51 rays was traced. To create an even smoother plot (at the cost of computational time), the ray grid can be increased.

1. Define surfaces for each element in the optical system
2. Define an input ray or set of rays to trace
3. Trace rays through the system
4. Create a Jones pupil from a dense grid of rays traced through the system
5. Create a diattenuation and retardance vector map
6. Create the amplitude response matrix from the Jones pupil
7. Create the point spread function

Table 2.1: Steps for polarization ray tracing and analyzing the polarization properties of a system

CHAPTER 3

Controlling Light With a Coronagraph

3.1 Introduction

There are two main challenges to direct imaging of exoplanets: (1) spatially resolving the off-axis planet PSF from the on-axis star PSF and (2) suppressing the starlight enough that the much dimmer planet light is able to be detected. The first challenge, resolution, is dictated by the size of the telescope's primary mirror and the wavelength of the light. It is also constrained by the ability of the telescope to correct for common path phase errors with adaptive optics (A/O). Adaptive optics will be addressed in a later section, though not in depth. The second challenge, starlight suppression, is addressed by the use of a coronagraph. This chapter will give a brief overview of image formation in telescopes and how resolution is determined, introduce coronagraphy, and discuss some of the computational considerations of calculating the effects of a coronagraph on a PSF. We will end the chapter with an example of applying a specific type of coronagraph, a vector vortex, to a simple imaging system. The code to create the figures for this chapter is provided in appendix F.

3.2 Image Formation Simulation Using Fourier Optics

In scalar imaging theory, the imaging elements of an optical system can be lumped into a "black box", and the properties of the system can be described by specifying how that black box transfers light from the entrance pupil of the system to the exit pupil of the system, the amplitude transfer function [44]. For a perfect imaging system, the amplitude transfer function is a scaled version of the pupil function, $P(x, y)$, a function that details the shape of the exit pupil. In general, wavefront errors (aberrations) may be present in an optical system, so the generalize pupil function takes on the form

$$\mathscr{P}(x, y) = P(x, y)e^{ikW(x,y)} \tag{3.1}$$

where $\mathscr{P}$ is the generalized pupil function, k = $2\pi/\lambda$, and W(x,y) is a path-length error. The Fourier transform of the (scaled) generalized pupil function is the amplitude impulse response of the system. The modulus square of the amplitude impulse response is the point spread function (PSF) of the system. In vector imaging theory, the generalized pupil function is replaced with the Jones pupil and the amplitude impulse response is replaced with the amplitude response matrix, detailed in chapter 2.

An ideal imaging system has a Jones pupil that is an identity matrix within the finite extent of the pupil. That is

$$\mathbf{J}(r,\theta) = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & r \leq \frac{D}{2} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & r > \frac{D}{2} \end{cases} \tag{3.2}$$

where D is the diameter of the exit pupil. Using Fraunhofer diffraction to represent the propagation of the wavefront from the exit pupil to the image plane as a scaled Fourier transform of $J(r,\theta)$ [48], the amplitude response matrix for an ideal system can be written as:

$$ARM(r,\theta) = e^{ikz} e^{i\frac{k}{2z}(r^2)} \frac{A}{i\lambda z} \frac{2J_1(\frac{kDr}{2z})}{\frac{kDr}{2z}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.3}$$

$A = \pi\frac{D^2}{4}$, z is the distance from the exit pupil to the image plane, and J_1 is the Bessel function of the first kind, first order. When using ray tracing to calculate the Jones pupil and amplitude response matrix, the continuous cases shown in equations 3.2 and 3.3 are implemented with discrete samples across the pupil/image plane. Figure 3.1 shows the Jones pupil for an ideal imaging system from a ray trace using perfect coatings (*i.e.* the reflection coefficient from each surface is always unity) and figure 3.2 shows the modulus square of the amplitude response matrix of the ideal Jones pupil on a log_{10} scale. The amplitude response matrix is calculated using discrete Fourier transforms. The optical system used to create this Jones pupil and ampltide response matrix is the same parabolic reflecting surface used in chapter 2, but with perfect coatings instead of aluminum coatings.

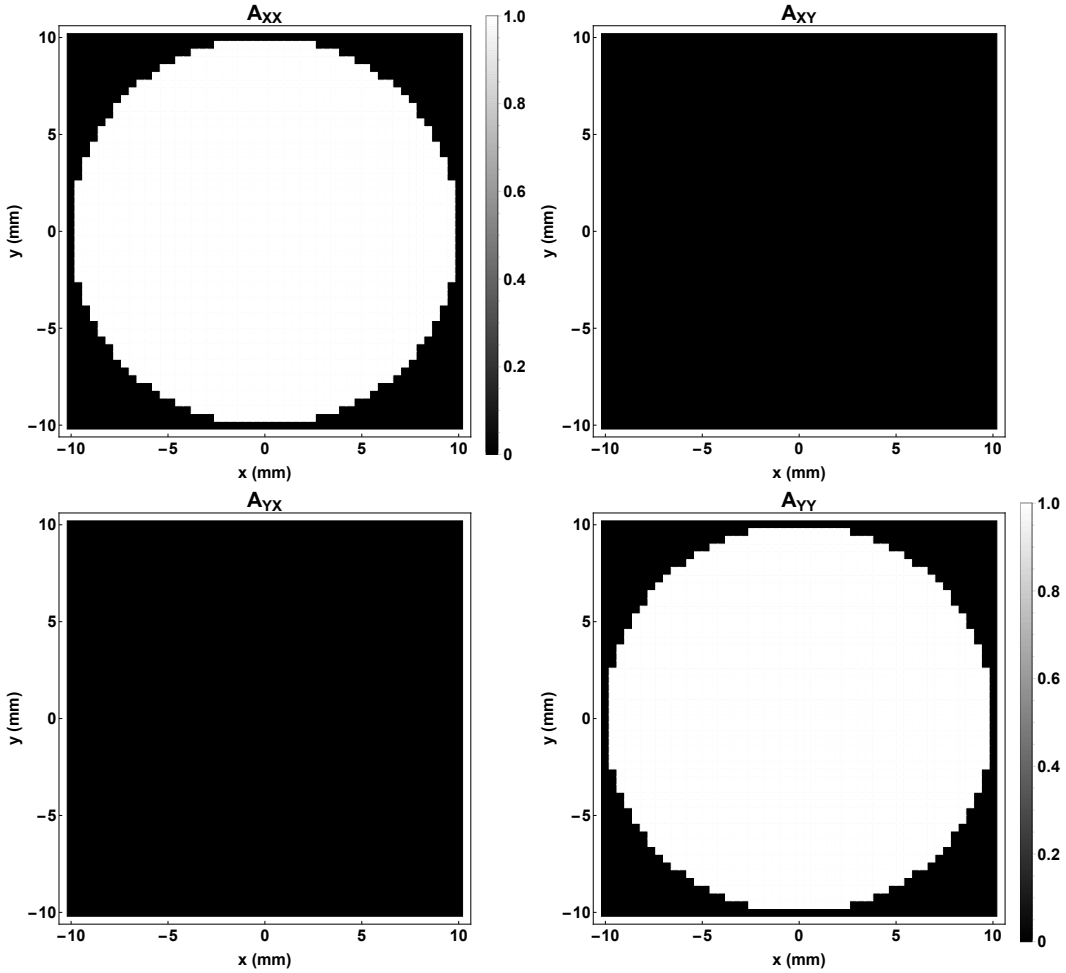


Figure 3.1: Jones pupil for an ideal imaging system. The size of the pupil is 20 mm across with 51 samples in each direction. The spacing between samples is 0.4 mm.

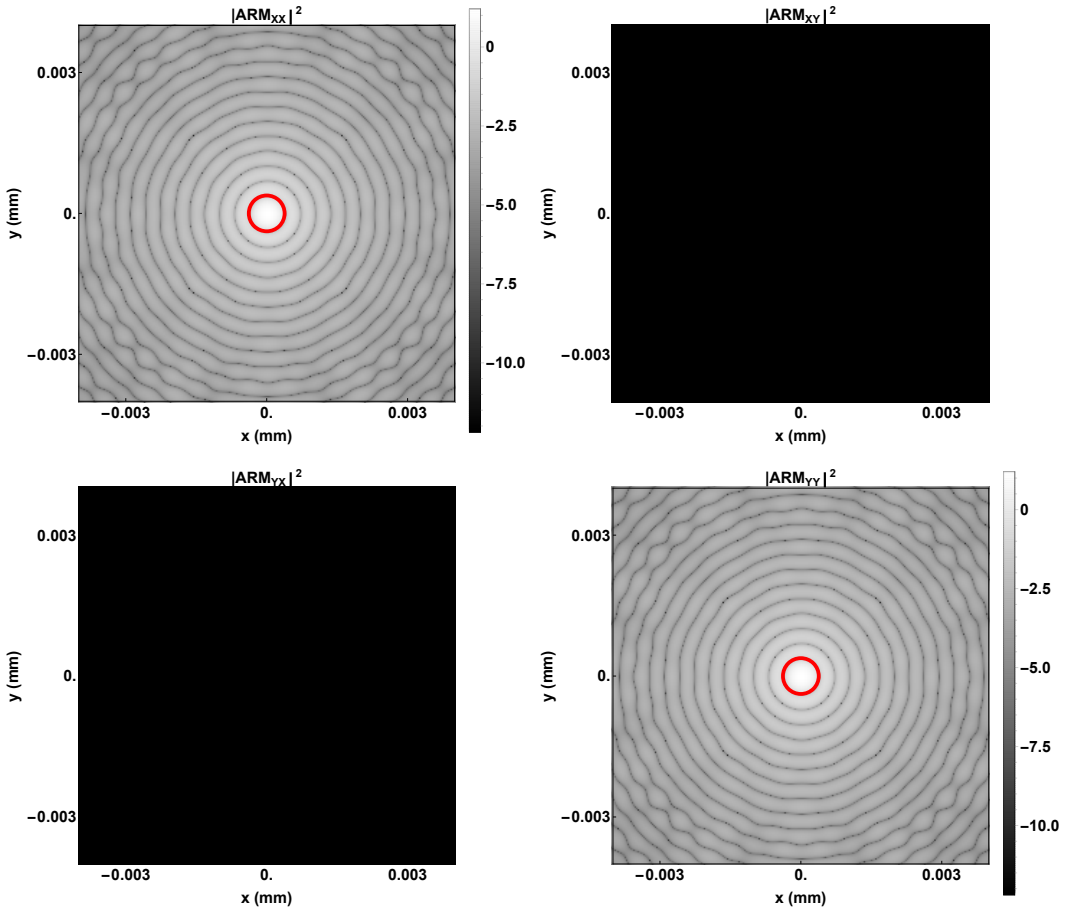


Figure 3.2: Zoomed in plot of the square magnitude of the amplitude response matrix. The red circle in the XX and YY elements represents the radial position corresponding to $1.22\lambda z/D$ in the image plane.

The amplitude response matrix here was calculated by first zero-padding the Jones pupil to ten times its original size, from 51x51 samples across the pupil to 511x511 samples across the pupil (the zero-padding is odd numbered to ensure we sample the chief ray). The spacing between samples in the ARM is given by:

$$\Delta s = \frac{\lambda * L}{n * \Delta x} \tag{3.4}$$

L is the distance from the exit pupil to the image plane, n is the number of samples across the Jones pupil (including zero-padding), and Δx is the spacing

of the Jones pupil. For this example, the wavelength is 500 nm, L is equal to the focal length, 12.5 mm, n is 511, and Δx is 0.4 mm. Thus, the spacing between samples in the ARM is $3 * 10^{-5}$ mm. The central core of the on-diagonal patterns in figure 3.2 has a diameter of 2.44 $\lambda * z/D$ because the Bessel function first goes to zero at a radial position of $1.22\lambda * z/D$. This means that the central core is $7.6 * 10^{-}4$ mm across, allowing about 25 samples across the core for this example. More samples in the ARM can be calculated by increasing the zero-padding of the Jones pupil.

The diameter of the core of the PSF sets the diffraction limited resolution of the telescope. According to the Rayleigh resolution criteria[49], any two object points are considered barely resolved if they are separated by $1.22\lambda z/D$. It is for this reason that the angular resolution of telescopes are generally approximated as simply λ/D.

In this idealized case, the off-diagonals of the ARM are zero because the off-diagonals of the Jones pupil are zero. The on-diagonals take on the form of the Airy pattern, the expected, diffraction limited PSF of a circular pupil. In the non-ideal case, the Jones pupil is no longer an identity matrix like in figure 3.1, but both the phase and amplitude across the pupil will vary due to aberrations like we saw in figure 2.5. The Fourier transform of the aberrated Jones pupil will change the PSF by broadening the central core which reduces the resolution of the telescope. Correcting the wavefront is the subject of adaptive optics[50], wherein deformable mirrors in the optical system are fitted with different shapes to change the intensity distribution in the image plane. However, A/O systems depend on scalar, not vector, imaging theory. The

extent to which A/O can correct an aberrated Jones pupil are discussed later.

3.3 Coronagraphy

The ratio of the flux of a star to a planet orbiting that star is on the order of 10^{10} for earth-like exoplanets[11]. Due to the much larger flux of the parent star, the stellar PSF will overwhelm the exoplanet's PSF. So even with a perfect imaging system, it is necessary to find a way to suppress the starlight in order to detect an exoplanet. Coronagraphy is a means of suppressing bright starlight so as to search for faint companions[51]. The coronagraph this dissertation analyzes consists of a phase mask and Lyot filter that, in conjunction, serve block the core of the PSF of an on-axis point source and suppress the bright diffraction rings, removing light that would otherwise reduce the dynamic range of the image, enabling faint off-axis planets to be observed [52]. The most iconic example of a coronagraph is the Lyot coronagraph.

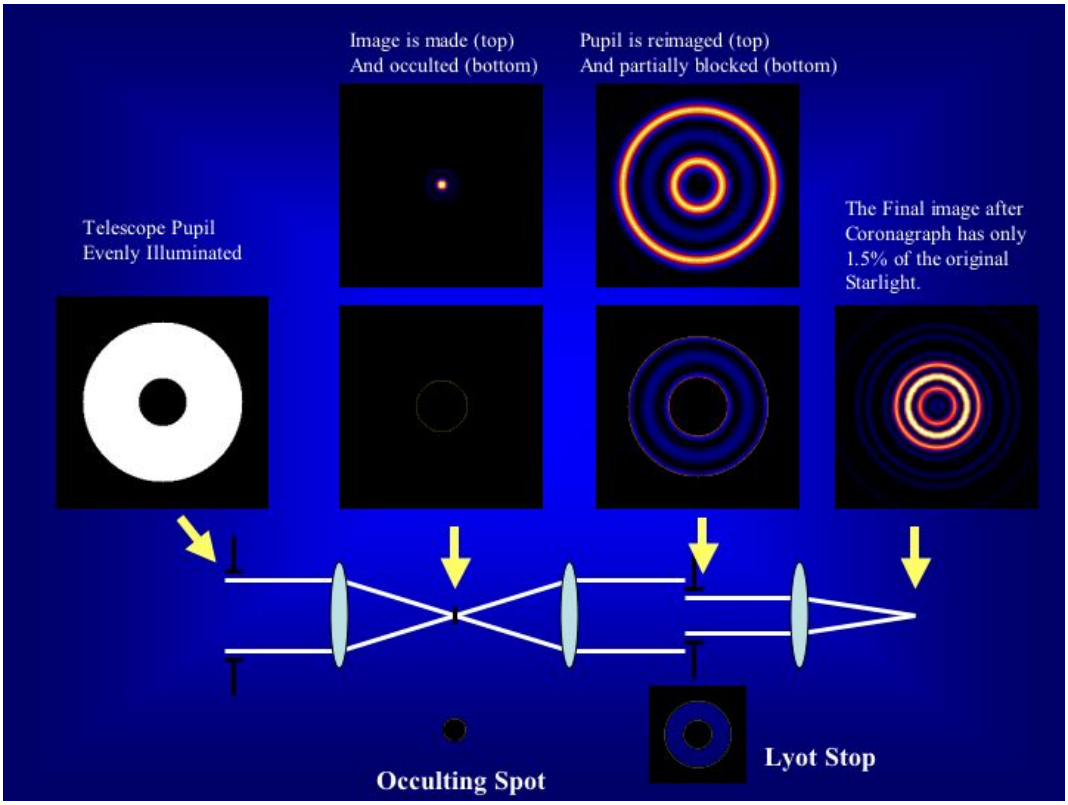


Figure 3.3: Demonstrating how a Lyot coronagraph operates (adapted from Sivaramakrishnan et al. 2001). This telescope is an on-axis Cassegrain with a secondary mirror, so the pupil has a central obscuration.

As shown in figure 3.3, a Lyot coronagraph works by inserting a binary amplitude mask at an intermediate image plane. This occulting mask serves to block the central core of the on-axis star's PSF but allow an off-axis planet's light to propagate through the system. Another mask, the Lyot stop, is used downstream of the original occulting mask to reject the light that is diffracted from the occulting mask. At the final image plane, most of the on-axis star light has been rejected, allowing the dimmer planet light to be detected. Similar to the Lyot coronagraph, focal plane phase mask coronagraphs implement a mask at an intermediate image plane. However, instead of a binary amplitude mask, a phase mask is used instead. The coronagraph that this dissertation is concerned with is the vector vortex coronagraph[53].

3.3.1 Vector Vortex Coronagraph

A vector vortex mask is a space-variant birefringent optical element[51]. The fast axis of a vector vortex rotates at a rate proportional to the angle of any point in the vortex relative to the center. This has a similar effect as a scalar vortex phase mask represented by $e^{il\phi}$, where ϕ is the angle in polar coordinates and l is the topological charge. This charge determines how many 2π rotations occur in the phase as ϕ goes from 0 to 2π.

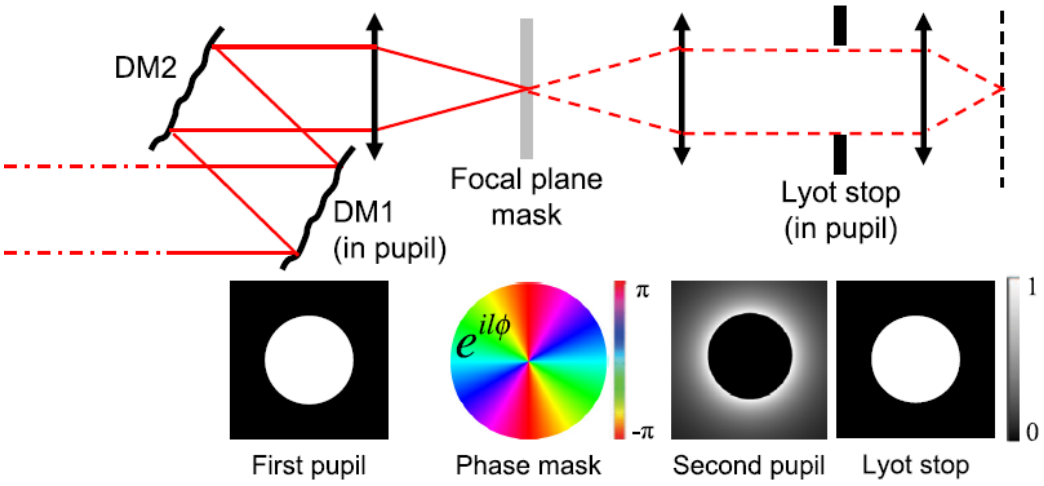


Figure 3.4: Basic example of how the scalar vortex coronagraph operates. Figure is courtesy of Ruane et. al

Figure 3.4, from Ruane et. al [54], shows a basic telescope schematic with a scalar vortex charge 2 (l = 2) mask and Lyot stop. The wavefront at the first pupil is uniform and free of aberrations. The vortex mask is placed at an intermediate image plane where the field will be close to an Airy diffraction pattern. Using Fourier optics, the field at the next pupil plane is a scaled Fourier transform of the product of the electric field distribution at the mask and the phase mask. Due to the high spatial variation of the phase mask near the center of the Airy pattern, where most of the energy is located, the Fourier

transform of the product of the phase mask and Airy pattern causes this energy to be dispersed into a ring of light at the second pupil, leaving a dark central hole. The Lyot stop at this second pupil is sized to be slightly smaller than this ring of light, rejecting anything outside of its spatial extent. The result is that the on-axis starlight is suppressed in the final image plane.

The basic schematic of a telescope with a vector vortex follows the same process as figure 3.4, except the first and second scalar pupils are replaced with Jones pupils and the scalar phase mask is replaced with the Jones matrix of a vector vortex. The Jones matrix of the vector vortex is given by[55]:

$$\mathbf{VV}_m(x,y) = \begin{bmatrix} \cos(m\tan^{-1}(y/x) & \sin(m\tan^{-1}(y/x)) \\ \sin(m\tan^{-1}(y/x) & -\cos(m\tan^{-1}(y/x) \end{bmatrix} \tag{3.5}$$

where m is the charge of the vortex.

In contrast to the on-axis starlight, off-axis planet light will have a shifted and scaled Airy pattern in the intermediate image plane. Because the off-axis Airy pattern does not have its peak at the center of the phase mask, where the highest spatial variations occur, the resulting field for the planet light at the Lyot stop is not diffracted into a ring of light like the on-axis star light. This, in turn, means that the off-axis planet light is *not* rejected by the Lyot stop and propagates to the final image plane.

3.3.2 System Performance Metric

The principal coronagraph performance metric is contrast. There are a few different variations of this contrast. According to Krist [56], "Contrast is defined ... as the ratio of the peak pixel value of the unocculted stellar point spread

function (PSF) to the mean per-pixel surface brightness measured within a specified field around the star." Alternatively, the contrast can be calculated as the planet's irradiance normalized by the star's irradiance[57]. In this dissertation, the contrast will be defined as the PSF of a telescope with a coronagraph mask inserted normalized by the maximum value of the PSF with *no* coronagraph mask inserted.

$$C = \frac{PSF_{masked}(x, y)}{PSF_{unocculted}(0, 0)} \tag{3.6}$$

3.4 Numerical Methods for Modeling Coronagraphs

Calculating the PSF of an optical system requires ray tracing *and* Fourier optics. Ray tracing is used to determine the Jones pupil that transfers light from the entrance pupil of an optical system to its exit pupil. Fourier optics is then used to calculate the PSF by Fourier transforming the Jones pupil and taking its modulus squared. Modeling the effects of a coronagraph on an electric field also requires diffraction algorithms, which cannot be applied in the middle of a ray trace. The novel approach to properly modeling an optical system with a coronagraph mask is to use pupil decomposition to split the optical system into two separate systems, use ray tracing to determine the Jones pupil for each system and Fourier optics to calculate the effect of the coronagraph mask on the electric field at the intermediate image plane and the final PSF of the system.

3.4.1 Pupil Decomposition

Consider an optical system divided into two separate subsystems, system A and system B, where the exit pupil of A is the entrance pupil of B. The coronagraph mask is located at an intermediate image plane between A and B. System A encompasses all the optics up to (but not including) the coronagraph mask, and system B encompasses all the optics after (but not including) the coronagraph mask. The advantage of this pupil decomposition is that the Jones pupil of each subsystem (denoted $\mathbf{JP}_A$ and $\mathbf{JP}_B$) can be computed from ray traces which do not need to be repeated for varying coronagraph mask designs[58]. Without the coronagraph mask inserted into the system, the total Jones pupil is $\mathbf{JP}_B \cdot \mathbf{JP}_A$, the matrix multiplication of the two pupils. The following steps are taken to calculate the effects of the coronagraph mask via pupil decomposition. (*n.b.* these steps use the vector vortex charge 6 mask, but these steps apply when using any focal plane mask)

1. Calculate the electric field distribution at the mask by taking the spatial Fourier transform of $\mathbf{JP}_A$, $\mathscr{F} = \mathcal{F}\{\mathbf{JP}_A\}$.

2. Apply the vector vortex coronagraph mask, $\mathcal{M}$, to the electric field distribution at the intermediate image plane, $\mathscr{M} = \mathscr{F} \odot \mathcal{M}$. $\odot$ is the Hadamard product, which is an element by element multiplication rather than a matrix multiplication.

3. Take the inverse Fourier transform of $\mathscr{M}$, $JP_m = \mathcal{F}^{-1}\{\mathscr{M}\}$, to calculate the necessary pupil distribution that would yield a field distribution equal to $\mathscr{M}$.

4. Take the product of JP_m and JP_B, $JP_{total} = JP_B \cdot JP_m$, to calculate the Jones pupil for the entire optical system including the coronagraph mask.

5. Take the modulus square of the spatial Fourier transform of JP_{total} to determine the PSF of the system with the mask inserted, $PSF_{masked} = |\mathcal{F}\{JP_{total}\}|^2$

These steps serve to encode the effects of the coronagraph focal plane mask into the Jones pupil for the fore-optics of an optical system. These effects are then propagated to the final image plane by combining the Jones pupils of the fore-optics with the Jones pupil of the optics after the coronagraph focal plane mask and using this total system pupil to generate the PSF of the occulted system.

Careful consideration of the sampling scheme is necessary when calculating the Fourier transform of the vector vortex due to the increase in spatial frequency towards the center of the vortex. One method to handle these higher spatial frequency variations, the semi-analytic method, was developed for coronagraph applications [59]. This algorithm applies two discrete Fourier transform (DFT) operations: (1) a DFT that highly samples the central region of the field attenuated by the vector vortex, $\mathscr{M}$, and (2) a DFT that more coarsely samples the entire attenuated field. These two DFTs are then stitched together in the pupil plane via a windowing algorithm to yield JP_m.

3.5 Example Implementation of VV6

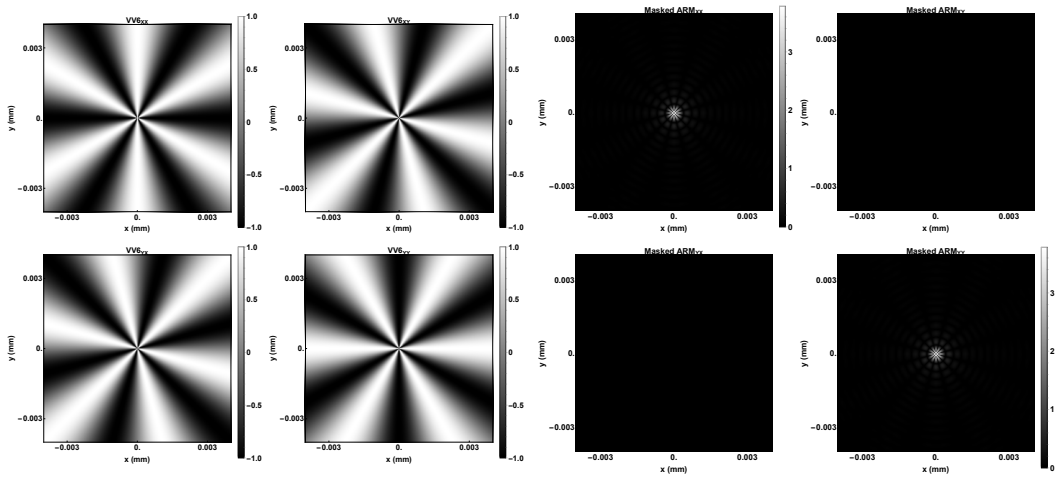
This section goes through an example of simulating a vector vortex coronagraph in a simple telescope system. We will assume ideal pupils here for ease of implementation, but the next chapter will have real pupils calculated from models of actual telescope designs. The basic setup is the same as in figure 3.4. The Jones pupil is an identity matrix inside of the pupil. The vector vortex is applied at an intermediate image plane. An inverse Fourier transform of the masked field via the semi-analytic method yields the masked pupil. Multiplication of the masked Jones pupil with the Jones pupil for the second half of the system gives the total system Jones pupil. The modulus square of the Fourier transform of the system Jones pupil provides the PSF of the system with the vector vortex applied.

3.5.1 Step 1: Calculate the Electric Field at the Intermediate Image Plane

The first step to applying the vector vortex to our perfect imaging system is to calculate the electric field distribution at the intermediate image plane, which were shown in figures 3.1 and 3.2.

3.5.2 Step 2: Apply the Mask

Once the electric field at the intermediate image plane has been calculated, the next step is to apply the vector vortex mask.



(a) (b)

Figure 3.5: The Jones matrix for a vector vortex charge 6 (a) and the product of the vector vortex mask with the amplitude response matrix for an ideal imaging system (b). The pixel spacing is the same for each plot, and is identical to the spacing from figure 3.2

Figure 3.5a shows the Jones matrix for the vector vortex charge 6. Figure 3.5b details the magnitude of the electric field immediately after the vector vortex mask. The off-diagonals are zero while the on-diagonals are similar to Airy patterns that are modulated as the azimuth changes.

3.5.3 Step 3: Calculate Pupil With Mask Encoded Into It

Now that we have the electric field distribution from interaction with the vector vortex mask (figure 3.5b), the next step is to take an inverse Fourier transform of that electric field to get back to the first pupil via the semi-analytic method. This is done by choosing one small, tightly sampled region of the masked electric field near the center and a larger, more sparsely sampled region across the entirety of the masked electric field. Discrete Fourier transforms (DFT) are applied to both regions, then the two DFT'd samples are added together in the pupil plane. The output of this inverse Fourier transform is the necessary Jones

pupil distribution that will create the electric field distribution calculated in step 2. The result of the inverse Fourier transform can be seen in figure 3.6.

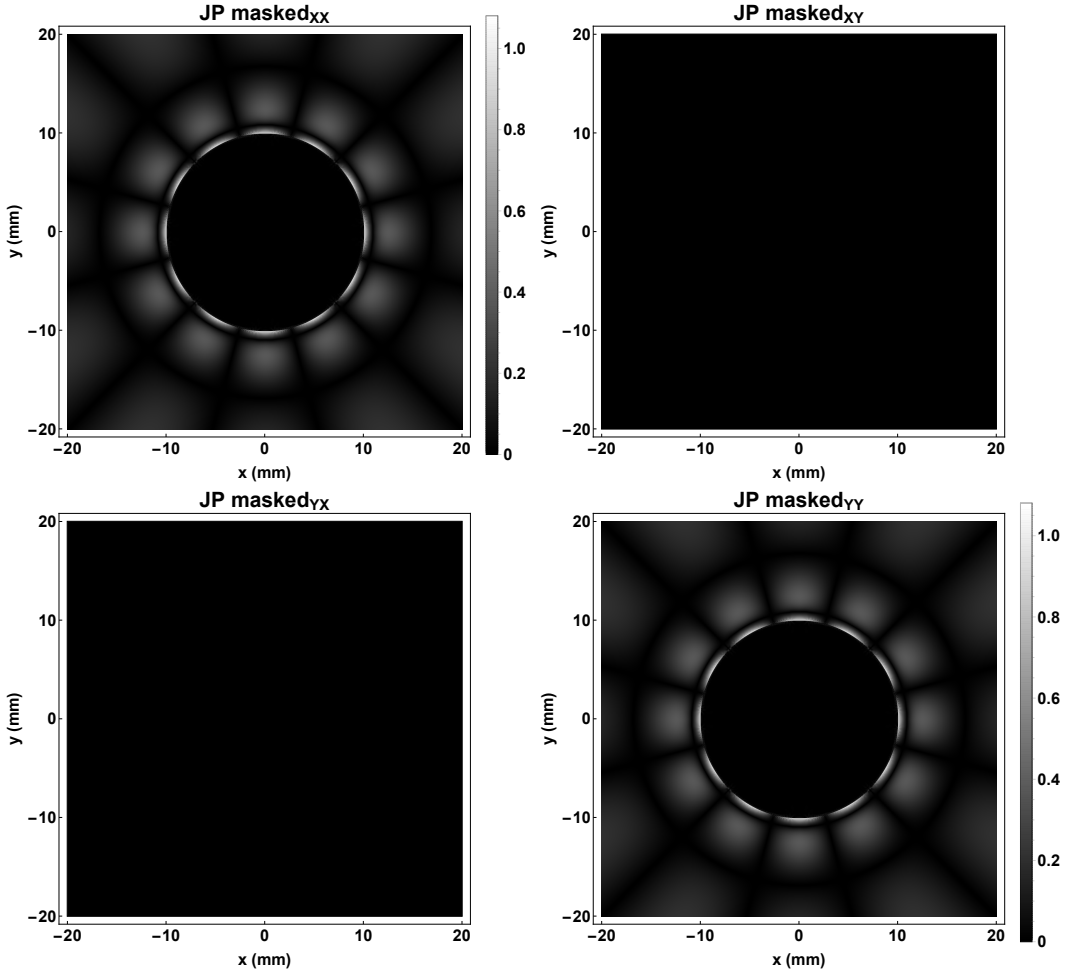


Figure 3.6: The first pupil with the effects of the vector vortex phase mask encoded into it. The effect of the vector vortex is to redistribute the light at the core of the intermediate image plane into a ring of light at the pupil. The original pupil is 20 mm across, but this pupil shows a 40 mm section in the pupil plane to illustrate how the light is redistributed due to the vector vortex. The diameter of the ring of light is the same size as the original pupil, 20 mm across.

The scale provided in figure 3.6 shows that inside the original pupil there is almost no amplitude. Instead, the amplitude has been distributed into a ring (the so-called ring of fire) that is the same in diameter as the original pupil.

3.5.4 Step 4: Calculate Entire System Jones Pupil

Utilizing the reverse of pupil decomposition, the next step is to calculate the Jones pupil for the entire optical system from its constituent Jones pupils. This is simply the multiplication of the two Jones pupils for the system, the updated masked Jones pupil and the Jones pupil for the second half of the system. For our perfect imaging system, the Jones pupil for the second half of the system will be similar to the first pupil: unit amplitude and no phase variation across a circular aperture. However, the second pupil also has the effects of the Lyot stop, causing the diameter of the second pupil to be smaller than the first pupil. In this case, it has been setup so that the Lyot stop has a diameter that is 0.9 the diameter of the first pupil.

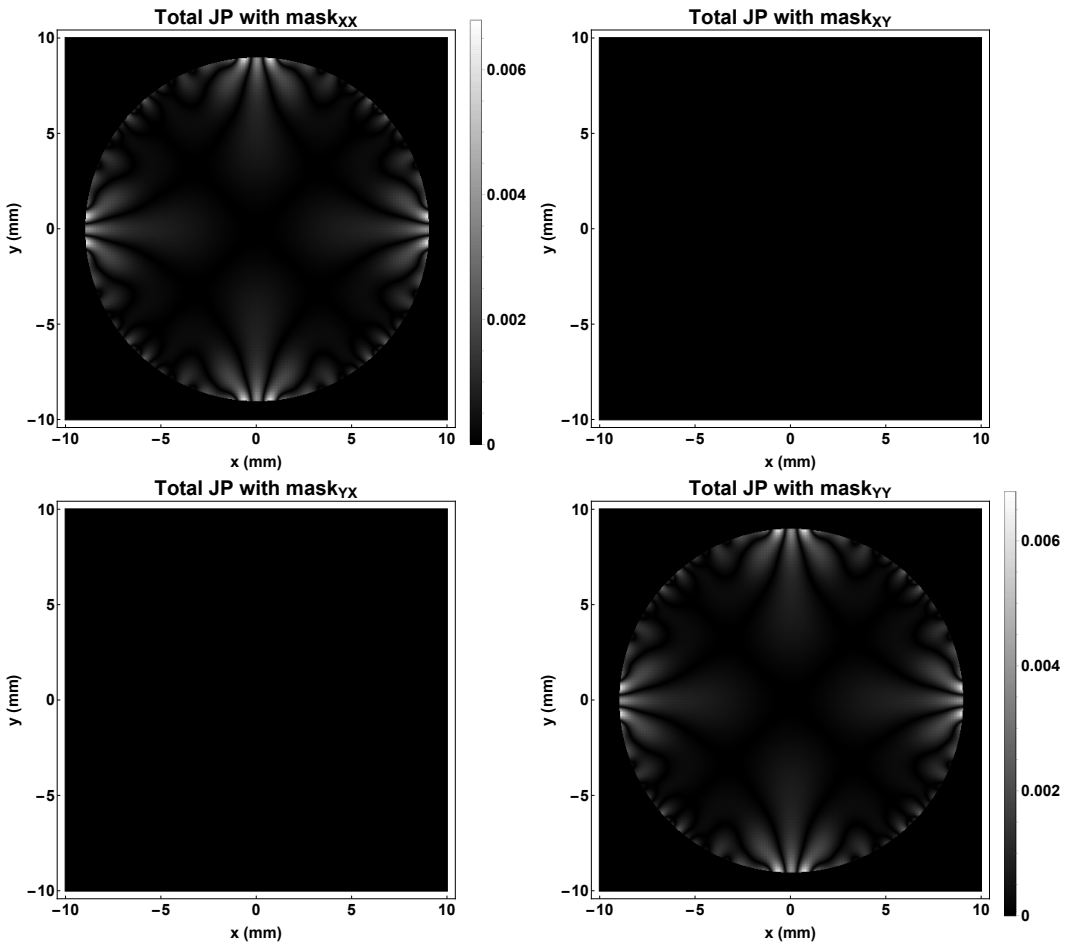


Figure 3.7: Jones pupil for the entire system with the effects of the vector vortex mask.

It is interesting to note that, while it looks like the amplitude in the dark center is zero in figure 3.6, the residual shown in figure 3.7 still has some non-zero amplitude. In the perfect case where the Airy pattern and vector vortex mask at the intermediate image plane are both infinite in extent and continuous, there would be no residual leftover in the dark hole. However, a real mask cannot be infinite in extent and the discretization of the mask and pupil lead to numerical artifacts. This leads to a small amount of on-axis light getting through the system even with the vector vortex mask applied.

3.5.5 Step 5: Determine PSF With Mask Applied

The last step is to calculate the PSF of the masked system at the final image plane. Figure 3.8 shows the four components of the ARM that have been modulus squared, on a logarithmic scale. Each component has been normalized by the maximum of the $|ARM|^2$ of the system with no mask installed, so it follows our definition of contrast.

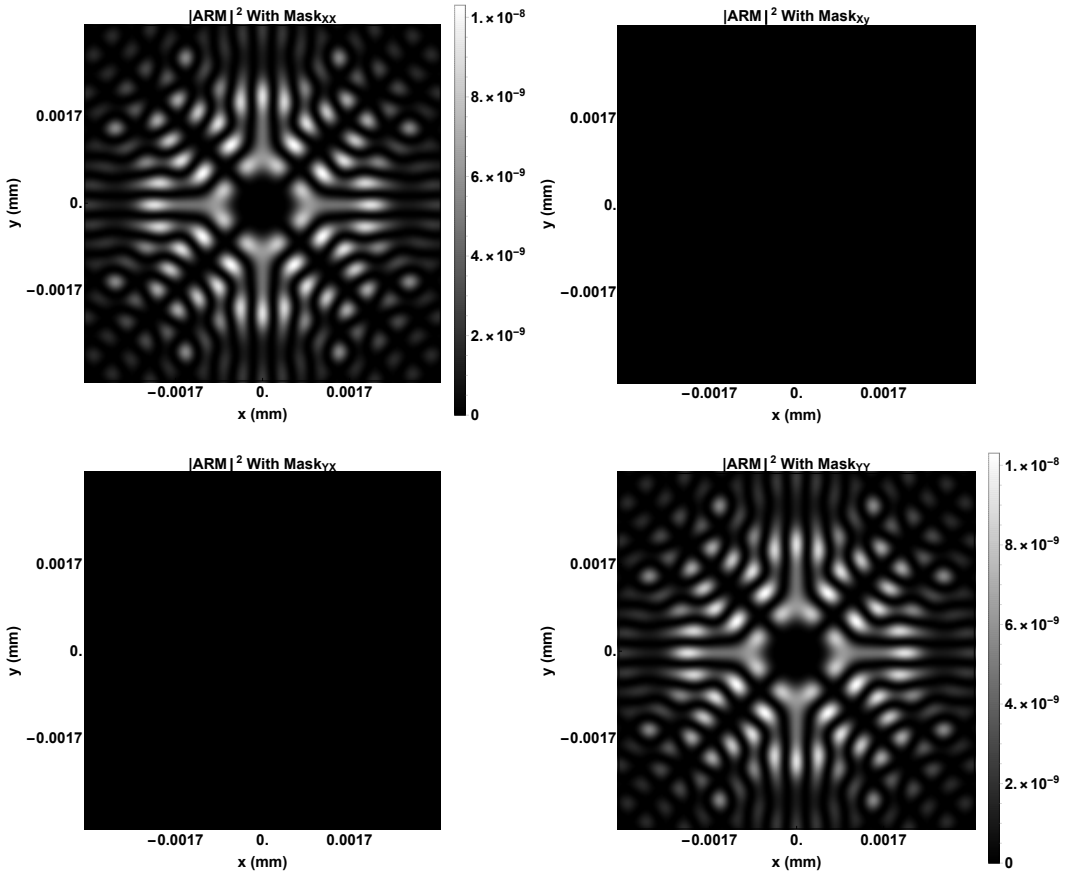
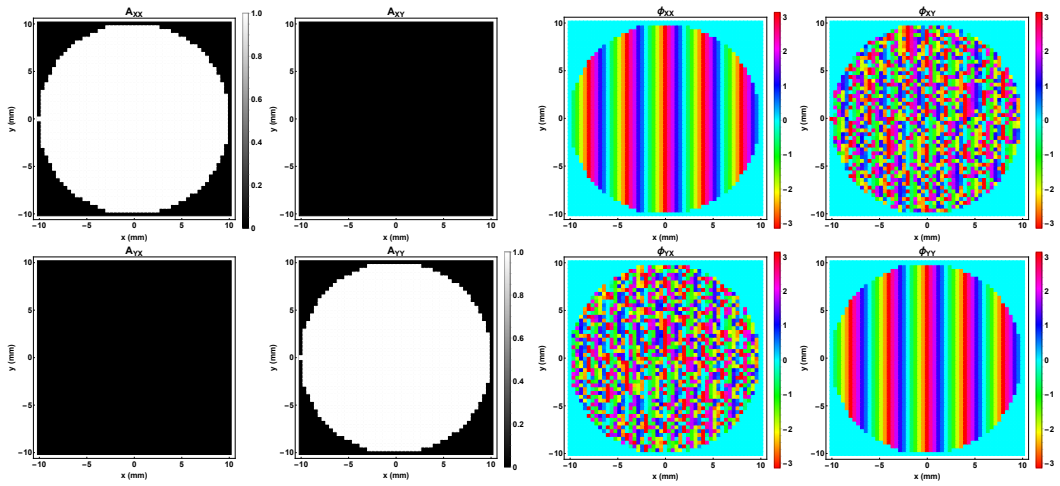


Figure 3.8: Components of the $|ARM|^2$ of a perfect imaging system with a vector vortex charge 6 mask applied, plotted on a log_{10} scale. This plots have been normalized by the maximum value of the $|ARM|^2$ of the system with no coronagraph.

The total PSF of this system for unpolarized input light is the sum of all four components shown in figure 3.8. However, since the off-diagonals are zero, it's simply the sum of the on-diagonal elements.

3.5.6 Effects on an Off-Axis Planet

Steps 1-5 are now repeated for an off-axis planet to see the effects of the vector vortex when the center of the vector vortex mask does not align with the center of the off-axis planet's Airy diffraction pattern. An off-axis planet was simulated by ray tracing the same system as before, but this time the propagation direction of the wavefront was rotated around the y-axis by an angle of 5 λ/D, or 0.000125 radians. The result of this tilt is that the phase of the Jones pupil picks up a linear phase while the amplitude stays the same. The off-diagonals show noisy phase due to the off-diagonals being machine precision zero (with values around $10^{-20} + 10^{-20}i$).



(a) (b)

Figure 3.9: Jones pupil for an off-axis planet. The system is still perfectly reflecting, so there are no amplitude variations across the pupil. The phase of the pupil has a linear phase due to the tilt of the wavefront at the pupil (the phase is wrapped to π here so it is hard to see the linearity).

The amplitude response matrix corresponding to this off-axis planet's Jones pupil is shown in figure 3.10. The tilt angle of the wavefront is $5\lambda/D$, which should correspond to a physical shift of $5\lambda f/D$, or 0.0015625 mm. This is exactly where the peak value is in figure 3.10.

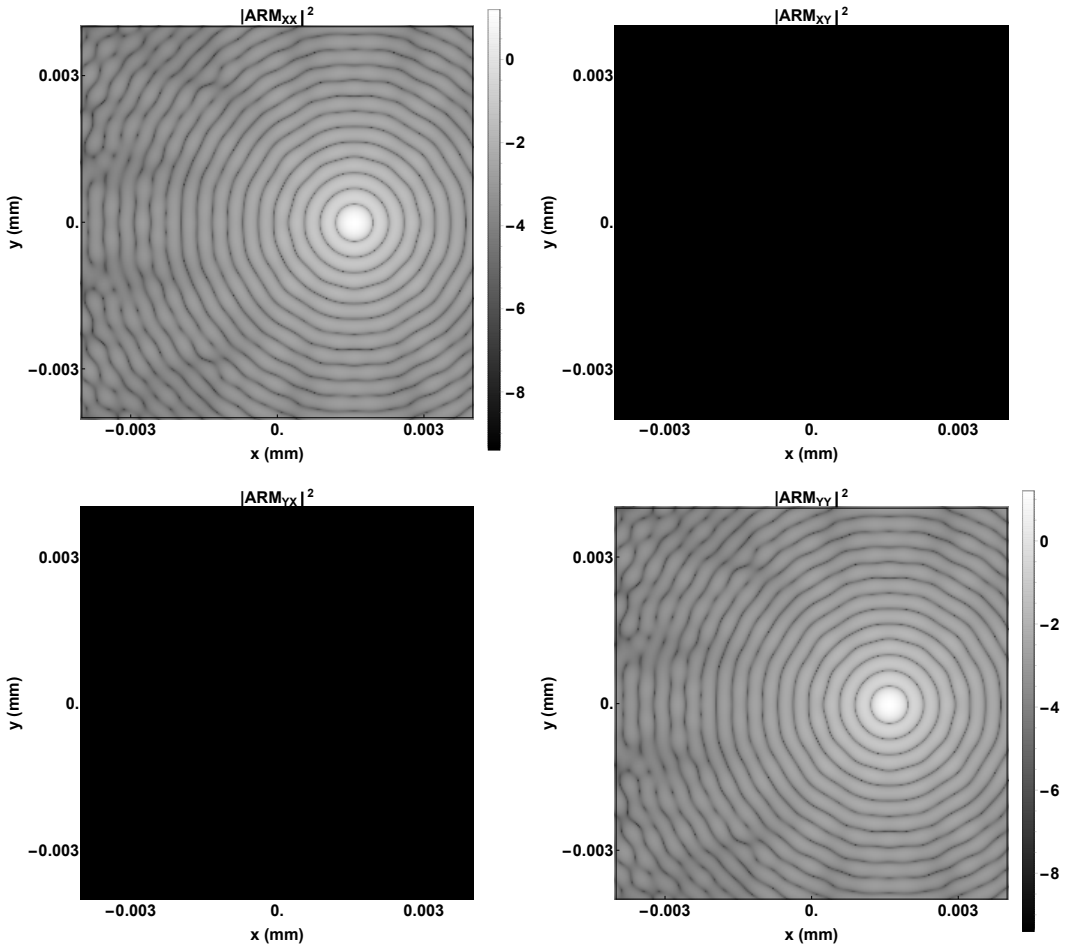
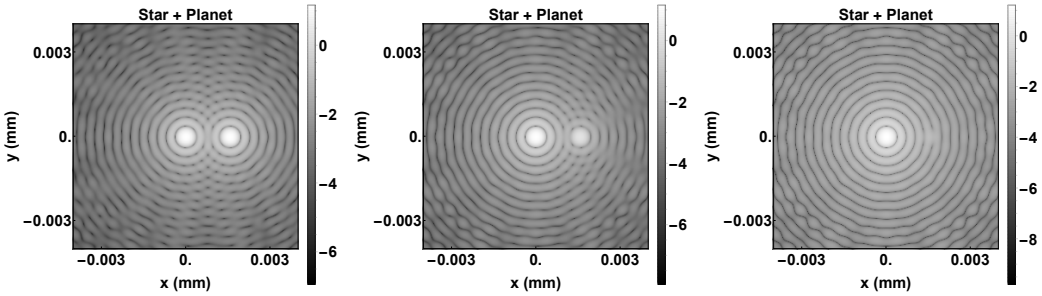


Figure 3.10: Magnitude squared of the amplitude response matrix for the off-axis planet. Compared to figure 3.2, the center of this pattern is shifted along the x-direction.

Comparing the on-axis star's ARM and off-axis planet's ARM, we can explicitly see the shift. Figure 3.11 shows the XX component of the $|ARM|^2$ for the planet and star together for different ratios of the flux of the planet. As the planet gets dimmer, it is harder and harder to discern the planet from the rings of the star's light distribution. The vector vortex operates to block the star's light so the dimmer planet can be seen.



(a) (b) (c)

Figure 3.11: Comparison of the XX component of the $|ARM|^2$ for the on-axis star and off-axis planet for different ratios of the planet to star flux. (a) flux ratio of 1, (b) flux ratio of 0.1 (planet has 10% of star's flux), and (c) flux ratio of 0.001 (planet has 0.1% of star's flux).

Figure 3.12 shows the product of the vector vortex mask with the off-axis planet's ARM. Unlike the on-axis star's case (figure 3.5b), the center, highly spatially varying portion of the vector vortex does not align with the center of amplitude response matrix. This mis-alignment will cause the Jones pupil for the off-axis planet with the vector vortex applied to be closer to the original input Jones pupil than the "ring of fire" we would see from the on-axis starlight.

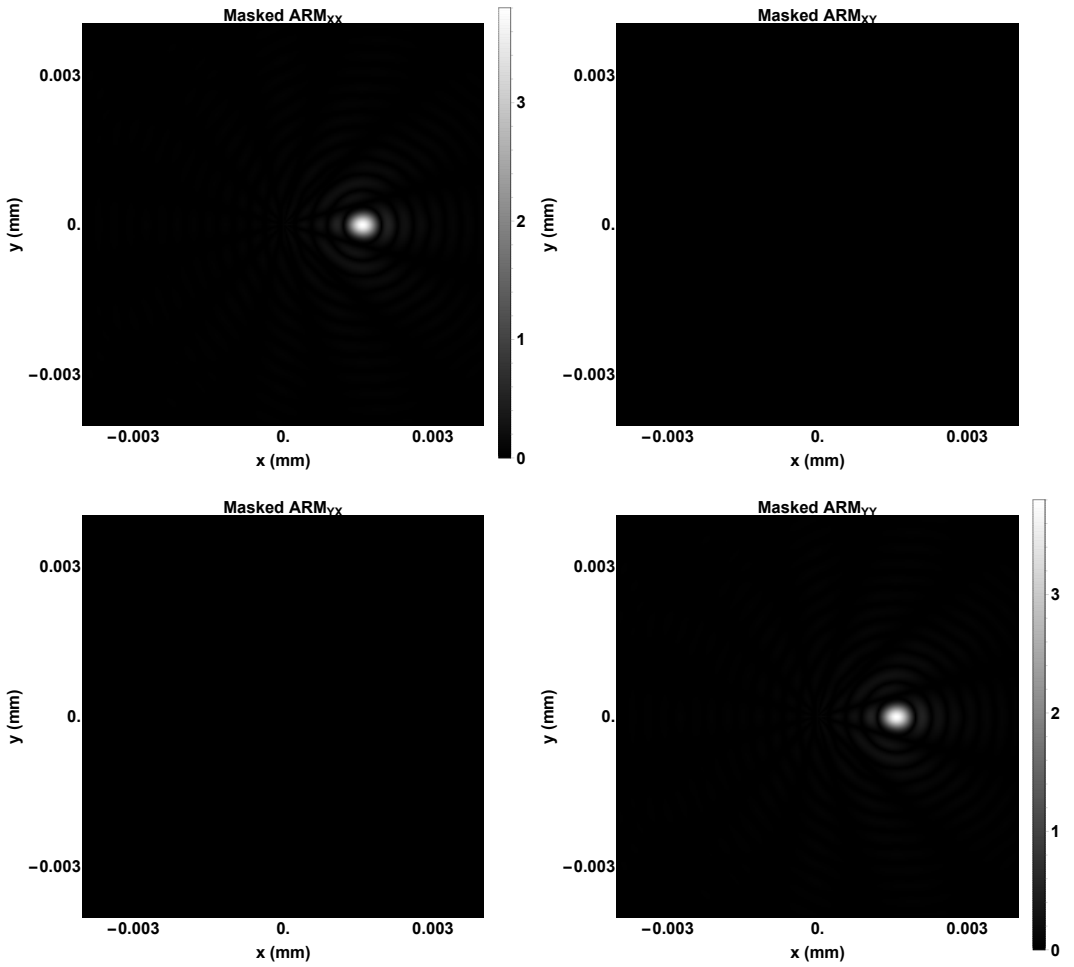
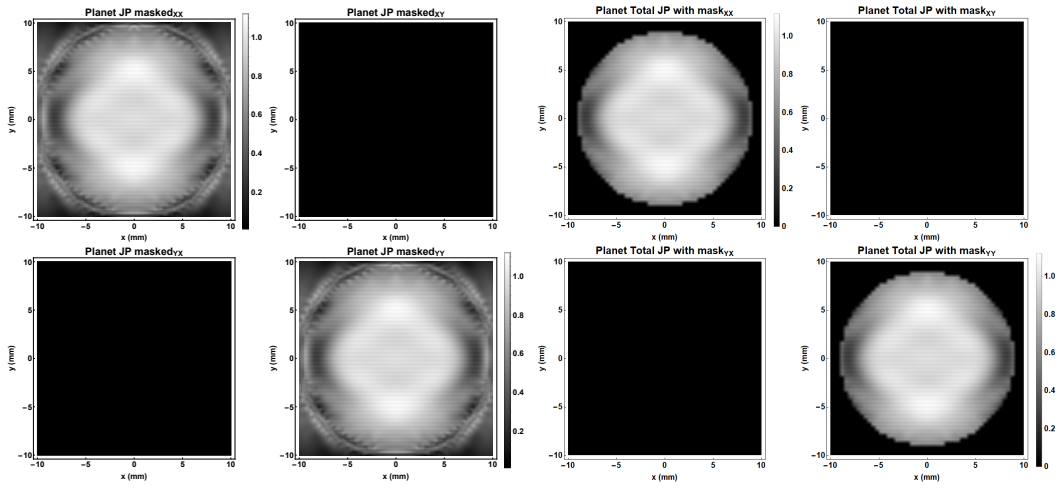


Figure 3.12: Product of the vector vortex charge 6 with the amplitude response matrix for the off-axis planet.

Figure 3.13 shows the Jones pupil with the effects of the vector vortex as well as the Jones pupil for the entire system (incorporating the effects of the Lyot stop). The light in the pupil has not been redistributed into a ring around the pupil, so it will propagate through the Lyot stop and to the final image.



(a) (b)

Figure 3.13: (a) Amplitude of Jones pupil at the first pupil for the off-axis planet. Unlike the on-axis star, the vector vortex has not redistributed the off-axis planet light into a ring, so it will propagate through the Lyot stop and to the final image plane. (b) Amplitude of the Jones pupil for the entire system for the off-axis planet. Unlike the on-axis star case, the Jones pupil for the off-axis planet still has most of its light propagate through the system

The corresponding $|ARM|^2$ plot for the off-axis planet is shown in figure 3.14. The light from the planet has not been attenuated like the on-axis star. The amount of on-axis starlight that makes it through to the final image plane is extremely small. In contrast, a large amount of the original off-axis planet light is able to make it through the vector vortex to the final image plane. To calculate the relative amount of on-axis and off-axis light that makes it through the system, Parseval's theorem is used to calculate the total power from the pupil maps. Parseval's theorem states that[60]:

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |F(\xi)|^2 \, d\xi \tag{3.7}$$

where $f(x)$ and $F(\xi)$ are a Fourier transform pair. The total integrated power at the image plane is given by the integral of the PSF. Since the PSF is simply the modulus square of the ARM and the ARM and Jones pupil are a Fourier

transform pair, Parseval's theorems allows us to calculate the total power by taking the modulus square of the Jones pupil and summing all the elements together. Thus, summing up the modulus square of each point in the pupil in figure 3.1 and comparing it to the sum of the modulus square of every point in the pupil in figure 3.7, the amount of on-axis starlight that arrives at the final image plane after interacting with the vector vortex is calculated to be 10^{-8} of the original input. For the off-axis planet, the sum of the modulus square of every point in the tilted pupil, figure 3.9, is compared to the sum of the modulus square of every point in the masked tilted pupil, figure 3.13. The amount of off-axis planet light that arrives at the final image plane after interacting with the vector vortex is calculated to be 67% of the original input planet light.

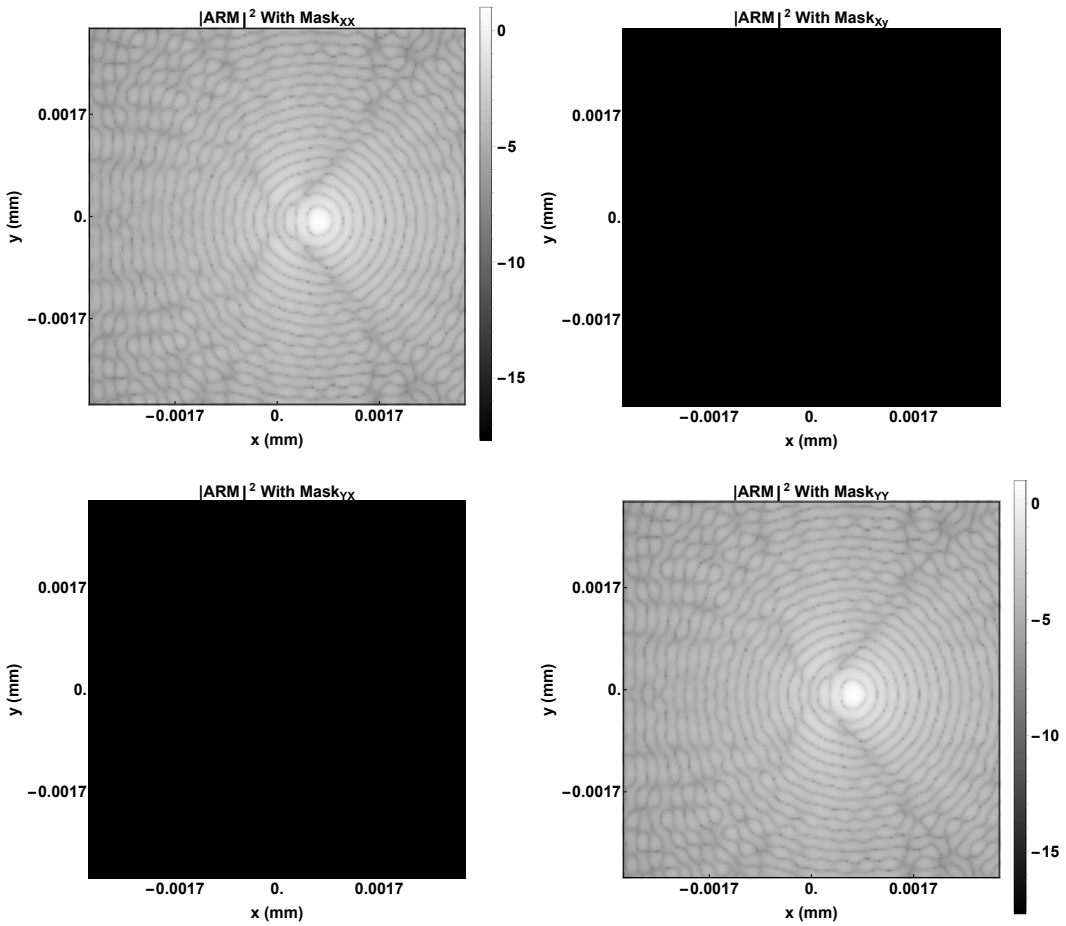
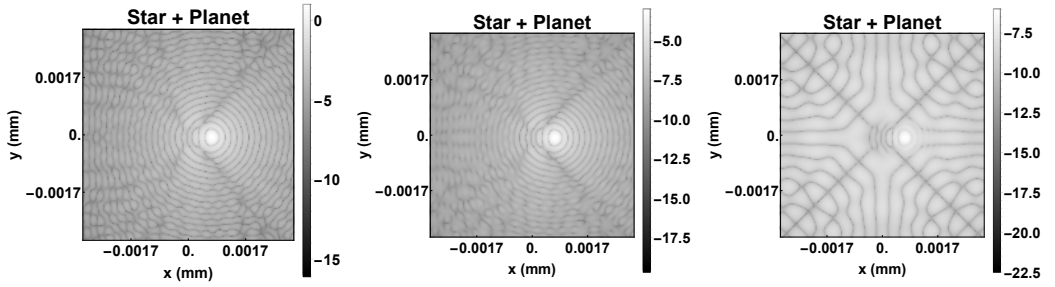


Figure 3.14: $|ARM|^2$ for the off-axis planet with the vector vortex mask on a log_{10} scale, normalized by the maximum of the unocculted, on-axis star's $|ARM|^2$ so the two can be directly compared. Though there are some slight variations from an Airy pattern, the planet light has not been attenuated as much as the on-axis starlight.



(a) (b) (c)

Figure 3.15: Comparison of the PSF at the final image plane for the sum of an on-axis star and an off-axis planet with different ratios of the planet to star flux. (a) flux ratio of 1, (b) flux ratio of 10^{-4}, and (c) flux ratio of 10^{-7}.

Figure 3.15 shows the sum of the on-axis star's PSF with the off-axis planet's

PSF for different ratios of the planet-star flux. The planet light overwhelms the final image plane when its flux is greater than 10^{-4} of the star's flux. When the planet's flux becomes dimmer than 10^{-7} of the star's flux, the residual starlight that the vector vortex could not attenuate becomes visible. The analysis so far has been with perfect imaging. In the next section, the polarization aberrations for a more complex telescope/coronagraph system are calculated and the effect of the polarization aberrations on the contrast are calculated.

CHAPTER 4

Telescope and Coronagraph Simulation

4.1 Introduction

This chapter applies the methods of polarization ray tracing, described in chapter 2, and image formation in coronagraph systems, described in chapter 3, to analyze the polarization effects on contrast in HabEx. HabEx is one of the candidate flagship missions being studied in detail by NASA, the habitable exoplanet observatory [61], HabEx. Pupil decomposition and polarization ray tracing are used to create Jones pupils for the fore-optics (everything before the coronagraph) and post-optics (everything after the coronagraph) and calculate the contrast of HabEx in these configurations: (1) ideal case, (2) the isotropic coating case, and (3) the anisotropic primary coating case. The Polaris-M code used to polarization ray trace HabEx is supplied in appendix D. The Matlab code used to incorporate the effects of the vector vortex coronagraph is supplied in appendix E. A second mission, the large UV, optical, IR surveyor mission[62] (LUVOIR) is also analyzed using polarization ray tracing, and the Jones pupil of its fore-optics is detailed at the end of the chapter. The code to create most of the figures in this chapter is provided in appendix G.

4.2 HabEx

HabEx is a concept for a mission to directly image planetary systems around Sun-like stars. Its primary goal is to directly image Earth-like exoplanets, and characterize their atmospheric content. HabEx will have multiple configurations and access to different instruments including: a "workhorse" camera capable of imaging from the UV to near-IR, a high resolution UV spectrograph, a starshade, and a coronagraph. This dissertation only studies the 450-550 nm channel of the coronagraph which is one half of the "blue" channel, but the same process applied in this chapter is applicable to all wavelengths. Extensive literature about the other instruments is found in references [63, 64, 65, 66, 67].

4.2.1 Optical Prescription

The HabEx coronagraph system contains 15 reflecting surfaces: ten curved mirror surfaces and five flat mirror surfaces. Two of the flat mirror surfaces are deformable mirrors but are modeled as flat mirrors. Figures 4.1 and 4.2, taken from Martin et. al 2017, display the overall telescope system and the coronagraph subsystems. As shown in figure 4.2, the coronagraph mask sits at the intermediate image plane before off-axis parabola (OAP) 4. The fore-optics of the HabEx coronagraph is every optical surface before the coronagraph mask. It is comprised of 11 reflecting surfaces (six curved surfaces and five flat mirrors). The post-optics is every optical surface after the coronagraph mask. It contains four reflecting surfaces, all of which are OAPs. Figure 4.3 shows an alternate view of the post-optics and fore-optics of the "blue" channel.

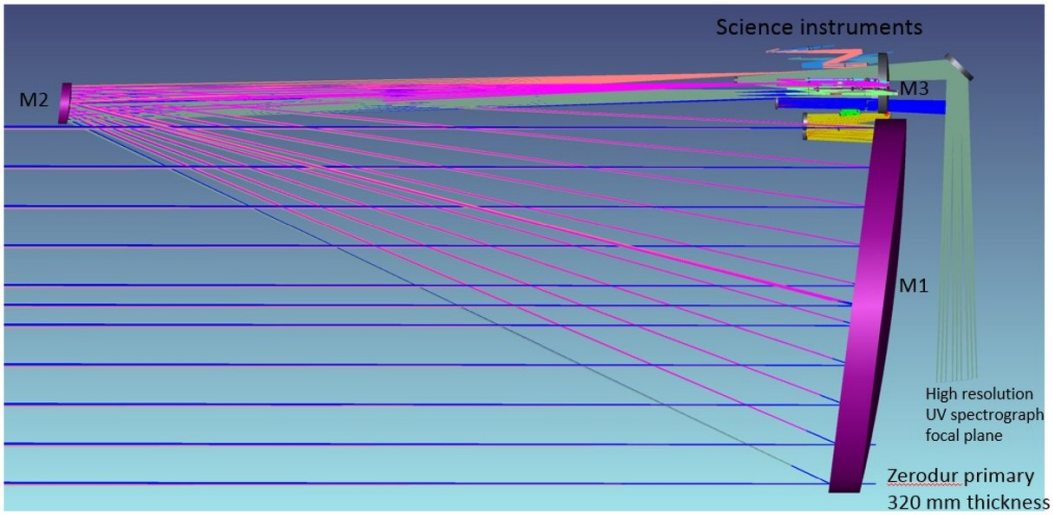


Figure 4.1: Side view of the HabEx telescope optical system

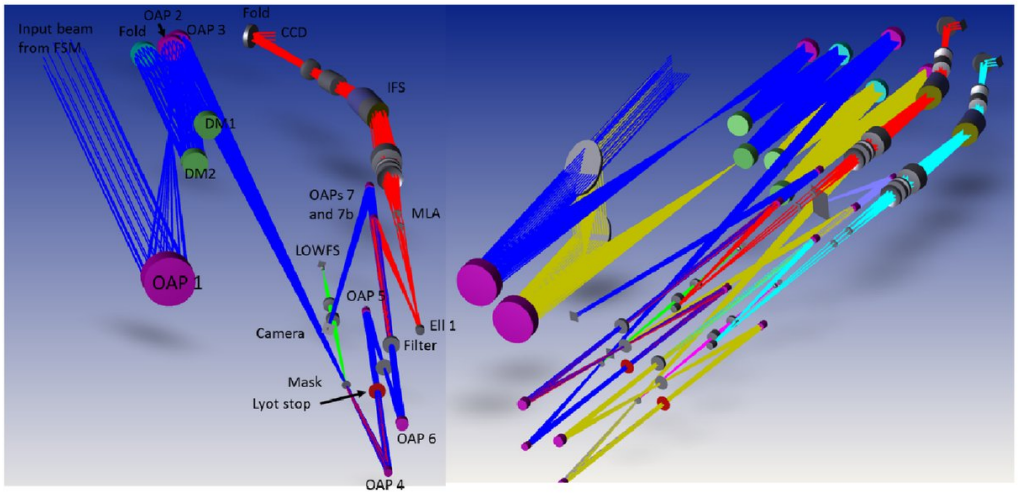


Figure 4.2: Zoomed in view of the HabEx coronagraphs. Left is the "blue" channel, right is both channels together.

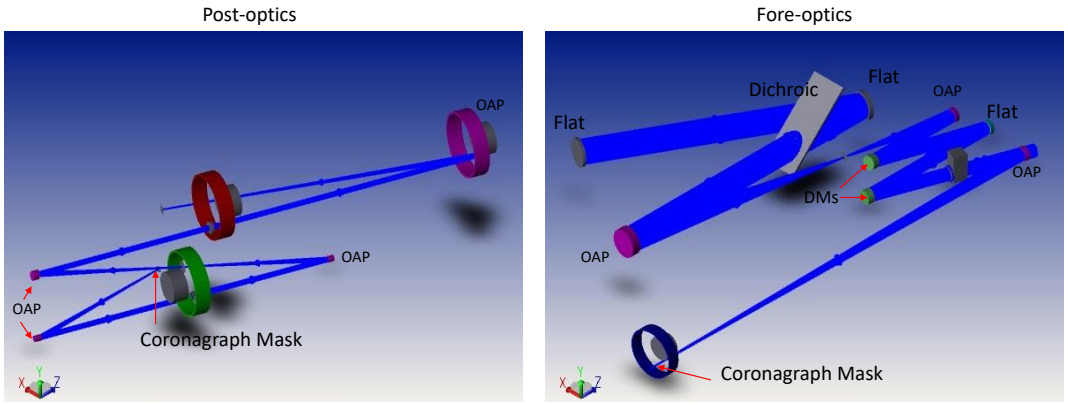
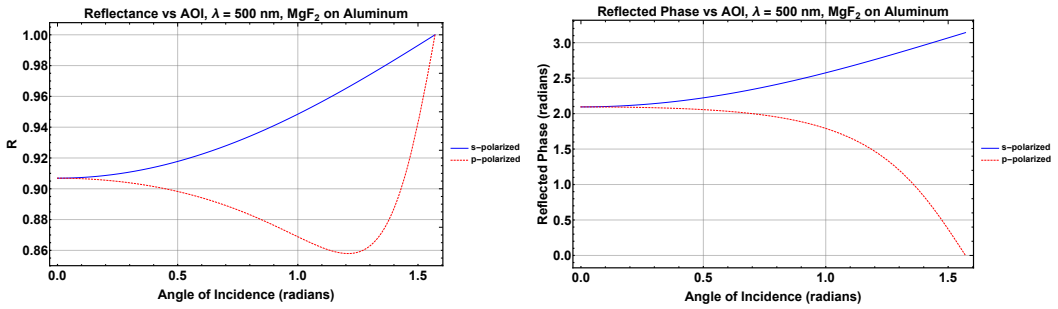


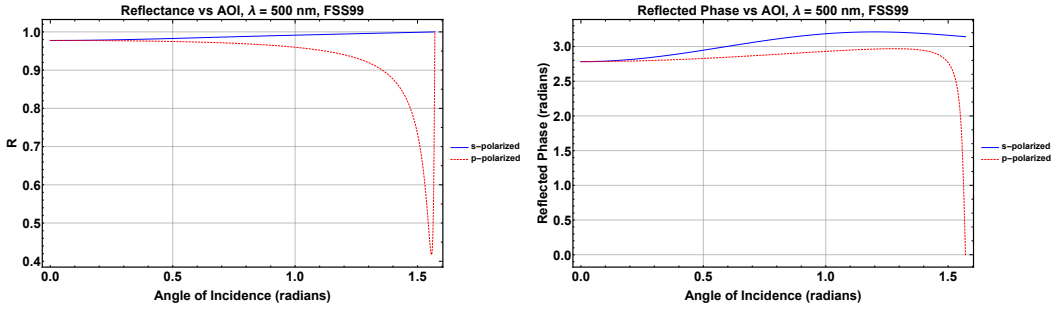
Figure 4.3: Alternate view of the post-optics and fore-optics for the "blue" channel. Light from the tertiary mirror is incident on the flat mirror in the upper left of the "fore-optics" figure.

The first two mirrors in the HabEx coronagraph layout (primary and secondary) are aluminum coated mirrors overcoated with 25 nm of magnesium fluoride. Every subsequent mirror is coated with FSS99-600, a highly reflective multilayer coating on silver. The prescription for FSS99-600 can be found in Appendix C. Figure 4.4 shows the reflectance and phase change for s- and p-polarized light at 500 nm for a coating of 25 nm of MgF2 on aluminum. Figure 4.5 shows the reflectance and phase change for s- and p-polarized light at 500 nm for FSS99.



(a) (b)

Figure 4.4: Thin film performance of 25 nm MgF_2 on aluminum at a wavelength of 500 nm. (a) shows the reflectance of the s- and p-polarized light versus angle of incidence. (b) shows the reflected phase for s- and p-polarized light versus angle of incidence



(a) (b)

Figure 4.5: Thin film performance of FSS99-600 at wavelength 500 nm. (a) the s- and p-reflectance versus angle of incidence. (b) the s- and p-reflected phase

Table 4.1 shows the angles of incidence (AOIs) in radians for the chief ray at

each surface. It also shows the minimum and maximum angles of incidence for rays traced to those surfaces.

	Chief AOI	Max AOI	Min AOI
M1	0.125625	0.209709	0.0442911
M2	0.140142	0.235599	0.0496826
M3	0.096655	0.106247	0.0872098
Folds	0.146103	0.146104	0.146103
FSMs	0.255159	0.25516	0.255159
Dichroic	0.785398	0.785399	0.785398
Dichroic back	0.504705	0.504705	0.504705
M4	0.0996687	0.119605	0.0800525
M5	0.0996687	0.119604	0.080053
DM1	0.122172	0.122173	0.122172
Fold	0.122172	0.122173	0.122172
DM2	0.174533	0.174534	0.174532
M6	0.173246	0.178055	0.168444
Coronagraph mask	$4.55413 * 10^{-6}$	0.0107238	$4.55413 * 10^{-6}$
M7	0.173246	0.178054	0.168447
Lyot stop	$2.4872 * 10^{-6}$	$6.82699 * 10^{-6}$	$2.20517 * 10^{-7}$
M8	0.119425	0.124291	0.11458
Field stop mask	0.238853	0.248581	0.229163
M9	0.119428	0.12429	0.114583
M10	0.0748598	0.0773301	0.0723998

Table 4.1: Chief ray, maximum, and minimum AOIs in radians for each surface in HabEx

4.2.2 HabEx Polarization Ray Trace Outputs: Jones Pupil and ARM

The Jones pupil of optics before the mask will be referred to as JP_A and the Jones pupil of optics after the mask as JP_B. As described in chapter 3, there are five steps to calculating contrast with a vector vortex coronagraph: (1) calculate the field at the coronagraph mask plane via Fourier transform of JP_A, (2) apply the coronagraph mask, (3) inverse Fourier transform the modulated field to get a modulated pupil (called JP_{masked}), (4) form a system end-to-end Jones pupil via multiplication of JP_B and JP_{masked}, and (5) Fourier transform

and modulus square the system Jones pupil to calculate the PSF. The calculation of the Jones pupils JP_A and JP_B is carried out with a polarization ray trace in Polaris-M.

Figure 4.6 shows the Jones pupil for the fore-optics returned from the polarization ray trace. This pupil was created by tracing a sqaure grid of 101x101 rays in Polaris-M. The spacing between pixels in the Jones pupil is 3.8 mm. The left side shows the amplitude of the Jones pupil and the right side shows the phase in radians of the Jones pupil. The amplitude of the pupil is relatively close to an identity matrix, scaled by a factor of about 0.82. This difference from unity is caused by absorption of the light at each of the 10 reflecting surfaces. The small off-diagonals indicate that leakage from input x-polarized or y-polarized into an orthogonal polarization state makes up only a small fraction of the light. The XX and YY elements are different in distribution because the s- and p- coefficients of reflection are slightly different at non-normal angles of incidence. The on-diagonal phase terms reveal that the fore-optics suffer from astigmatism and differing amounts of tilt. The π phase jump in the phase of the off-diagonals is the result of the zero crossing in the amplitude of the off-diagonals. Figure 4.7 shows the cumulative diattenuation and retardance of the fore-optics. The RMS diattenuation and retardance across the the fore-optics Jones pupil is 0.0322077. The closer the RMS diattenuation and retardance is to zero, the closer the Jones pupil is to the identity matrix.

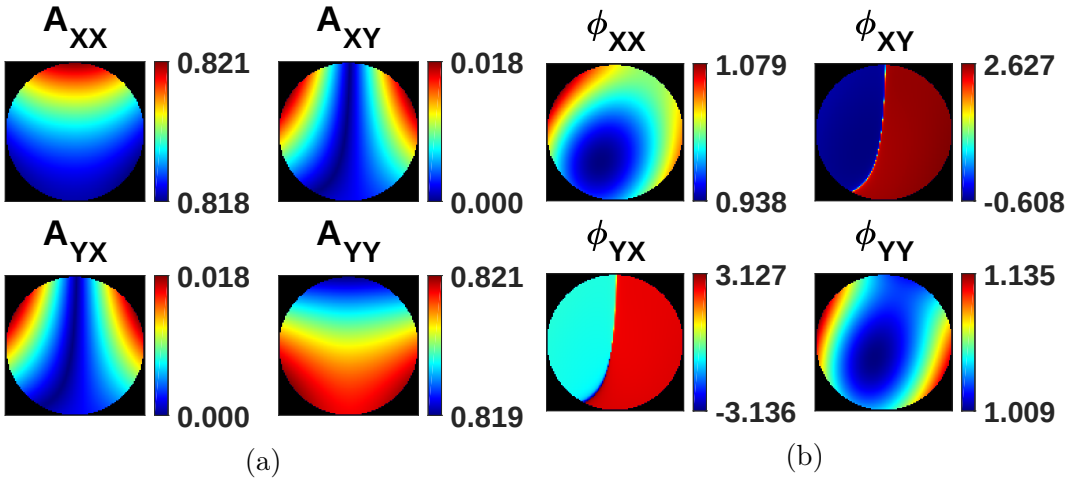


Figure 4.6: JP_A for a wavelength of 500 nm. (a) is the amplitude of the Jones pupil, (b) is the phase of the Jones pupil in radians

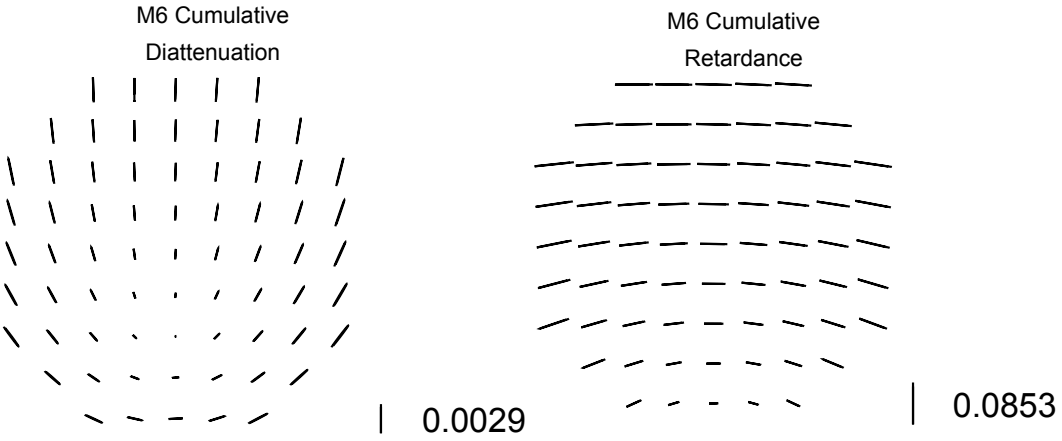


Figure 4.7: Diattenuation and retardance maps for JP_A. M6 is the sixth non-flat mirror in the system and is the optical element right before the coronagraph mask plane, making it the last element in the fore-optics.

Figure 4.8 shows the Jones pupil for the optics after the coronagraph mask returned from the polarization ray trace. This pupil was created by tracing a square grid of 101x101 rays in Polaris-M. The pixel spacing for this Jones pupil is 0.8 mm. The left side shows the amplitude and the right side shows the phase in radians. Similar to JP_A, the amplitude is very close to a scaled identity matrix. The amplitudes are not unity due to absorption from the four reflecting surfaces. The phase reveals that the optics after the mask plane do

not introduce as much astigmatism as the optics before the mask. Instead, the phase of the XX and YY elements show slight tilts in different direction which will cause the center of the final image to be slightly shifted depending on the polarization of the light. Figure 4.9 shows the cumulative diattenuation and retardance of the optics after the coronagraph mask. The magnitude of the diattenuation and retardance for JP_B are an order of magnitude smaller than the diattenuation and retardance for JP_A, indicating that the optics after the coronagraph mask barely contribute to the overall polarization aberrations of the entire system. The RMS diattenuation and retardance across the post-optics Jones pupil is 0.00202847.

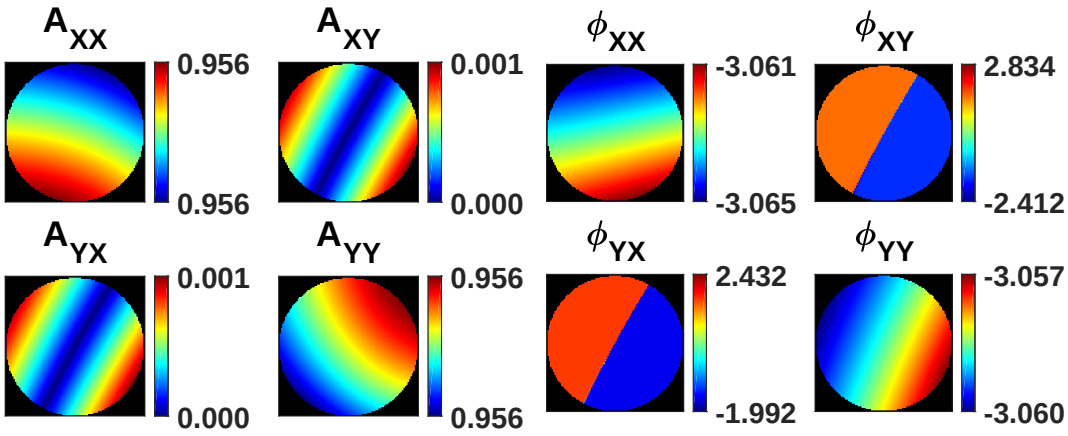


Figure 4.8: JP_B for a wavelength of 500 nm. (a) is the amplitude of the Jones pupil, (b) is the phase of the Jones pupil in radians

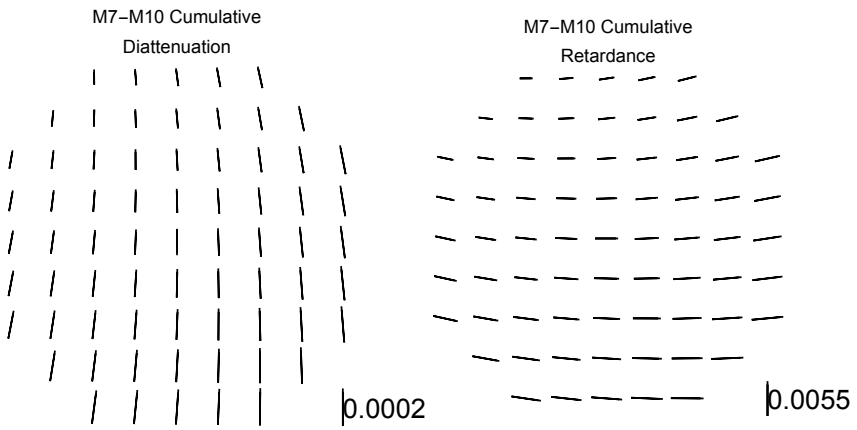


Figure 4.9: Diattenuation and retardance maps for JP_B. M7-M10 make up the last four mirrors in HabEx after the coronagraph mask.

The amplitude response matrix (ARM) associated with JP_A details the electric field distribution at the coronagraph mask plane. To calculate the ARM, each element of the Jones pupil was zero-padded by a factor of four before taking a discrete Fourier transform. Figure 4.10 shows the ARM corresponding to the Jones pupil shown in figure 4.6 on a log_{10} scale. The x and y axes are reported in arcseconds of angle on sky. The DFT extended out to ± 1.6 arcseconds, so this plot is zoomed by a factor of about 3. The ARM has been normalized by the maximum value of the ARM. The on-diagonals are very similar to an Airy pattern. The maximum value of the YY component is slightly larger than the XX component, but they are very close to one another. The off-diagonals are about two orders of magnitude smaller than the on-diagonal components and contain a bisected structure running through their center vertically.

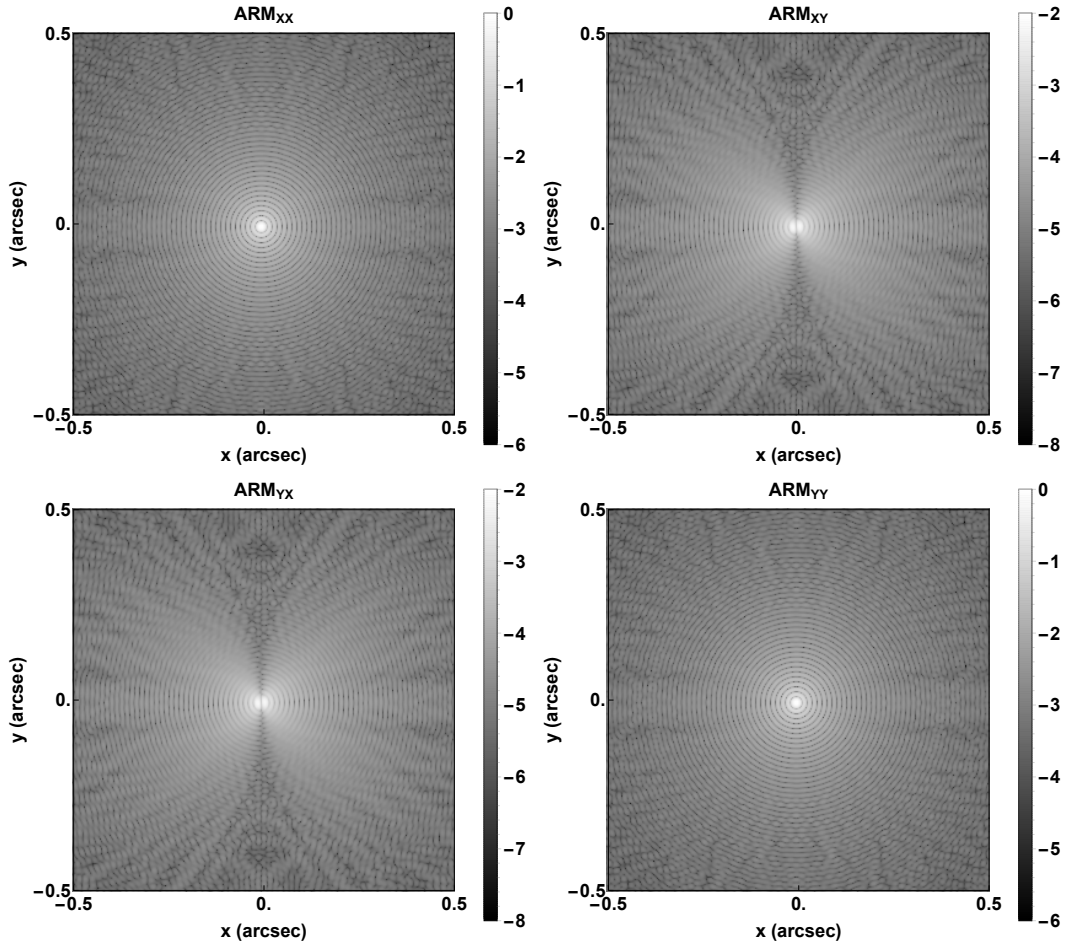


Figure 4.10: Amplitude response matrix of JP_A at 500 nm on a log_{10} scale. The on-diagonals are similar to an Airy pattern, while the off-diagonals have bifurcated structure due to the shape of the off-diagonals of the Jones pupil.

4.2.3 Implementing the Vector Vortex

Now that we have JP_A, JP_B, and the electric field distribution at the coronagraph mask plane from the polarization ray trace, the vector vortex can be implemented and the PSF at the final image plane can be calculated. The Jones matrix of the vector vortex is given by[55]:

$$\mathbf{VV}_m(x,y) = \begin{bmatrix} \cos(m\tan^{-1}(y/x) & \sin(m\tan^{-1}(y/x)) \\ \sin(m\tan^{-1}(y/x) & -\cos(m\tan^{-1}(y/x) \end{bmatrix} \tag{4.1}$$

where x and y are spatial coordinates in the coronagraph mask plane and

m is the charge (index) of the vector vortex. The vector vortex mask under consideration for HabEx is a charge 6 vector vortex. Figure 4.11 shows a visual representation of the Jones matrix for a charge 6 vector vortex mask.

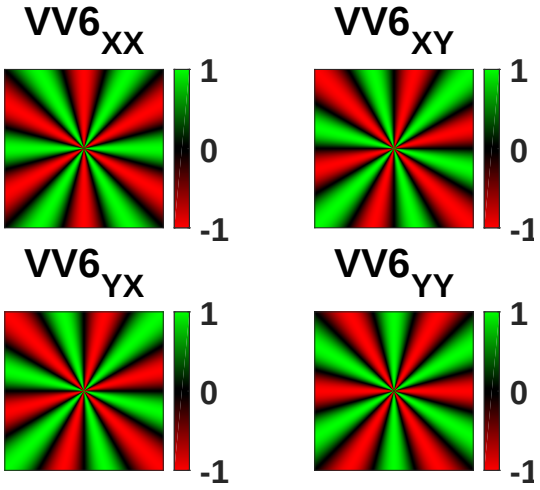


Figure 4.11: Jones matrix for a vector vortex charge 6 mask. The retardance is a half-wave for every point in the vector vortex.

The semi-analytic method[59] is used properly sample the highly spatially varying center of the vector vortex mask. This is done by applying a Tukey (tapered cosine)[68] windowing function to the product of the ARM and the vector vortex mask to select the highly varying core of the ARM/vortex product. The inverse of the windowing function selects the outer, less varying portion of the ARM/vortex product. Inverse DFTs of each windowed section are then stitched together to yield the masked Jones pupil. The masked pupil is shown in figure 4.12. The masked Jones pupil is the same size as the original Jones pupil for the fore-optics.

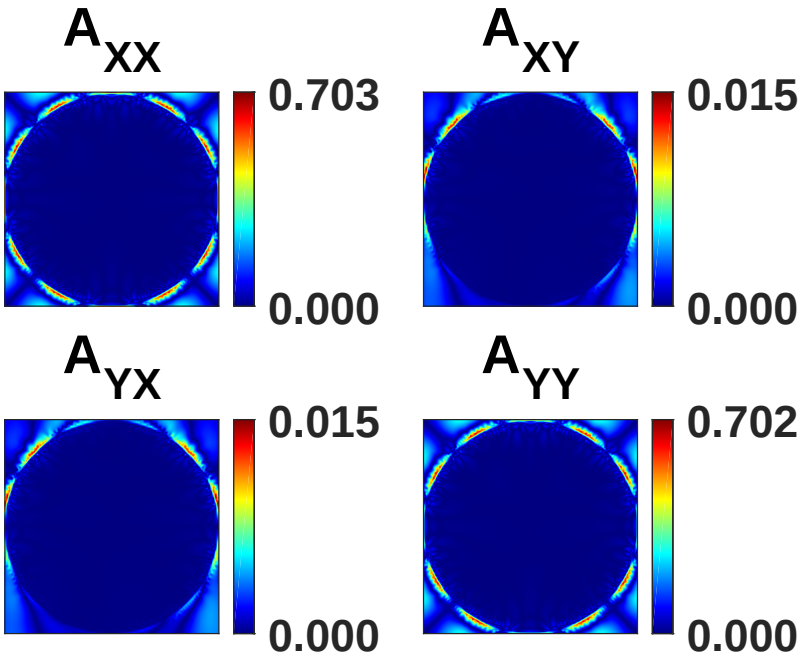


Figure 4.12: Amplitude of the Jones pupil at 500 nm with the vector vortex coronagraph. The ring of light around the pupil is the expected result of applying the vector vortex.

The masked pupil shows a "ring of fire", a distribution of light into a ring in the pupil. Though it is hard to see with the given color scale, there is still light inside the ring of fire. This residual light is a consequence of both the polarization aberrations and the finite extent of the coronagraph mask itself.

To calculate contrast, the overall Jones pupil for the system both with *and* without the vector vortex mask needs to be calculated. The overall Jones pupil for the system without the mask is the matrix product $JP_{total} = JP_B \cdot JP_A$. The system Jones pupil with the coronagraph mask is $JP_{total,masked} = JP_B \cdot JP_{masked}$. Figure 4.13 shows the magnitude and phase of the total system pupil with no mask. Figure 4.14 shows the diattenuation and retardance maps of the end-to-end Jones pupil. The RMS diattenuation and retardance for the Jones pupil without the mask is 0.0337945, which is very similar to the RMS diattenuation and retardance for the fore-optics' Jones pupil. Figure 4.15 shows

the magnitude and phase of the total system pupil with the vector vortex mask applied.

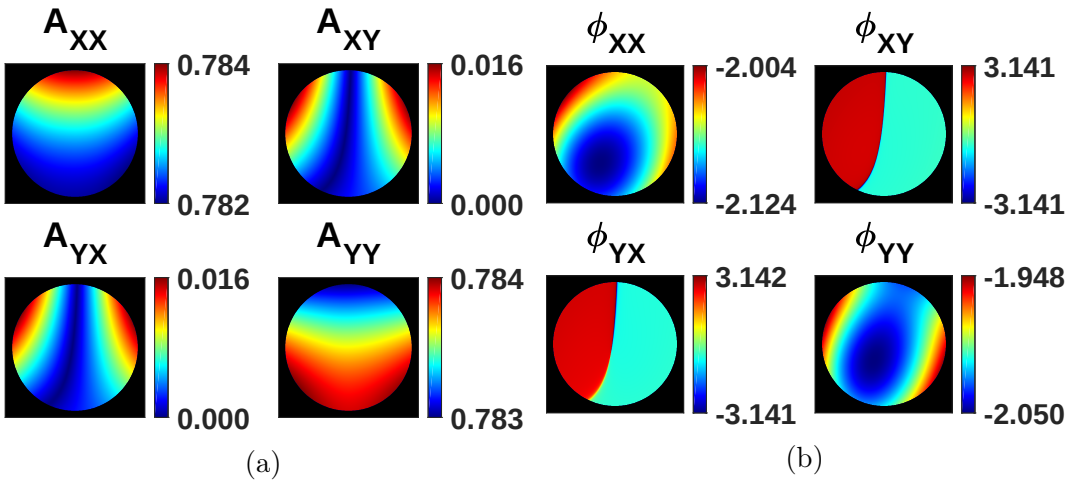


Figure 4.13: Magnitude (a) and phase in radians (b) of the total system Jones pupil at 500 nm with no mask applied.

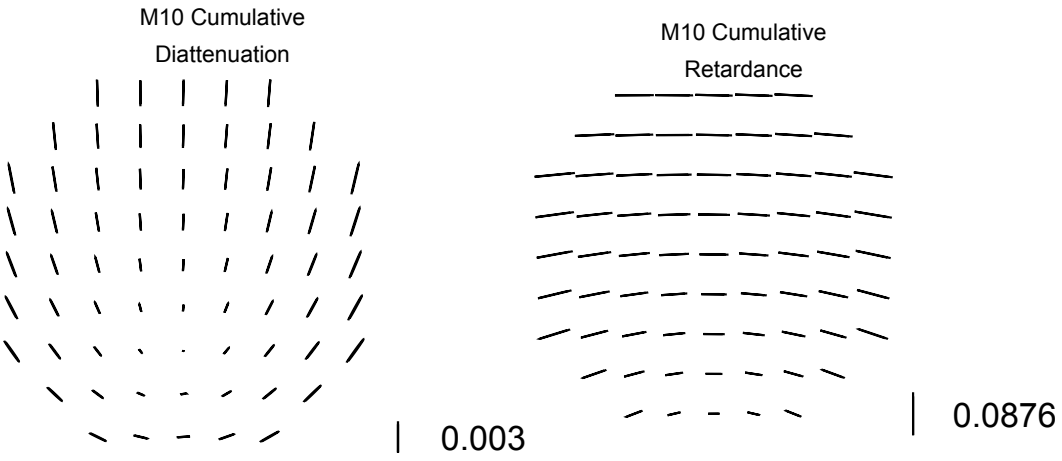


Figure 4.14: Diattenuation and retardance maps for JP_{total}. M10 is the last mirror in the HabEx system, making these maps cumulative for the entire system. Like the Jones pupil plots in figure 4.13, the diattenuation and retardance maps for the end-to-end Jones pupil are similar to diattenuation and retardance maps for JP_A. This is because the four mirrors after the coronagraph mask contribute very little to the overall polarization aberrations of the system.

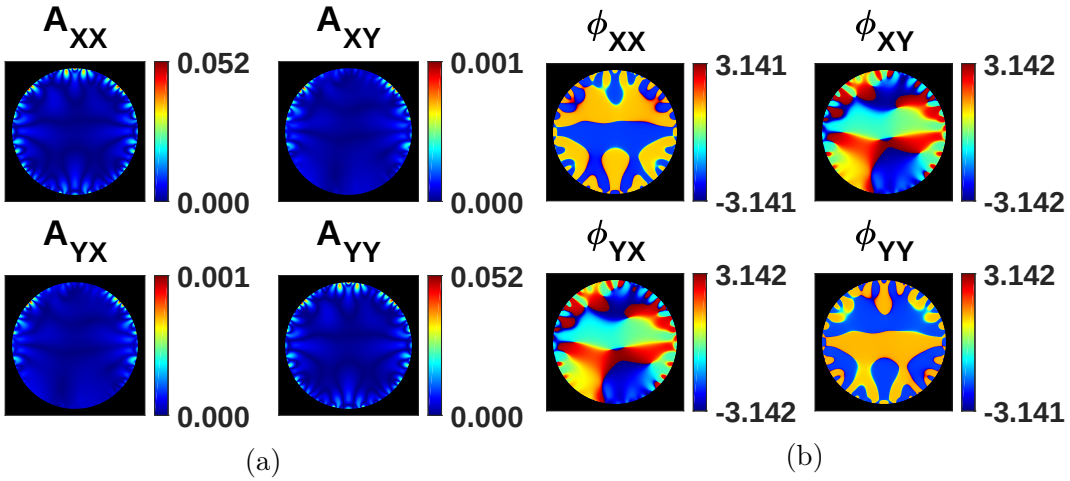


Figure 4.15: Magnitude (a) and phase in radians (b) of the total system Jones pupil at 500 nm with a vector vortex 6 mask applied. Large variations in the amplitude exist near the edges of the pupil. The phase is symmetric in the off-diagonals and anti-symmetric in the on-diagonals.

The total system pupil without the vector vortex mask looks very similar to JP_A, with some slight changes to the overall magnitude and phase. The total system pupil with the vector vortex mask shows the residual light that is able to pass through the vector vortex that is not rejected by the Lyot stop. There are high variations in the amplitude near the edges of each component of the system Jones pupil with the vector vortex, while the inner portion of the pupil is more slowly varying. The Lyot stop itself is applied to JP_B as an opaque binary mask with a radius that is 95% of the size of JP_B. Some of the high variations near the edges of the pupil could be eliminated by applying a smaller Lyot stop, but at the cost of losing light that can make it through the system. In the ideal case, the Jones pupil with the mask applied should have no light leftover after the Lyot stop. However, the finite size of the coronagraph mask itself leads to some of the on-axis light still getting through.

4.2.4 Contrast

Once the total pupil for the system has been calculated, all that is left to determine the system contrast is to find the PSF of the system with no mask and the PSF of the system with the mask. For unpolarized illumination, the four components of the amplitude response matrix are incoherent with respect to one another[18]. The system PSF (with or without the mask) is the sum of the intensities represented by the square modulus of the components of the ARM (with or without the mask), $PSF = |ARM_{XX}|^2 + |ARM_{XY}|^2 + |ARM_{YX}|^2 + |ARM_{YY}|^2$.

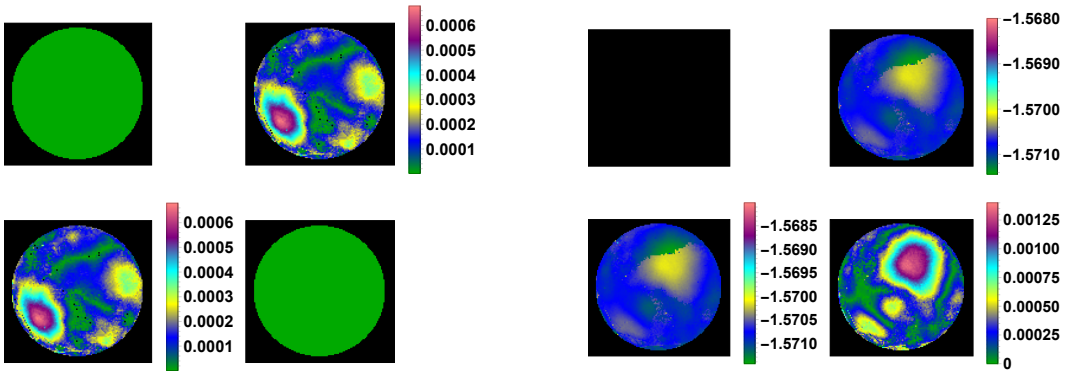
The contrast for three different circumstances is presented here: the ideal case, the isotropic coatings case, and the isotropic coatings case with an anisotropic map applied to the primary mirror. The ideal case replaces the Jones pupils calculated from the polarization ray trace with identity matrices within the circular support of the pupil. The isotropic coatings case is calculated directly from the polarization ray trace, using the pupils already shown above. The anisotropic primary case utilizes the isotropic Jones pupils but adds in an anisotropic map before the first Jones pupil to simulate the effects of form birefringence on the primary mirror.

4.2.4.1 Form Birefringence

Form birefringence is refractive anisotropy due to the micro-structure of the metal thin film. Mirror form birefringence can also contribute to polarization aberrations. For example, in the process of depositing a thin film coating of aluminum, the microscopic grain structure of the aluminum may be amorphous

in certain regions and columnar in other regions. This columnar structure appears to be attributed to oxidation of the tiny vapor droplets as they adhere to the glass substrate. This anisotropy leads to regions of the mirror exhibiting spatially-varying retardance associated with the deposition process.

The coating geometry used to deposit large area primary mirrors ($> 12m^2$) is different than that used for smaller mirrors ($< 2m^2$). The spatial dependence of the polarization reflectivity of a bare aluminum 3.75-meter diameter test sphere was measured at the University of Arizona Mirror Lab by Daugherty (2018)[42]. This mirror is used for optical wavefront testing of the 8-meter GMT primary mirrors. The bare aluminum was deposited using the same chamber and processes employed to coat the primary mirror for the Mayall 4-meter telescope at Kitt Peak National Observatory. The coating shows a spatial dependence of the polarization reflectivity caused by form birefringence. Figure 4.16 shows the Jones matrix associated with the form birefringence. To understand the potential role of primary mirror form birefringence in image formation, this form birefringence measurement was scaled and applied to the primary mirror of HabEx.



(a) Amplitude

(b) Phase in radians

Figure 4.16: Jones matrix amplitude (a) and phase in radians (b) of the birefringence map of the 3.75-meter diameter test sphere. The XX and YY components of the amplitude are unity across the map. The phase of the XX component is zero across the entire map due to convention in the conversion of the birefringence map from a Mueller matrix into a Jones matrix.

4.2.4.2 Two Dimensional Contrast Plots

The contrast maps presented here are broadband maps, utilizing PSFs for wavelengths in the waveband [450, 550] nm, the "blue" channel of the HabEx coronagraph. These broadband maps are created by calculating the PSF with the coronagraph mask at each wavelength, summing them together incoherently, calculating the PSF without the coronagraph mask at each wavelength, summing them together incoherently, then finally normalizing the broadband PSF with the mask by the maximum value of the broadband PSF without the mask (this maximum is at the center of the final image plane in these calculations).

$$Contrast(x,y) = \frac{\sum_{\lambda=450nm}^{550nm} PSF_{VV6}(x,y,\lambda)}{\sum_{\lambda=450nm}^{550nm} PSF_{nomask}(0,0,\lambda)} \tag{4.2}$$

Figure 4.17 shows the PSF calculated for five wavelengths with ideal apertures and no vector vortex mask on a log_{10} scale as well as the PSF of the sum

of all five wavelengths. Figure 4.18 shows the PSFs with ideal apertures and the vector vortex mask. The spacing between pixels in these PSFs is 0.003 arcseconds. At this sampling, there is no detectable shear between the XX and YY components of the ARM. However, higher sampling of the ARM (down to tenths of milliarcseconds) would reveal slight shifts in the center maximum for the XX and YY components of the ARM and thus a slight elongation of the center of the PSF.

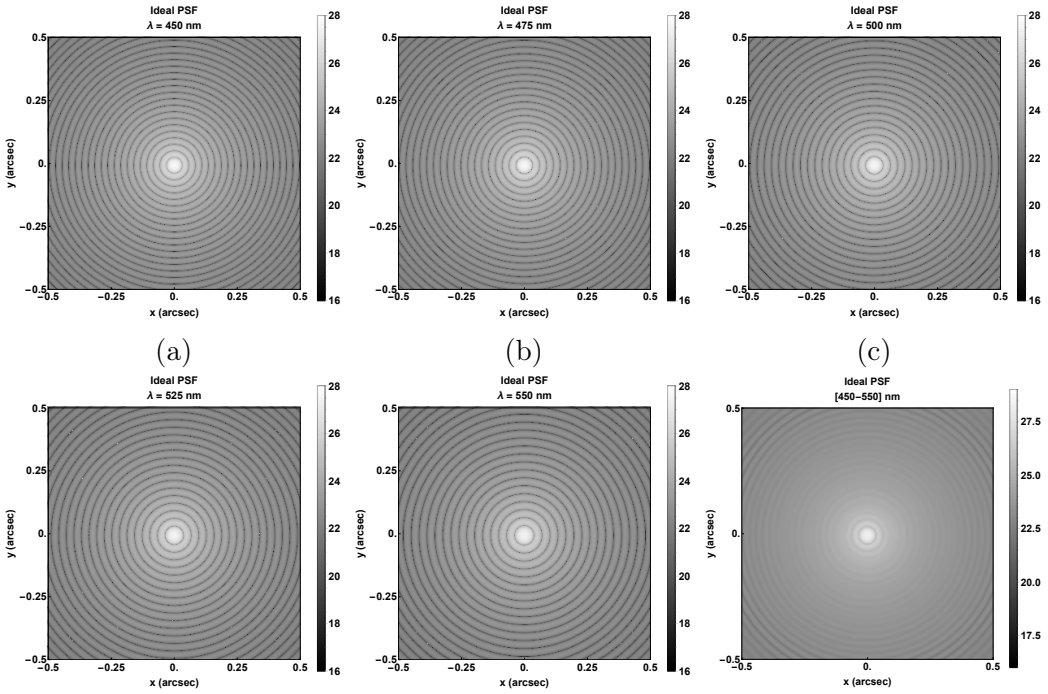


Figure 4.17: PSF for the ideal aperture case and no vector vortex mask on a log_{10} plot. (a) - (e) show the individual wavelengths' PSFs while (f) shows the sum of all the PSFs.

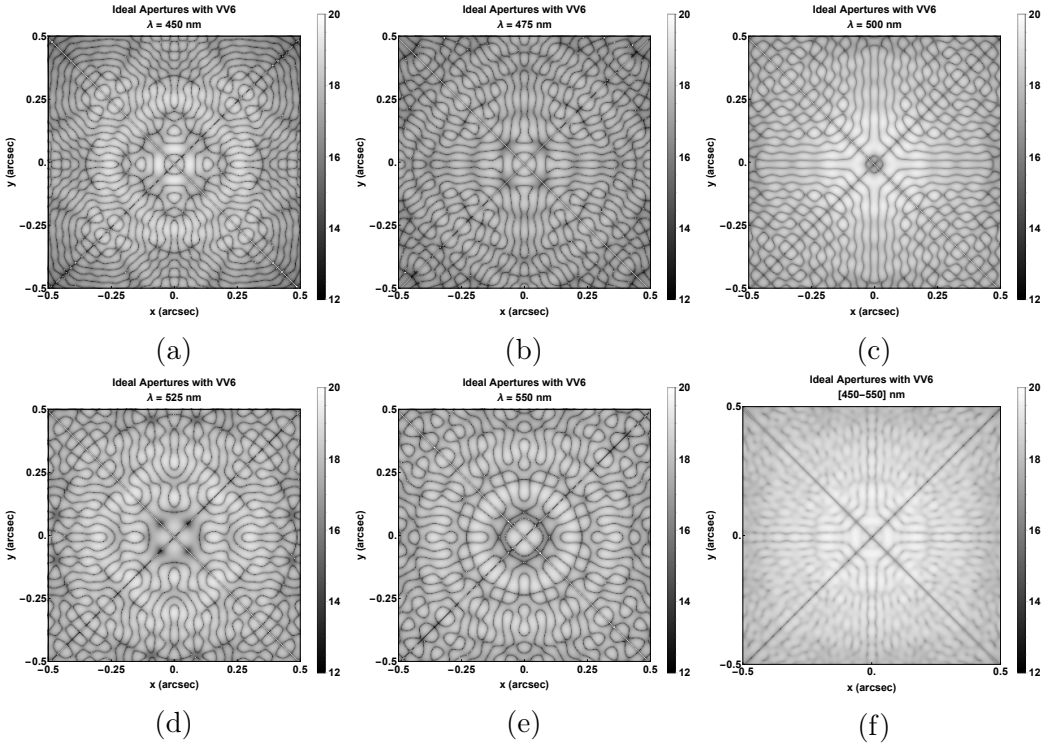
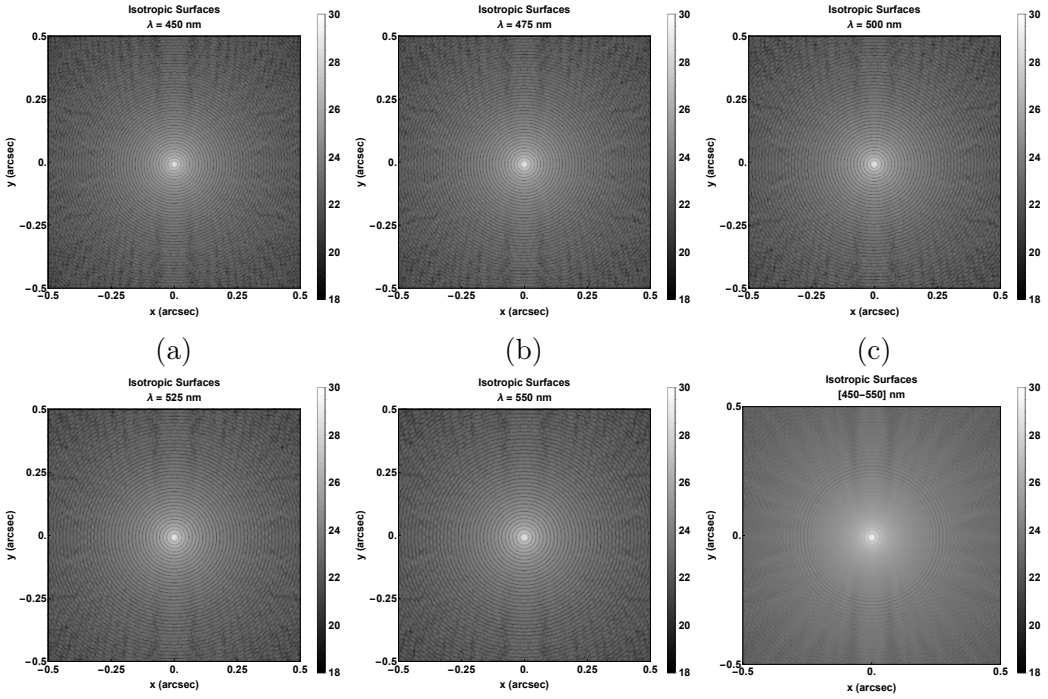


Figure 4.18: PSF for the ideal aperture case with the coronagraph mask on a log_{10} plot. (a) - (e) show the individual wavelengths' PSFs while (f) shows the sum of all the PSFs.

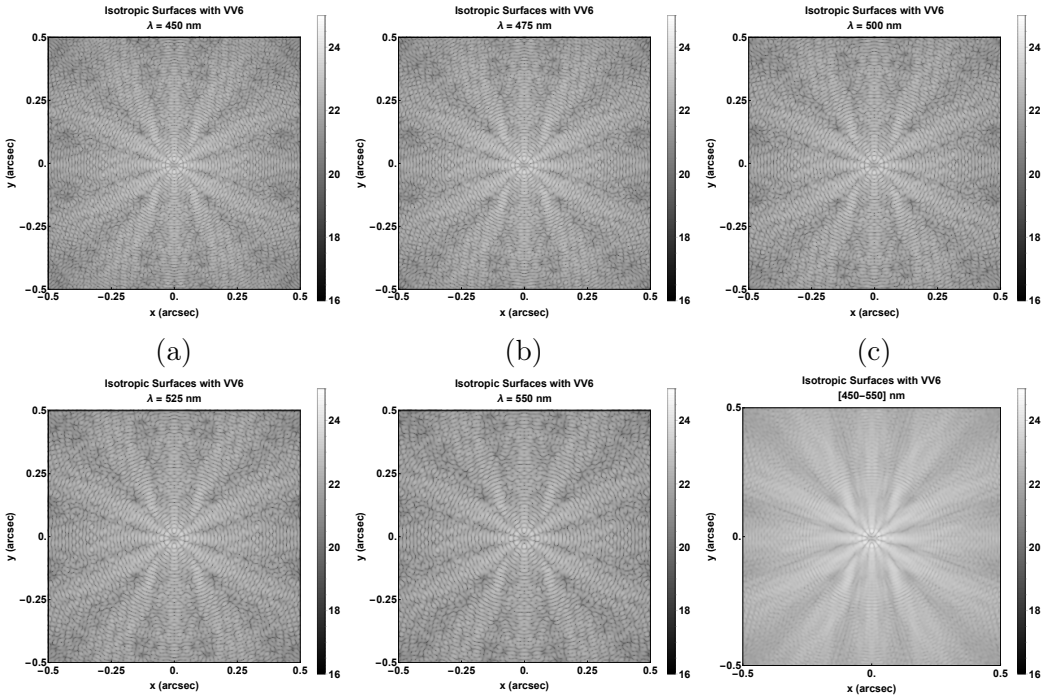
Figure 4.19 shows the PSF calculated for five wavelengths with isotropic coatings on the mirror surfaces and no vector vortex mask on a log_{10} scale as well as the PSF of the sum of all five wavelengths. Figure 4.20 shows the PSFs with isotropic coatings on the mirror surfaces and the vector vortex mask.



(a) (b) (c)

(d) (e) (f)

Figure 4.19: PSF for the isotropic coatings case and no vector vortex mask on a log_{10} plot. (a) - (e) show the individual wavelengths' PSFs while (f) shows the sum of all the PSFs.



(a) (b) (c)

(d) (e) (f)

Figure 4.20: PSF for the isotropic coatings case with vector vortex mask on a log_{10} plot. (a) - (e) show the individual wavelengths' PSFs while (f) shows the sum of all the PSFs.

Figure 4.21 shows the PSF calculated for five wavelengths with isotropic coatings on the mirror surfaces, the form birefringence map on the primary, and no vector vortex mask on a log_{10} scale as well as the PSF of the sum of all five wavelengths. Figure 4.22 shows the PSFs with isotropic coatings on the mirror surfaces, the form birefringence map on the primary, and the vector vortex mask.

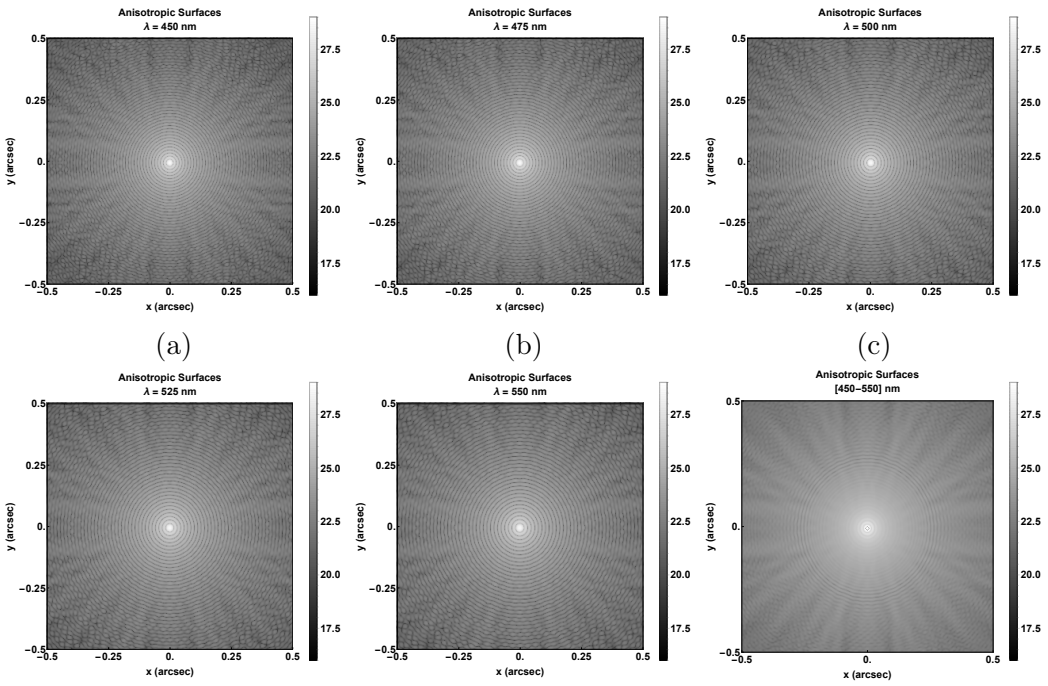
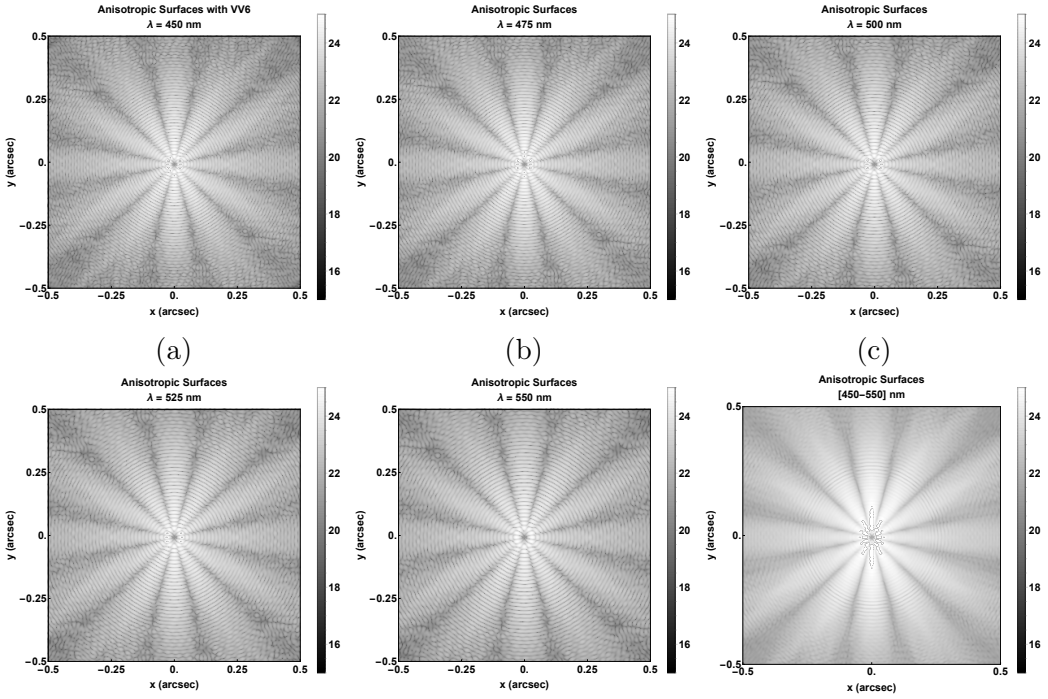


Figure 4.21: PSF for the isotropic coatings case, birefringence map on the primary, and no vector vortex mask on a log_{10} plot. (a) - (e) show the individual wavelengths' PSFs while (f) shows the sum of all the PSFs.

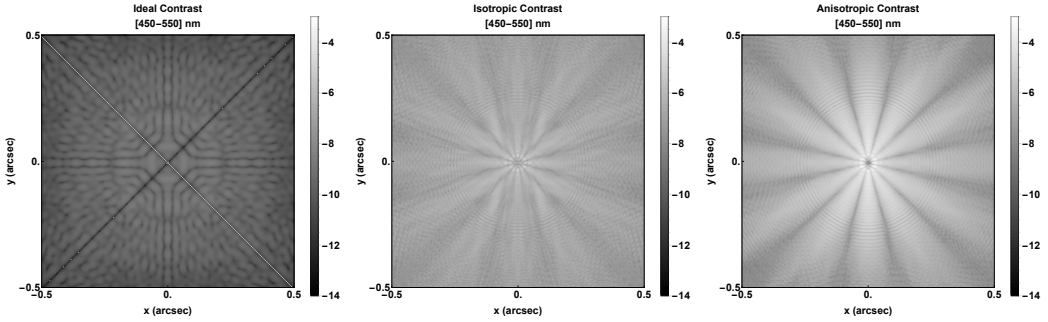


(a) (b) (c)

(d) (e) (f)

Figure 4.22: PSF for the isotropic coatings case, birefringence map on the primary, with vector vortex mask on a log_{10} plot. (a) - (e) show the individual wavelengths' PSFs while (f) shows the sum of all the PSFs.

Figure 4.23 shows two dimensional plots of the contrast calculated for the each of the three cases: (a) ideal apertures, (b) isotropic coatings on every mirror surface, and (c) isotropic coatings on every mirror surface *and* the form birefringence map added to the primary mirror.



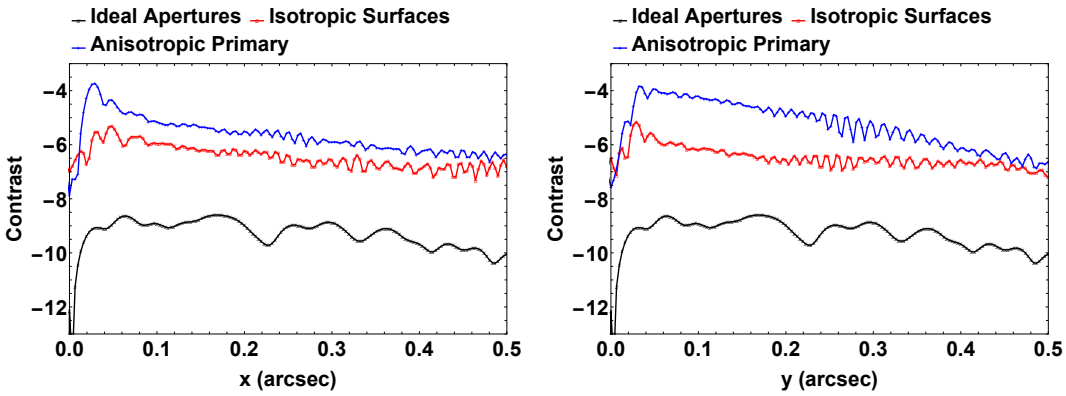
(a) Ideal Apertures (b) Isotropic Coatings (c) Anisotropic Primary

Figure 4.23: Contrast plots on a log_{10} scale for (a) ideal apertures, (b) isotropic coatings on every mirror, and (c) isotropic coatings on every mirror and the form birefringence map added to the primary mirror.

The contrast degrades as we move from ideal apertures to isotropic coatings by a few orders of magnitude. After adding the form birefringence to the primary mirror, the contrast degrades by another couple orders of magnitude. This is more immediately obvious when we take a closer look at cross-sectional slices through the two dimensional contrast maps.

4.2.4.3 Cross-sectional Slices of Contrast

Figure 4.24 is arguably the most important figure in this dissertation. It shows horizontal and vertical cross-sectional slices through the two dimensional contrast maps on a log_{10} scale. These slices are nearly symmetric about the center of two dimensional contrast maps, so the horizontal range spans only [0, .5] arcseconds. The gap between the black curve, representing ideal contrast, and the blue and red curves, representing contrast with isotropic coatings and contrast with an anisotropic primary mirror respectively, is several orders of magnitude regardless of position in the final image plane. This indicates that the ability of the vector vortex to suppress on-axis starlight worsens as polarization aberrations are added to the system. The difference between the isotropic case and anisotropic primary case is greatest near the center of the final image plane. This indicates that form birefringence will severely limit a coronagraph's ability to suppress light near the ideal case's inner working angle. The inner working angle is the smallest separation angle at which a faint exoplanet can theoretically be detected when the coronagraph is in use[69]. The inner working angle for the "blue" channel's coronagraph is reported to be $2.5\lambda/D$ or 58 milliarcseconds[67].



(a) Horizontal slice (b) Vertical slice

Figure 4.24: (a) Horizontal and (b) vertical cross-sectional slice through the two dimensional contrast maps for the HabEx telescope with a VV6 coronagraph.

4.2.5 Closing Remarks

In modern coronagraph applications, there is often a system of wavefront control that is used to "dig out" a dark hole[70], an area in the final image plane where unwanted on-axis starlight has been moved and redistributed to other parts of the image plane. This control is achieved by using correcting elements, often deformable mirrors, to change the wavefront as it propagates to the final image plane. Processes like electric field conjugation [71] are used in scalar image formation to calculate the effect of probing different parts of the deformable mirror on the final image plane. However, with polarization considerations, the wavefront at the final image plane is the incoherent sum of the four elements of the modulus square of the ARM, and the deformable mirrors can *not* correct all four elements at once. It can correct for any one of these four elements or the mean of all four elements.

The contrast analysis presented here does not include adaptive optics. Implementation of wavefront control with polarization considerations was not feasible for the author, but the results presented here (most importantly figure

4.24 stand as a cautionary tale; modeling polarization aberrations shows how coatings negatively affect the possible contrast of a coronagraph. Therefore, polarization should be taken into account when designing adaptive optics and wavefront control that dig out the dark hole in the final image plane lest the faint exoplanet's signal be masked by polarization effects.

4.3 An Aside About Index of Refraction Interpolation in Ray Tracing Programs

When different ray tracing programs trace an optical system, if their thin film packages are different, the resulting Jones pupils will inevitably be different. This section presents an interpolation issue noted when comparing the thin film outputs from OpticStudio's Zemax [72] and AiryOptics' Polaris-M [21].

4.3.1 Background

Reflecting surfaces in an optical system are generally coated with a dielectric thin film stack to both protect the surface and enhance the reflection from the surface. The polarization properties from the metal coated surface as well as the thin film stack need to be calculated so that the overall affect of the polarization properties of the system can be assessed. The polarization properties of a surface are dictated by the angle of incidence at the surface and the indices of refraction for each layer of the thin film stack. Thus any thin film analysis program needs both angle and refractive index to return meaningful polarization data. A ray tracing program will calculate the angle of incidence from the ray trace itself, so all that needs to be supplied is the index

of refraction for the thin film. In Zemax, this data is supplied to the program in the form of a coating file. This coating file contains index of refraction data for different materials and prescriptions for different thin films. The refractive index is input as a table of data with three columns: wavelength, real part of the index of refraction, and imaginary part of the index of refraction. Table 4.2 shows an example for aluminum.

Wavelength (μm)	n (real)	k (imaginary)
0.400000	0.400000	-4.450000
0.436000	0.470000	-4.840000
0.450000	0.510000	-5.000000
0.492000	0.640000	-5.500002
0.500000	0.665936	-5.576610
0.546000	0.820000	-5.990000
0.578000	0.930000	-6.330000
0.650000	1.300000	-7.110000
0.700000	1.550000	-7.000000
0.750000	1.800000	-7.119999
0.800000	1.990000	-7.050000
0.850000	2.080000	-7.150001
0.900000	1.960000	-7.700000

Table 4.2: Example of refractive index tabulated data for aluminum

But what if the ray trace needs refractive index data at a wavelength not present in the table of refractive index data fed into the program? For example, what if 633nm light was being traced through the system? Since the table of data has 578nm and 650nm data, but no data point in between, the program must use its own interpolation to extract the value of the real and imaginary parts of the refractive index at that wavelength.

4.3.2 Thin Film Performance, No Subroutine Interpolation

Before looking at the outputs of the thin film packages with interpolated input data, we should first confirm that the thin film subroutines in Polaris-M and

Zemax agree with one another with a given index of refraction. One of the data points in table 4.2 is 700nm, so we'll look at the reflection coefficients from the two programs at this wavelength. Figure 4.25 shows the difference between Zemax and Polaris-M for the magnitudes and phases of the reflection coefficients when using data from a non-interpolated wavelength. Across a broad range of angles, the differences between the two thin film subroutines are very small, on the order of 10^{-6}. The rapid oscillations seen in the difference of the phase is due to the precision of the numerical data extracted from Zemax. These non-zero differences do show that there is a slight difference in the thin film algorithms, and it is up to the reader whether differences at these levels are significant. For the purpose of this paper, this level of agreement is acceptable.

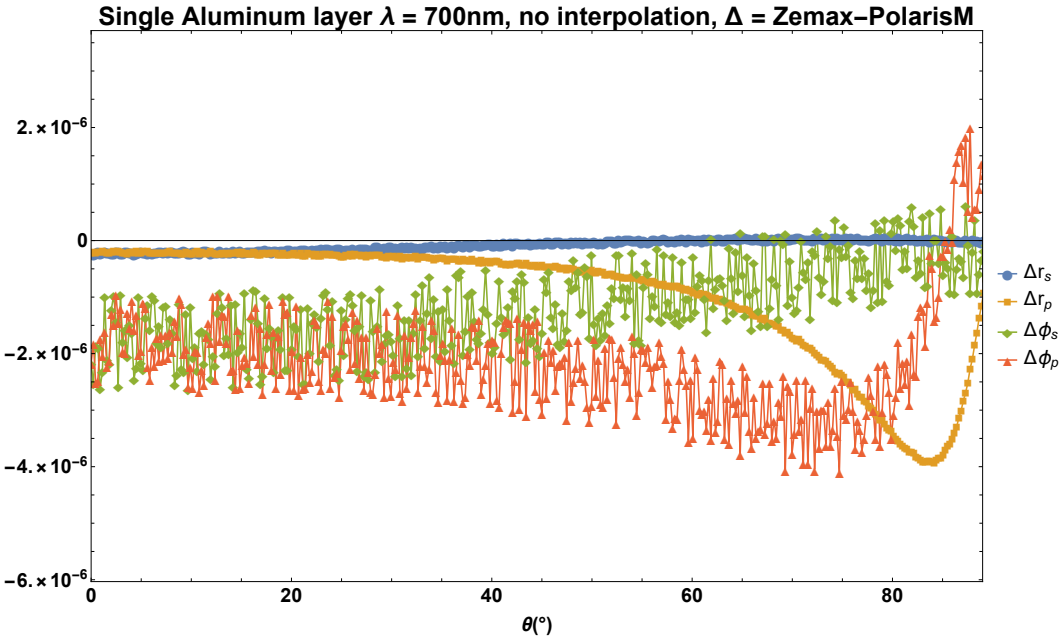


Figure 4.25: Difference between Zemax and Polaris-M reflection (magnitude and phase) for a non-interpolated wavelength, 700nm

4.3.3 Thin Film Performance With Subroutine Interpolation

Figure 4.26 shows the difference between Zemax and Polaris-M for the magnitudes and phases of the reflection coefficients when using data from an interpolated wavelength. Whereas the differences at a non-interpolated wavelength were on the order of 10^{-6}, the differences with an interpolated wavelength are on the order of 10^{-3}. So allowing each program to do its own interpolation leads to differences in the calculated magnitude and phase that are three orders of magnitude worse than the non-interpolated case.

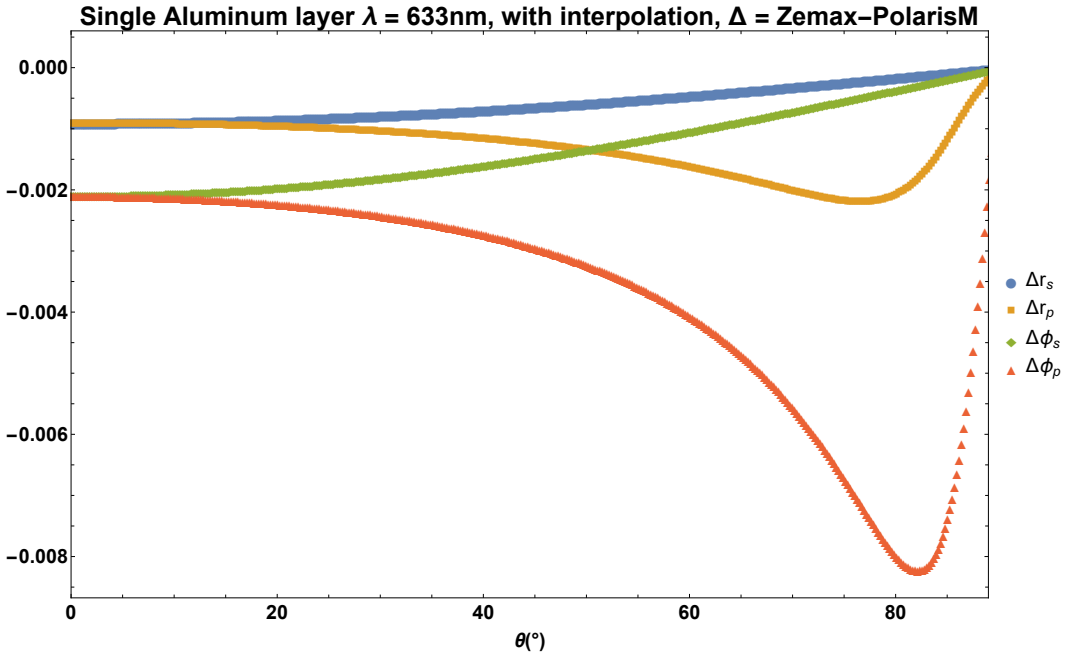


Figure 4.26: Difference between Zemax and Polaris-M reflection (magnitude and phase) for an interpolated wavelength, 633nm

4.3.4 Current Solution

Since the only thing that changed between calculating the thin film performance at 700nm and 633nm was the interpolation, we have established that the interpolation algorithm that Zemax and Polaris-M use are different. It is not within the scope of this dissertation to deem either algorithm "correct" or

"incorrect". With all else being the same, the polarization dependent pupils that Zemax and Polaris-M output will be different because of this interpolation issue. Consulting with Jet Propulsion Lab optical engineer Stefan Martin, the current solution to this problem was deemed to be straightforward; don't let the programs use their interpolation. Instead, the interpolation is done by hand, then that data is fed into the table of refractive index values. This way, the programs won't use their own interpolation and go back to being in agreement at the 10^{-6} level. Table 4.3 shows the new input data set to be fed into Zemax and Polaris-M. Figure 4.27 shows the results when we force both Polaris-M and Zemax to recognize 633nm as a data point in its table of refractive index data. Once again, the rapid oscillations seen in figure 4.27 are attributed to issues with precision from the data extracted from Zemax.

Wavelength (μm)	n (real)	k (imaginary)
0.400000	0.400000	-4.450000
0.436000	0.470000	-4.840000
0.450000	0.510000	-5.000000
0.492000	0.640000	-5.500002
0.500000	0.665936	-5.576610
0.546000	0.820000	-5.990000
0.578000	0.930000	-6.330000
0.633000	1.206590	-6.981450
0.650000	1.300000	-7.110000
0.700000	1.550000	-7.000000
0.750000	1.800000	-7.119999
0.800000	1.990000	-7.050000
0.850000	2.080000	-7.150001
0.900000	1.960000	-7.700000

Table 4.3: Example of refractive index tabulated data for aluminum. The highlighted row shows the forced interpolation for 633nm that has been added to the table

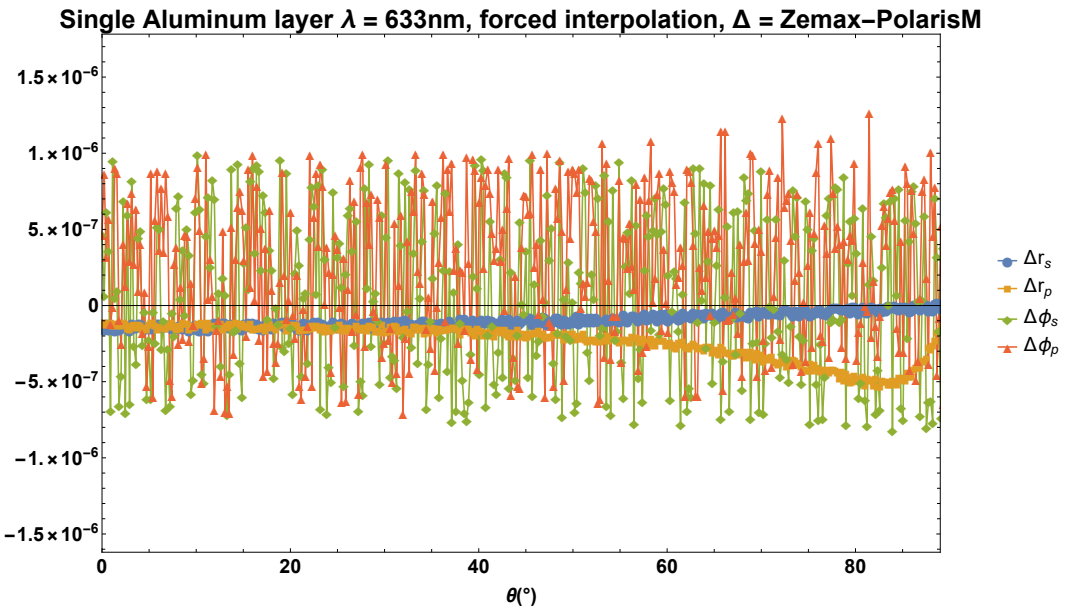


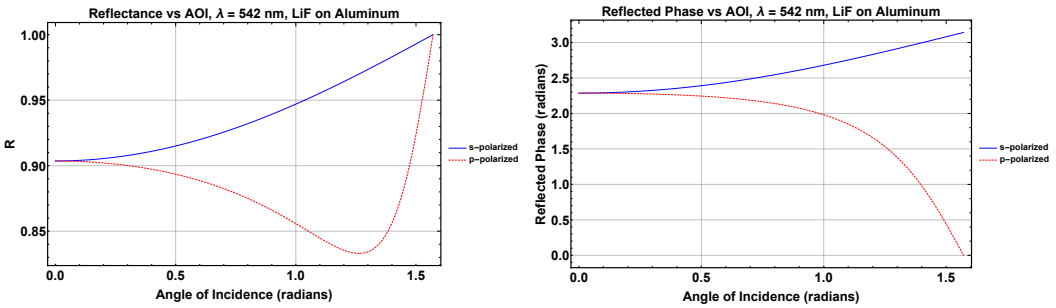
Figure 4.27: Difference between Zemax and Polaris-M reflection (magnitude and phase) for a wavelength that was interpolated by hand, 633nm

4.4 LUVOIR

The Large UV/Optical/IR Surveyor (LUVOIR) is a concept for a multi-wavelength space observatory. This mission will investigate a broad range of science, from the epoch of reionization, through galaxy formation and evolution, star and planet formation, to solar system remote sensing. LUVOIR also has the goal of characterizing a wide range of exoplanets, including those that might be habitable or even inhabited[62]. LUVOIR will have access to a UV-optical-NIR coronagraph, imager and spectrographs for the optical and NIR channels, a high definition imager for astronomical imaging, and ultra-violet multi-object spectrograph[73, 74, 75, 76, 77]. Presented here is a preliminary analysis of the polarization properties of LUVOIR's visible channel coronagraph fore-optics.

4.4.1 Optical Prescription

The fore-optics of the LUVOIR coronagraph system (that is, everything before the coronagraph focal plane mask) has 15 reflecting surfaces. The first four surfaces have a coating of 21 nm of lithium fluoride on aluminum, the next four surfaces (surfaces five to eight) have a coating of 25 nm of magnesium fluoride on aluminum, and every surface beyond that is coated in FSS99. Figure 4.28 shows the reflectance and phase change for s- and p-polarized light at 542 nm, which is the central wavelength of the visible channel, for a coating of 21 nm of lithium fluoride on aluminum. See figures 4.4 and 4.5 for the performance of the other two coatings (n.b. those figures are for a reference wavelength of 500 nm, but the performance does not change appreciably between 500 nm and 542 nm).



(a) (b)

Figure 4.28: Thin film performance of 21 nm LiF on aluminum at a wavelength of 542 nm. (a) shows the reflectance of the s- and p-polarized light versus angle of incidence. (b) shows the reflected phase for s- and p-polarized light versus angle of incidence

Figures 4.29, 4.30, and 4.31 (taken from Bolcar et. al 2017 and Bolcar et. al 2018) show the on-axis version of the LUVOIR system's optical telescope element (OTE) and the corongraph paths. A second, off-axis architecture is being considered[78]. Bolcar only shows the on-axis version, but this analysis

will utilize the off-axis version of the main telescope assembly to avoid the central obscuration of the primary.

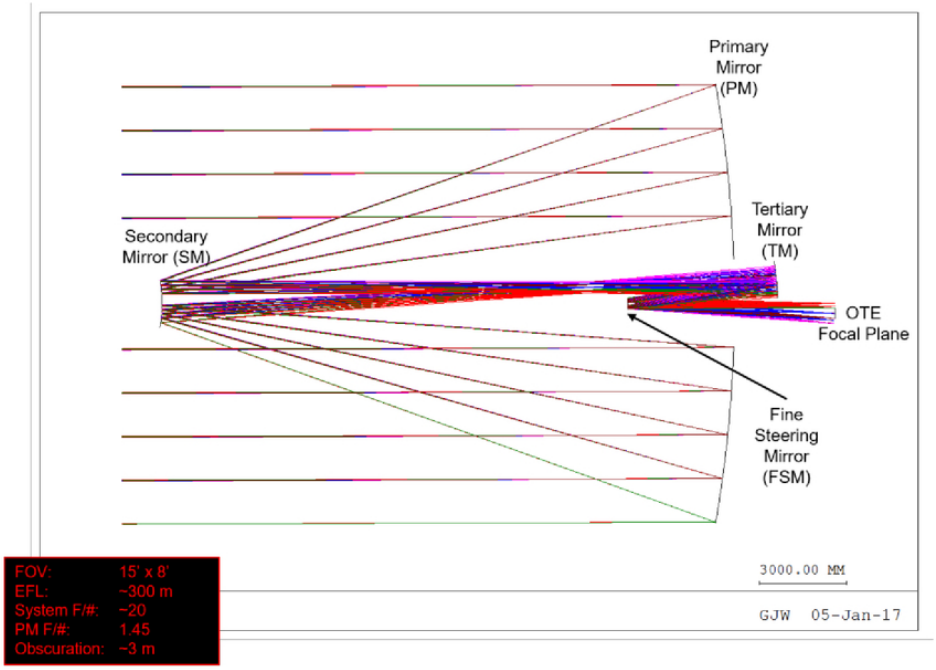


Figure 4.29: LUVOIR OTE optical design. First-order optical parameters are shown in the inset.

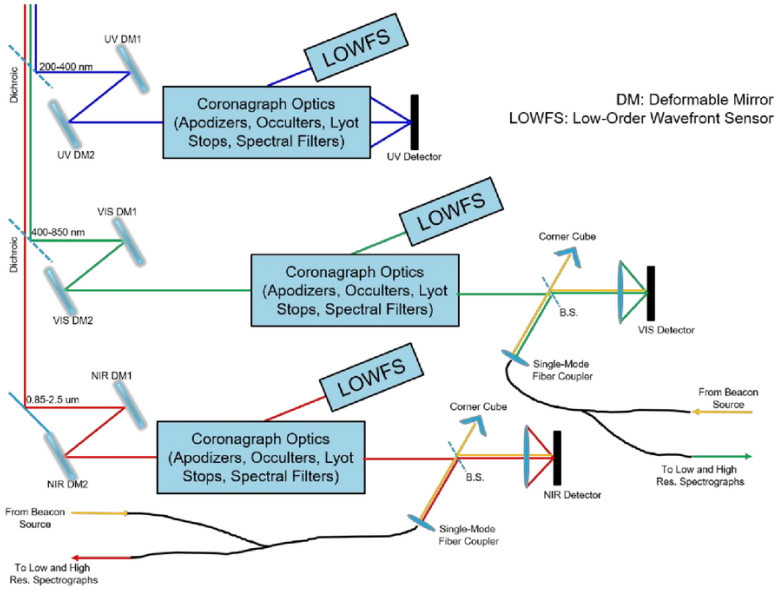


Figure 4.30: Block diagram of the coronagraph instrument. Light from the OTE enters in the upper left corner and a sequence of dichroic beamsplitters separates the UV, Vis, and NIR channels of the instrument.

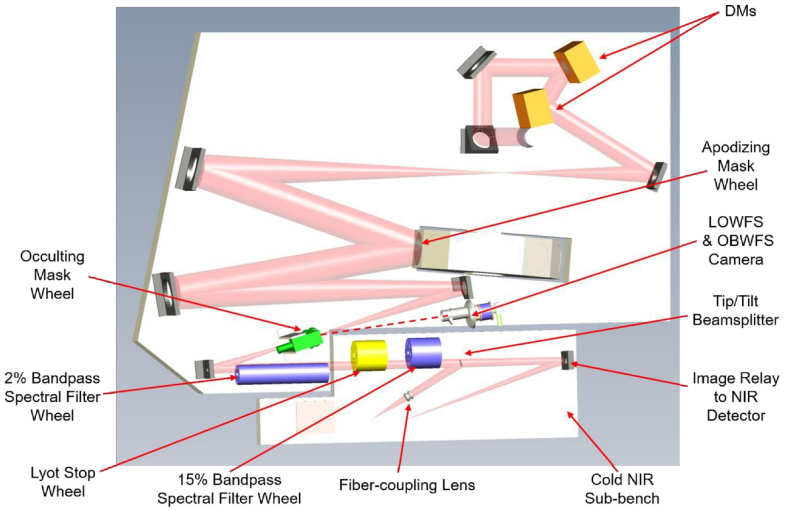


Figure 4.31: Top-down view of just the NIR channel of the coronagraph. Light enters from the OTE perpendicular to the plane in the upper right hand corner before being sent to the pair of deformable mirrors.

Though the current plans for the LUVOIR primary involve a primary aperture made of segmented hexagons, this analysis makes the simplifying assumption that the primary is continuous across its surface and that the planform area of the primary is equal to the area of the circle that circumscribes the outer-most hexagons.

4.4.2 Polarization Ray Trace of Fore-Optics Outputs: Pupil and ARM

Figure 4.32 shows the Jones pupil for the fore-optics returned from the LUVOIR polarization ray trace. The pupil is about 3.7 mm across and sampled with 101x101 ray grid. The left side shows the amplitude of the pupil and the right side shows the phase in radians. The off-diagonals amplitudes are about 36 times smaller than the on-diagonals, so only a relatively small amount of the light is transformed into an orthogonal polarization state. The on-diagonals are close to one another in magnitude; there's only about 1% difference be-

tween the XX and YY amplitude. This difference arises from the different amplitude reflection coefficients for the coatings, compounded by the fact that there are 15 reflections which magnify the small differences in the s- and p-reflection coefficients. The on-diagonal phases show (in order from greatest to least amount) tilt, astigmatism, and coma. The off-diagonals suffer a π phase jump at the zero crossing that occurs in the center of the off-diagonal amplitude maps. Figure 4.33 shows the cumulative diattenuation and retardance maps for the fore-optics. The RMS diattenuation and retardance for the fore-optics is 0.0581159.

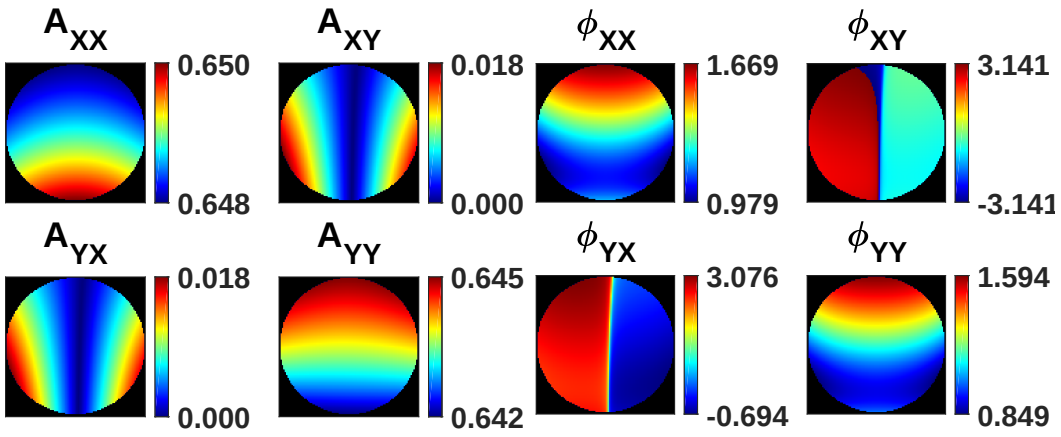


Figure 4.32: Magnitude (a) and phase in radians (b) of the LUVOIR fore-optics at 542 nm.

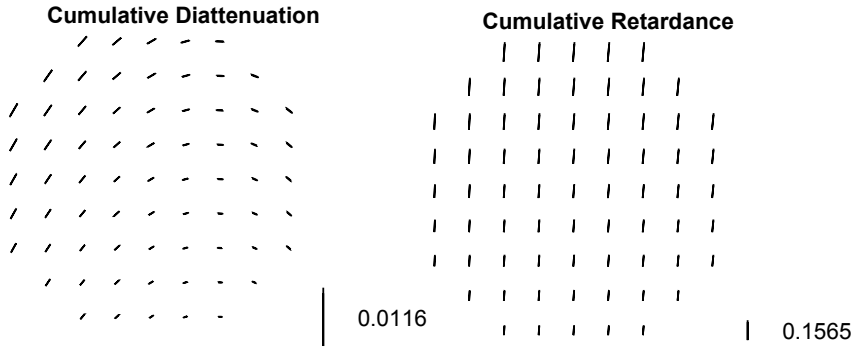


Figure 4.33: Cumulative diattenuation and retardance maps for the fore-optics of LUVOIR at 542 nm.

The amplitude response matrix associated with the LUVOIR fore-optics Jones pupil is shown in figure 4.34. The ARM was calculated by zero-padding Jones pupil by a factor of five and then taking a DFT. The figure is shown on a log_{10} scale and has been normalized by the maximum value of the XX component and has been zoomed in to a range of [-50,50] microns along x and y. This ARM is very similar to the HabEx fore-optics ARM (figure 4.10). The on-diagonals are close to an Airy pattern. The maximum value of the YY component is slightly larger than the XX component. The off-diagonals are about two orders of magnitude smaller than the on-diagonal components and contain a bisected structure running through their center vertically.

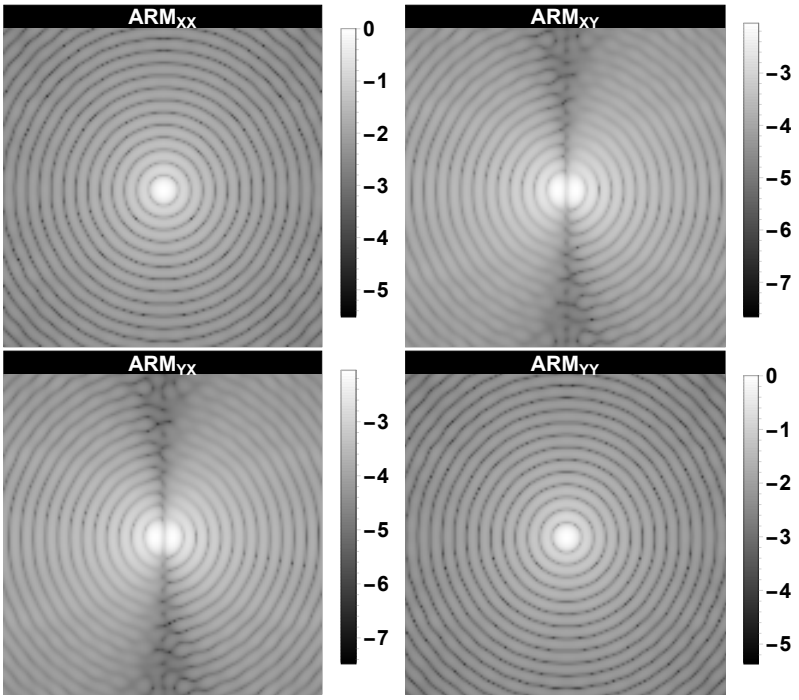


Figure 4.34: Amplitude response matrix of LUVOIR fore-optics Jones pupil at 542 nm on a log_{10} scale. The on-diagonals are similar to an Airy pattern, while the off-diagonals bifurcated due to the structure in the off-diagonals of the Jones pupil.

4.5 Conclusion

Polarization ray tracing was utilized to model the polarization aberrations of the HabEx coronagraph system and the fore-optics of the LUVOIR coronagraph system. These polarization aberrations, which can be seen from the deviations of the system Jones pupil from an identity matrix, serve to reduce the contrast of the coronagraph, inhibiting the ability to detect faint exoplanets. Adaptive optics were not modeled in these simulations; the deformable mirrors were modeled as stationary flat mirrors. Form birefringence was considered on the HabEx primary mirror, and it degraded the contrast beyond the isotropic coatings case. Therefore, polarization aberrations need to be included in adaptive optics modeling and tolerance analysis of the form birefringence should be carried out to ensure the contrast is not compromised.

CHAPTER 5

Polarization Mitigation Algorithm

This chapter describes one technique to mitigate some of the polarization aberrations inherent in an optical system. What follows is an explanation of how to rotate fold mirrors in an optical system to decrease the average diattenuation and retardance in an optical system and details an algorithm for the optimization of those fold mirror rotations.

5.1 Polarization Aberration Mitigation

The polarization aberrations diattenuation and retardance arise as a consequence of the difference in s-polarized and p-polarized Fresnel reflection and transmission coefficients as a function of angle of incidence. There are a few strategies to control and mitigate polarization aberrations. These include (1) reducing angles of incidence at each surface in an optical system, (2) reducing coating-induced polarization, (3) employing polarization compensating elements, and (4) using a crossed fold mirror configuration[18].

5.1.0.1 Angles of Incidence

Diattenuation and retardance are quadratic with respect to the angle of incidence (assuming the angles are not too large). Thus reducing the maximum angle of incidence at an optical surface will help reduce the diattenuation and

retardance. Reducing the maximum angle of incidence can be achieved by reducing the f-number of the surface and changing fold mirror orientations to shallower deviation angles.

5.1.0.2 Coating-induced Polarization

As the angle of incidence at a coated interface changes, the coefficients of reflection or refraction for s-polarized and p-polarized light will also change. How quickly these coefficients change with respect to angle and one another is dependent on the thin film prescription of the coating. The prescription may be optimized to adjust the diattenuation and retardance of the coating, thereby reducing the polarization aberrations of the system. However, it is be difficult to do so over a large range of angles or for multiple wavebands.

5.1.0.3 Polarization Compensating Elements

Diattenuation and retardance from an optical system can be compensated by inserting diattenuators and retarders that have equal magnitude but orthogonal orientation to the system diattenuation and retardance. Such elements can be viewed as the matrix inverse of the Jones pupil for the aberrated system. One possibility is to use liquid crystal polymers to fabricate an optical element that matches the spatially varying diattenuation and retardance in magnitude but orthogonally oriented[79]. However, the uniformity of liquid crystal retarders are usually around $\lambda/50$ while the retardance fluctuations for the HabEx system are on the scale of milliwaves ($\lambda/1000$), so it is not feasible to correct for these small polarization aberrations with liquid crystals.

5.1.0.4 Crossed Fold Mirrors

A crossed fold mirror configuration in which the s-orientation of the first fold mirror becomes the p-orientation at the second fold mirror will compensate for both diattenuation and retardance[80]. One point in the pupil can be completely compensated (zero diattenuation and retardance) while the rest of the pupil will have reduced diattenuation and retardance.

5.2 Rotating Surface Orientations

The crossed fold mirror configuration has been shown to compensate the polarization aberrations for a single angle to zero and leaves a linear variation of diattenuation and retardance for a spherical wavefront[81]. A second pair of crossed fold mirrors can compensate the remaining linear variation and leave reduced quadratic variation in a large field of view. While the crossed fold mirror configuration can be applied such that a fold mirror compensates for the fold mirror immediately before it, this compensating principle can be applied to any system that contains any number of fold mirrors (not necessarily in sequence) and does not have to be illuminated with a spherical wavefront. Instead of changing the orientation of a fold mirror to compensate for a fold mirror immediately before it, a fold mirror's orientation can be chosen to compensate for the overall system's aberrations.

5.2.1 Figure of Merit

Since the goal of using the fold mirrors is to compensate for diattenuation and retardance, the figure of merit will depend on the rotation angle of the fold

mirrors. The figure of merit selected is RMS diattenuation and retardance magnitude across the field of view. To calculate the diattenuation and retardance magnitudes from a Jones matrix, we first turn to the Pauli representation of a Jones matrix following Chipman's methodology[82].

$$\begin{aligned}
\mathbf{J} &= \begin{pmatrix} j_{xx} & j_{xy} \\ j_{yx} & j_{yy} \end{pmatrix} = c_0 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + c_1 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + c_2 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + c_3 \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\
&= c_0\sigma_0 + c_1\sigma_1 + c_2\sigma_2 + c_3\sigma_3 \\
&= c_0 \left(\sigma_0 + \frac{c_1}{c_0}\sigma_1 + \frac{c_2}{c_0}\sigma_2 + \frac{c_3}{c_0}\sigma_3 \right) \\
&= c_0 \left(\sigma_0 + f_1\sigma_1 + f_2\sigma_2 + f_3\sigma_3 \right)
\end{aligned} \tag{5.1}$$

c_0, c_1, c_2, and c_3 are the complex Pauli coefficients. The coefficient c_0 is related to the unpolarized amplitude and phase change, f_1 and f_2 are related to the linear diattenuation and retardance, and f_3 is related to the circular diattenuation and retardance. The c coefficients are directly calculated from the Jones matrix elements.

$$c_0 = \frac{j_{xx} + j_{yy}}{2}, \;\; c_1 = \frac{j_{xx} - j_{yy}}{2}, \;\; c_2 = \frac{j_{xy} + j_{yx}}{2}, \;\; c_3 = \frac{i(j_{xy} - j_{yx})}{2} \tag{5.2}$$

For weakly polarizing elements like mirrors (weak meaning the diattenuation and retardance magnitudes $<< 1$), the Jones matrix can be written as[82]

$$\mathbf{J} \approx \rho_0 e^{-i\phi_0} \left(\sigma_0 + \frac{D_H - i\delta_H}{2}\sigma_1 + \frac{D_45 - i\delta_45}{2}\sigma_2 + \frac{D_L - i\delta_L}{2}\sigma_3 \right) \tag{5.3}$$

D_H, D_{45}, and D_L are the horizontal, 45°, and circular diattenuations and δ_H, δ_{45}, and δ_L are the horizontal, 45°, and circular retardances. Using equations 5.1 and 5.3, for weak polarization effects the f-coefficients take on the form

$$f_1 = \frac{D_H - i\delta_H}{2}, \quad f_2 = \frac{D_{45} - i\delta_{45}}{2}, \quad f_3 = \frac{D_L - i\delta_L}{2} \tag{5.4}$$

So the real part of f_1 is half the linear horizontal/vertical diattenuation, the imaginary part of f_1 is half the linear horizontal/vertical retardance, the real part of f_2 is half the linear ±45° diattenuation, the imaginary part of f_2 is half the linear ±45° retardance, the real part of f_3 is half the circular diattenuation, and the imaginary part of f_3 is half the circular retardance.

To calculate the RMS diattenuation and retardance magnitudes, a grid of rays is traced through the optical system and the Jones matrix for each ray is calculated. Using that Jones matrix, the f-coefficients can be calculated using equations 5.1-5.4. Then the root mean square of the diattenuation and retardance magnitude for all rays in the field of view is calculated as

$$\Xi = 2\sqrt{\frac{1}{N}\sum_{n=1}^{N} |f_{1,n}|^2 + |f_{2,n}|^2 + |f_{3,n}|^2} \tag{5.5}$$

where $f_{1,n}$, $f_{2,n}$, and $f_{3,n}$ are the f-coefficients for the n^{th} ray in the field of view. Minimizing Ξ will minimize the distance of the Jones matrices in the field of view from an identity matrix, and thus minimize the overall polarization aberrations. If it is more important to specifically eliminate diattenuation

but not retardance or vice versa, Ξ can be modified by applying weighted coefficients to the real or imaginary parts of the f-coefficients in eq. 5.5.

5.3 Example Application of Rotating Surface Orientations

The HabEx optical system has five fold mirrors in its current design. Figure 5.1 shows the HabEx optical layout from the first fold mirror (fourth optical surface) until the fifth fold mirror (twelfth optical surface). The blue surfaces are the fold mirrors, the green surfaces are the front and back of the dichroic, and the red surfaces are OAPs. Collimated light enters this part of the system on the right side of the figure and proceeds through the system, reflecting a collimated beam off of the last fold mirror in the bottom left of the figure. The dichroic serves to divide the system into two channels. Shown here is the transmitted channel, for wavelegnths between 450 nm and 672 nm. Wavelengths longer than 673 nm will be deflected downwards from the green surface in the figure.

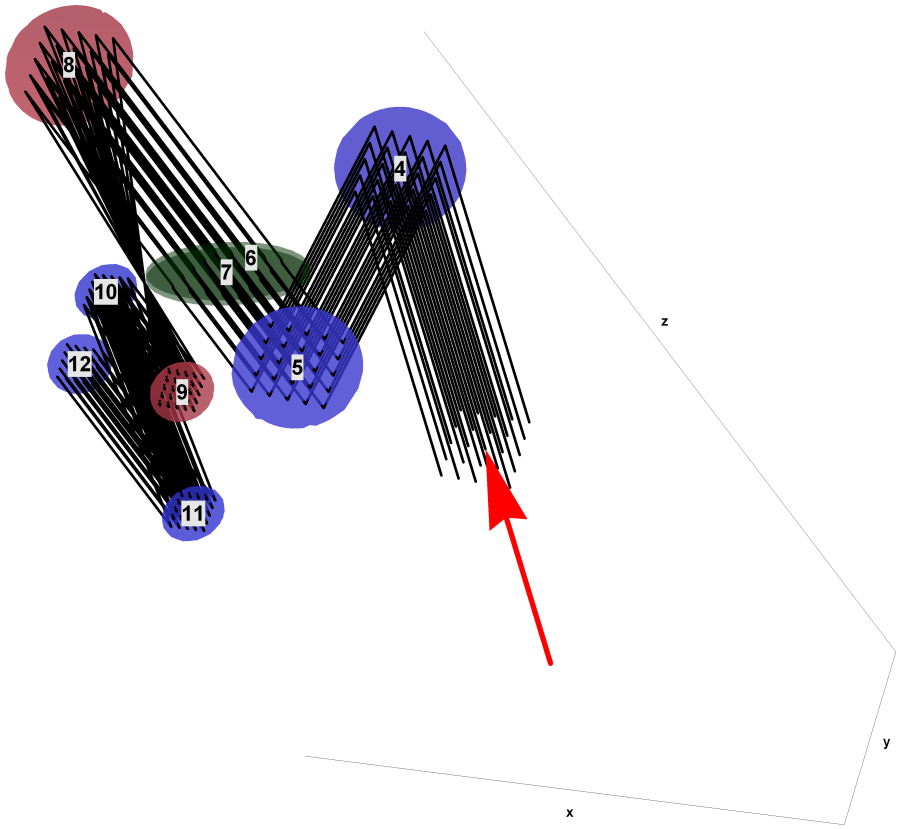


Figure 5.1: Original HabEx optical layout from first fold mirror (fourth optical surface) to fifth fold mirror (twelfth optical surface). The blue surfaces are the fold mirrors, the green surfaces are the front and back surfaces of the dichroic, and the red surfaces are OAPs. The red arrow in the lower right indicates the direction of the light arriving at the fourth mirror surface.

In this section, different surface orientation rotations will be applied to these fold mirrors to lower the polarization aberrations of the original system. It is important to note that while these rotations may lower the polarization aberrations, the compactness of the system is potentially lost. Thus what is reported here is an academic exercise and not a comment or suggestion for a better opto-mechanical system layout.

5.3.1 Rotating a Fold Mirror in HabEx

Before delving into the rotation of every fold mirror in the system simultaneously, we'll start by looking at the effect of rotating each fold mirror individually. The approach taken here is to leave the angle of incidence of each ray at a surface the same when applying these rotations. To do so, the first step is to extract the propagation vector incident on the fold mirror's surface from a ray trace. It is helpful that all the fold mirrors in HabEx are in collimated space, so the propagation vector of any of the rays will suffice for this calculation. Once the incident propagation vector is known, two rotation transforms are calculated: one transform gives the three dimensional rotation around the incident propagation vector as a function of rotation angle, the second transform gives the three dimensional rotation around the incident propagation vector anchored around a given vertex. The first rotation transform is then applied to the fold mirror's surface normal to produce the rotated surface normal. Similarly, the second rotation transform is applied to the fold mirror's surface vertex to produce the rotated surface vertex.

Rotating a surface upstream of any optical element will affect the vertex and surface normal of every optic downstream. These rotation transforms also have to be applied to every surface after the rotated mirror's surface. Once these transforms have been applied, the updated system with its rotated fold mirror is then polarization ray traced, and Ξ can be calculated. Table 5.1 outlines the step by step process to updating the optical system with rotated fold mirrors.

1. Ray trace original system
2. Extract propagation vector incident on fold mirror of interest
3. Create rotation transform that rotates about propagation vector, R_1
4. Apply R_1 to fold mirror's surface normal and the surface normal of every mirror after the fold mirror
5. Create rotation transform that rotates about propagation vector relative to an anchor vertex, R_2
6. Apply R_2 to fold mirror's surface vertex and the vertex of every mirror after the fold mirror
7. Repeat 1-6 for every fold mirror that is rotated
8. Polarization ray trace rotated system and calculate Ξ

Table 5.1: Steps for calculating Ξ as a function of fold mirror rotations

Figure 5.2 shows the calculated value of Ξ as each of the five fold mirrors are individually rotated through 360°. As the rotation angle changes, Ξ tends to change sinusoidally. This is to be expected because the axis of a polarization element (max transmission for polarizer or fast axis for retarder) is modulo 180°, meaning that rotating them by steps of 180° should be the same as not rotating them at all. Since rotating these fold mirrors essentially rotates the s-orientation and p-orientation of the surface, rotating those orientations by 180° should have very little effect. Immediately, we can note that the original system, which will have all five rotation angles for these mirrors be at 0°, is not at a minimum for Ξ. While rotating the orientations of the first two fold mirrors (the fourth and fifth surfaces in the layout) away from 0° actually increases our merit function, rotating any of the other three fold mirrors (surfaces 10, 11, and 12 in the optical layout) away from 0° serves to reduce Ξ.

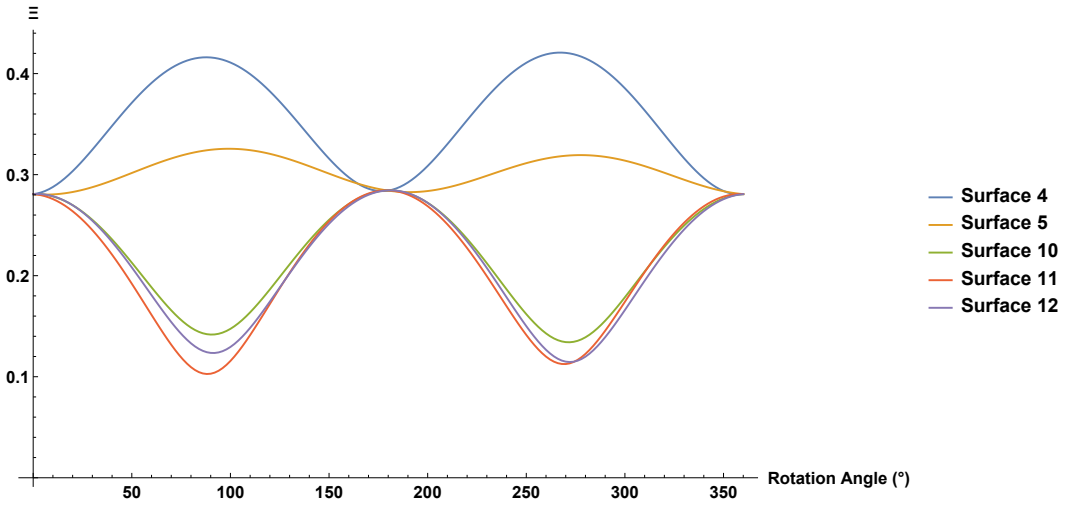


Figure 5.2: Plot of Ξ versus rotation angle, in degrees, as each of the five fold mirrors are indiviudally rotated.

So now that we know that our merit function can be reduced, the question becomes (1) what is the minimum value we can reduce it to and (2) what is the best approach for finding that minimum?

5.3.2 Rotating All Five Fold Mirrors in HabEx

We know from figure 5.2 that the original set of orientations for the five fold mirrors does not sit at a minimum for Ξ. Simply by rotating the fourth fold mirror (the eleventh surface) by about 90°, Ξ drops from 0.28 down to almost 0.1. Since a rotation angle of about 90° is also near a minimum for the third and fifth fold mirrors (surfaces 10 and 12), one might assume that Ξ could be further reduced by also rotating those fold mirrors by 90°. The problem, however, is that the relationship between the five fold mirror angles and Ξ is complex, and changing any one angle will affect the influence of changing the other four angles. Thus simply using the angles that show minima in figure 5.2 will not produce a global minimum for Ξ.

Since Polaris-M is built inside of Mathematica[83], minimization of the

merit function can be implemented with Mathematica's built in optimization routines[84]. By defining a function that calculates Ξ from a Polaris-M ray-trace, Mathematica's NMinize function was implemented to calculate the five mirror orientations that minimized Ξ. This minimization was done at only one wavelength, 500 nm. Additional optimization could be implemented by accounting for changes in the diattenuation and retardance across wavelength, but only 500 nm was used for this round of optimization. The optimization routine and code used in the optimization process can be found in Appendix B.

5.3.3 Original System Vs Optimized System

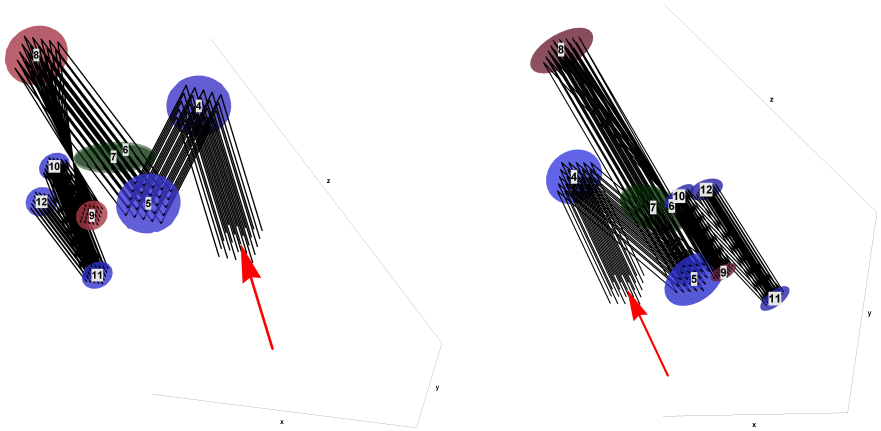
The outputs from the minimization routine compared to the original system orientation are detailed in table 5.2. The angles in the table are relative rotations about the surface normal from the original system's layout, so the original system's angles are all zero.

	Original	Optimized
Ξ	0.2807199	0.0104598
θ_4	0	182.032°
θ_5	0	107.936°
θ_{10}	0	187.287°
θ_{11}	0	348.068°
θ_{12}	0	65.3966°

Table 5.2: Ξ for the original system and the optimized system

Figure 5.3 shows the original HabEx system layout and the rotated system layout that has been optimized to minimize for polarization aberrations. The 180° rotation of the first fold mirror results in the input collimated beam coming in from the left side of the figure (figure 5.3b. The two figures are shown from

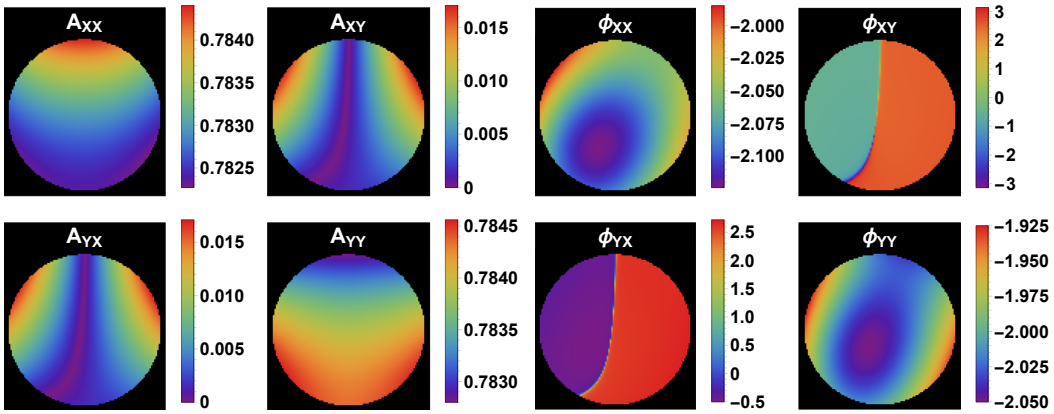
the same view points in global space.



(a) (b)

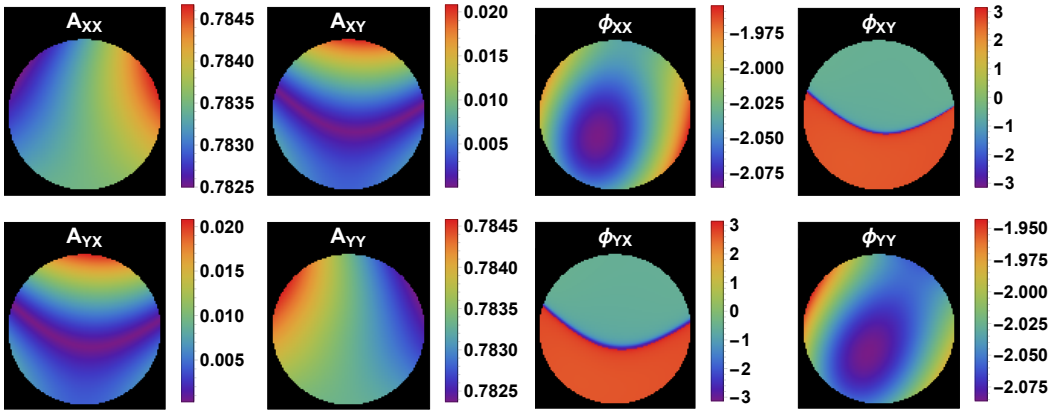
Figure 5.3: Original HabEx optical layout (a) and rotated HabEx optical layout (b) from first fold mirror (fourth optical surface) to fifth fold mirror (twelfth optical surface). The blue surfaces are the fold mirrors, the green surfaces are the front and back surfaces of the dichroic, and the red surfaces are OAPs. The red arrow in the lower right of (a) and lower left of (b) indicate the direction of the light arriving at the fourth mirror surface.

Figures 5.4 and 5.5 show the entire system Jones pupils for the original HabEx layout and the compensated layout using the fold mirror rotation angles shown in table 5.2. The left side of the figures shows the amplitude of the pupil and the right side shows the phase in radians of the pupil.



(a) (b)

Figure 5.4: Magnitude (a) and phase in radians (b) of the original HabEx optical system for $\lambda = 500$nm



(a) (b)

Figure 5.5: Magnitude (a) and phase in radians (b) of the compensated HabEx optical system for $\lambda = 500$nm

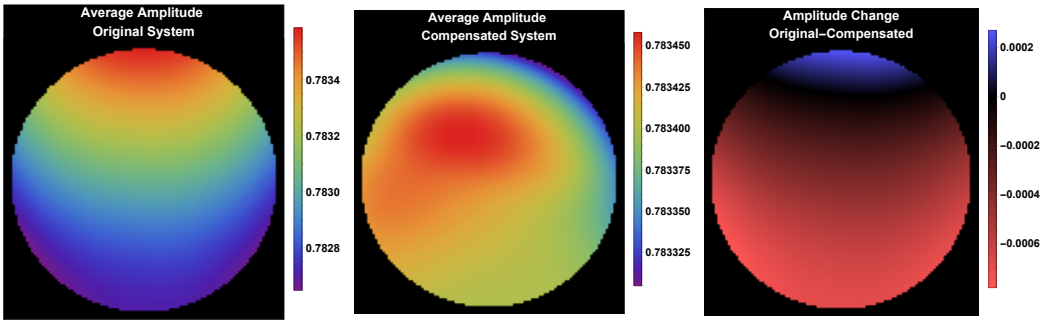
The most immediate difference between the original layout's Jones pupil and the compensated layout's Jones pupil is the structure of the amplitude and phase. The zero that runs through the off-diagonal amplitudes and corresponding π phase jump in the off-diagonal phases have rotated by 90°. The distribution of the on-diagonal amplitudes are also rotated by 90°. However, by design, we kept the angles of incidence on the fold mirrors the same between the original system and the compensated system, so the amplitude of the on-diagonals only change by a fraction of a percent. The on-diagonal phases are relatively similar, with an overall piston term being the most notable change. From eq. 5.3, the Jones pupils shown in figures 5.4 and 5.5 can be decomposed into a non-polarizing term, c_0, a term that encompasses the diattenuation, and a term that encompasses the retardance. The non-polarizing term can be split into the non-polarizing amplitude, ρ_0, and non-polarizing phase or wavefront, $e^{-i\phi_0}$. The diattenuation (D) and retardance (δ) magnitudes are

$$D = \sqrt{D_H^2 + D_{45}^2 + D_L^2}, \quad \delta = \sqrt{\delta_H^2 + \delta_{45}^2 + \delta_L^2} \tag{5.6}$$

Using the relationships in eq. 5.4, eq. 5.6 can be rewritten as

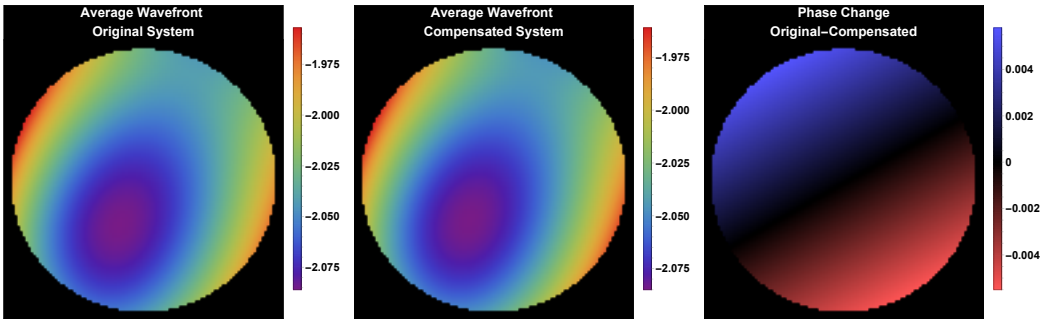
$$D = 2\sqrt{\mathscr{R}\{f_1\}^2 + \mathscr{R}\{f_2\}^2 + \mathscr{R}\{f_3\}^2}, \quad \delta = 2\sqrt{\mathscr{I}\{f_1\}^2 + \mathscr{I}\{f_2\}^2 + \mathscr{I}\{f_3\}^2} \tag{5.7}$$

where $\mathscr{R}\{f\}$ takes the real part of f and $\mathscr{I}\{f\}$ takes the imaginary part of f. Figures 5.6 and 5.7 show the unpolarized amplitude, ρ_0, and unpolarized phase, ϕ_0, for the original system, compensated system, and the difference between the two.



(a) (b) (c)

Figure 5.6: Unpolarized magnitude for the original system (a) and the compensated system (b) for $\lambda = 500$nm. The difference between the two is shown in (c)

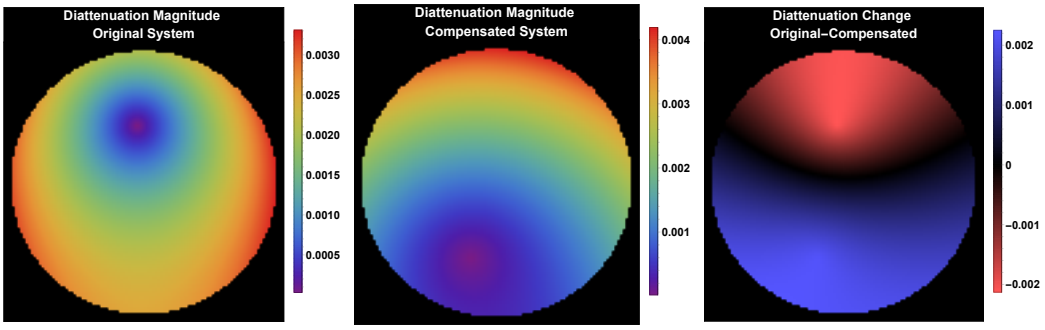


(a) (b) (c)

Figure 5.7: Unpolarized wavefront for the original system (a) and the compensated system (b) in radians for $\lambda = 500$nm. The difference between the two is shown in (c)

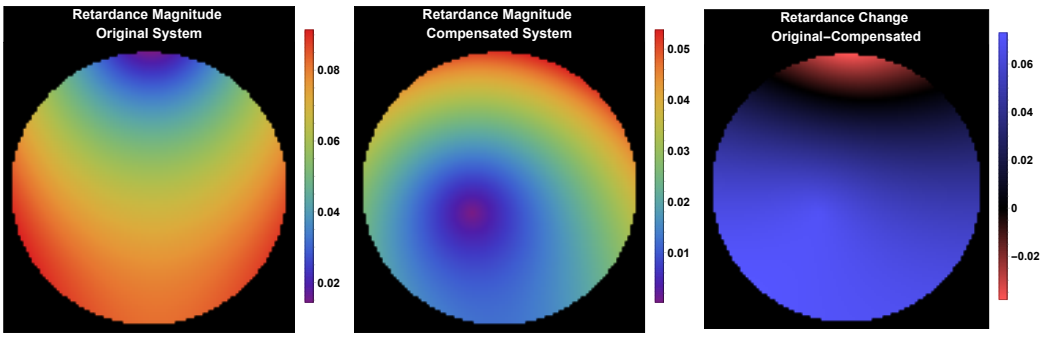
The differences in the unpolarized amplitude and phase between the original

system and the compensated system are relatively small, on the order of hundreths of a percent for the amplitude and tenths of a percent for the phase. Since the purpose of the compensating system is to compensate for polarization aberration, it is to be expected that non-polarized wavefront and apodization does not change appreciably between the two configurations. The diattenuation and retardance magnitudes, on the other hand, should vary appreciably between the two configurations. Figures 5.8 and 5.9 show the diattenuation and retardance magnitudes of the two configurations, as well as their differences.



(a) (b) (c)

Figure 5.8: Diattenuation magnitude for the original system (a) and the compensated system (b) for $\lambda = 500$nm. The difference between the two is shown in (c)



(a) (b) (c)

Figure 5.9: Retardance magnitude for the original system (a) and the compensated system (b) in radians for $\lambda = 500$nm. The difference between the two is shown in (c)

The maximum change in the diattenuation magnitude relative to the original

configuration is about 26%. The difference shown in figure 5.8 is symmetric about zero. The red upper portion indicates that that region of the pupil has more diattenuation in the compensated configuration. The blue central and lower portion indicates that that region of the pupil has less diattenuation in the compensated configuration. Overall, there is a greater portion of the pupil that has less diattenuation, resulting in an overall decrease in the average diattenuation across the pupil.

The maximum change in the retardance magnitude relative to the original configuration is about 41%. The difference shown in figure 5.9 indicates that much of the pupil has less retardance, as indicated by the blue portion of the pupil. There is only a small portion at the top of the pupil that has more retardance in the compensated configuration. There is an overall decrease in the average retardance across the pupil.

5.4 Closing Remarks

Metal coated mirrors and dielectric thin film coated reflecting surfaces cause polarization aberrations due to the variations of the Fresnel reflection coefficients with angle of incidence. By choosing the appropriate configuration for the fold mirrors, it is possible to compensate some of the polarization aberrations of an optical system. The proper choice of configuration depends on the merit function to be evaluated. The chosen merit function in this discussion was RMS diattenuation and retardance magnitude. These magnitudes can be easily calculated for most reflecting systems using the weak polarizing assumption. Minimizing this merit function is equivalent to minimizing polarization

aberrations in the system.

The optimized rotation angles about the surface normals for the five flat fold mirrors of the HabEx optical system were $\theta_4 = 182.032°$, $\theta_5 = 107.936°$, $\theta_{10} = 187.287°=$, $\theta_{11} = 348.068°$, and $\theta_{12} = 65.3966°$. With these fold mirror rotations, the average retardance of the system was reduced by a factor of 3 while the average diattenuation of the system was reduced by a factor of 20%. The unpolarized wavefront and apodization did not change appreciably. However, with these rotation angles, the compactness of the original system is lost. This analysis is not meant to be a critic on the current HabEx design, but is rather an academic exercise to show proof of concept and carry the design one step further.

CHAPTER 6

Longitudinal Chromatic Aberration

6.1 Introduction

In an ideal polychromatic imaging system, spherical wavefronts from the exit pupil converge to a point in the image plane [85, 86]. Misaligned components, spectral refractive index variations, and the geometry of the surfaces of an optical system lead to aberrations, which are departures from these spherical wavefronts [49]. This chapter is focused on longitudinal chromatic aberration and its relation to the wavefront aberration defocus, a quadratic departure from a spherical wavefront.

Longitudinal chromatic aberration is the change in effective focal length of an image forming system with wavelength [87, 88, 89]. In refracting systems, the wavelength dependence of the angle of refraction from Snell's law causes the focal point for different wavelengths to be separated. In reflecting systems, however, the law of reflection is wavelength independent. When scalar ray tracing analyzes a reflecting system, the focal point for all wavelengths coincide because all ray paths will have the same optical path length. But with polarization ray tracing, focal shifts are observed for paraxial rays. This is

also understood from using the small angle approximation for the reflected phase change of a coated interface, which is quadratic with angle of incidence. Thus wavefronts of different wavelengths at the exit pupil acquire different quadratic phases from dielectric coated reflecting surfaces[90]. In the paraxial regime, these phase differences in a reflecting system across wavelength are *longitudinal chromatic aberrations*. They degrade the image quality in polychromatic imaging systems and can cause unforeseen challenges in broadband optical systems that need precision alignment. What follows is a method for describing and calculating the longitudinal chromatic aberrations induced by dielectric coated mirrors.

Section 6.2 defines the wavefront aberration defocus and highlights the difference between how conventional scalar ray tracing software and polarization ray tracing software represent aberrations. Section 6.3 describes the calculation of longitudinal chromatic aberration from reflections off of a bare metal surface and dielectric coated metal surface. Section 6.4 discusses the physical shifts in focus that arise as a consequence of the longitudinal chromatic aberration. Section 6.5 provides our conclusions.

6.2 Aberration Representation

In practice, conventional ray tracing uses geometric rays to model propagation of the electric field of light. The phase change of the light along a ray segment is calculated from its optical path length (OPL),

$$\phi = \frac{2\pi OPL}{\lambda}, \quad OPL = nd, \tag{6.1}$$

where λ is the wavelength of the ray, n is the index of refraction of the media the ray is propagating through, and d is the physical distance the ray travels in that medium. Since wavefront aberrations are deviations of the phase fronts relative to a reference sphere, conventional ray tracing uses optical path difference (OPD) as the primary means of computing and displaying the wavefront aberration function, $W(x, y, \lambda)$ [72]. The OPD for a ray is defined as the difference between that ray's optical path length and the chief ray's optical path length, where the optical path length is measured from the source to a reference sphere at the exit pupil[91].

This scalar wavefront aberration function *does not* encompass polarization effects (such as the Fresnel reflection coefficients for s- and p-polarized light) in an optical system. To capture these effects, polarization ray tracing is used to calculate the polarization aberration function, an eight-valued function written as a spatially varying Jones matrix [92].

$$\mathbf{JP}(x, y, \lambda) = \begin{bmatrix} J_{xx}(x, y, \lambda) & J_{xy}(x, y, \lambda) \\ J_{yx}(x, y, \lambda) & J_{yy}(x, y, \lambda) \end{bmatrix} \tag{6.2}$$

JP is eight-valued because all four components ($J_{xx}, J_{xy}, J_{yx}, J_{yy}$) are complex valued. This Jones matrix can be decomposed into four components: the wavefront aberration, W, from the coatings *and* ray OPDs, the amplitude transmission, A, and diattenuation and retardance components in the form of a Jones matrix, **J**. The wavefront aberration and amplitude transmission are scalar terms.

$$\mathbf{JP}(x, y, \lambda) = A(x, y, \lambda)e^{i2\pi W(x,y,\lambda)/\lambda}\mathbf{J}(x, y, \lambda) \tag{6.3}$$

The amplitude transmission, A, is calculated from the average magnitude of J_{xx} and J_{yy} while the wavefront aberration, W, is calculated from the average phase of J_{xx} and J_{yy}. **J** can be further broken down into linear horizontal, linear 45°, and circular diattenuation and retardance terms or diattenuation and retardance piston, tilt, and defocus terms[18, 93, 94, 95, 96]. The two scalar terms, A and W, have been separated from the Jones matrix term, **J**, to emphasize the unpolarized amplitude and wavefront terms at the exit pupil, but in practice the spatially varying Jones matrix of eq. 6.2 is given by a polarization ray trace and A and W must be calculated from J_{xx} and J_{yy}. An example calculation of A, W, and **J** utilizing a polarization ray trace is provided in Appendix C.

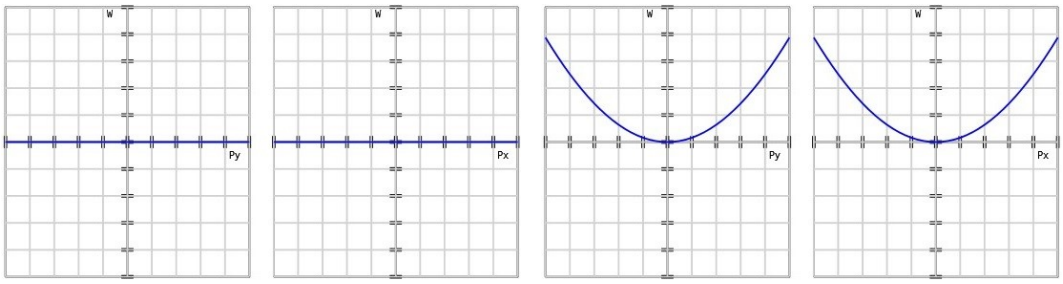
6.2.0.1 An Aside About Coating-Induced Astigmatism

In the paraxial regime, the Jones matrix in eq. 6.3 is referred to as the linear retardance defocus term [90, 92, 97], and it is responsible for coating-induced astigmatism. The magnitude of this coating-induced astigmatism is proportional to the difference between the s- and p-reflected phase changes and can be seen as quadratic retardance across the pupil (thus the name retardance defocus). An example calculation of the astigmatism seen from the linear retardance defocus term is provided in Appendix C.

6.2.1 Defocus Representation

Defocus is a longitudinal aberration, causing shifts in the focal point of the system along the optical axis [89, 98]. Aberrated wavefronts with defocus take on the the form of a quadratic relative to the reference wavefront. For scalar ray tracing, this quadratic shows up in the OPD of rays traced through the system. For polarization ray tracing, this quadratic shows up in the exponential term in eq. 6.3.

For example, figure 6.1 shows two different OPD plots generated from a scalar ray trace. The prescription of the parabolic mirror used to create these plots can be found in Appendix C. The left side figure shows the OPD plot when the image plane is at paraxial focus. The right side of the figure shows the same system with the image plane moved by 0.1 mm along the optical axis towards the mirror. These OPD plots show the relative pupil position along the x-axis and the wavefront aberration in waves along the y-axis. The example system is axially symmetric, so the plots as a function of relative x-position and relative y-position in the pupil are the same. The vertical scale the left side of figure 6.1 is 10^{-5} waves while the vertical scale on the right side is 20 waves.



(a) (b)

Figure 6.1: OPD plots for (a) a system with no defocus and (b) the same system with defocus. W is the amount of aberration, in waves. Py and Px are scaled coordinates in the entrance pupil along the y-axis and x-axis, respectively. The range of the plot for (a) is 10^{-5} waves and the range of the plot for (b) is 20 waves.

For an optical system free of geometric aberrations, the optical path length for every ray is identical, so the OPD for each ray is zero, as shown in figure 6.1a. When geometric defocus is introduced into the system by shifting the image plane towards the mirror, the OPD of rays changes quadratically away from the center of the pupil, as shown in figure 6.1b.

Figure 6.2 shows a plot similar to the OPD plots in figure 6.1a, except this time, the average reflected phase was calculated from a polarization ray trace [21]. Polarization ray tracing incorporates the effects of geometric phase (OPD) *and* polarization dependent phase (Fresnel/amplitude reflection coefficients). Figure 6.2 shows a quadratic change across the pupil. Since the OPDs of the rays have not changed, this quadratic phase is a direct result of the polarization properties of the coating. This quadratic changes as the wavelength changes, leading to chromatic aberration. Thus, longitudinal chromatic aberration in a reflecting system appears when the wavefront at the exit pupil displays a quadratic departure from a spherical wavefront due to the OPDs *and* amplitude reflection coefficients that vary with wavelength.

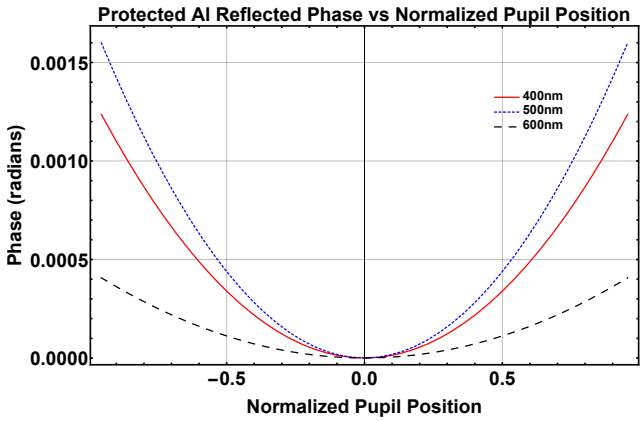


Figure 6.2: Example of quadratic phase for a dielectric coated reflecting surface. The same parameters were used for this system that created the OPD plot shown in figure 6.1a, except this system was polarization ray traced. The geometric aberrations for this surface are zero, but the amplitude reflection coefficients impart varying phase across the surface, causing defocus. The change of the quadratic across wavelength is chromatic aberration from this dielectric coated reflecting surface. The prescription and calculation process for this reflected phase are provided in Appendix C.

Light rays reflecting off of surfaces obey the law of reflection. The angle of reflection is the same as the angle of incidence [99]. Since the angle of reflection is *not* wavelength dependent, an OPD plot for a single reflecting surface will be invariant across wavelength. It is for this reason that it is generally assumed that reflecting optical systems display no longitudinal chromatic aberration; the geometric ray paths should be the same for all wavelengths. However, the wavefront at the exit pupil of a system is determined by the OPD *and* any phases due to reflection, which are described by the Fresnel reflection coefficients (for bare surfaces) or amplitude reflection coefficients (for thin film coated surfaces)[90]. The amplitude reflection coefficients are expressed as:

$$\alpha_s(\lambda,\theta) = \xi_s(\lambda,\theta)e^{i\phi_s(\lambda,\theta)} \tag{6.4}$$

$$\alpha_p(\lambda,\theta) = \xi_p(\lambda,\theta)e^{i\phi_p(\lambda,\theta)} \tag{6.5}$$

where ξ_s and ξ_p are the amplitude coefficients on reflection from a thin film for the s- and p-polarization states and ϕ_s and ϕ_p are the phase changes on reflection from a thin film for the s- and p-polarization states. Both ξ and ϕ are dependent on the wavelength (λ) and angle of incidence (θ). These amplitude coefficients encompass both amplitude and phase change after reflection from a thin film. From polarization aberration theory (eq. 6.3), the wavefront aberration function is the average of the s- and p-polarized phase changes from a reflecting surface. So while the geometrical optical path lengths from a ray trace may be identical, the complex Fresnel or amplitude reflection coefficients do contribute phase changes to the wavefront.

6.3 Chromatic Aberrations from a Metal-Coated Mirror

Chromatic aberration in a reflecting imaging system is the wavelength variation of defocus at the exit pupil from a fixed reference spherical wavefront. Such a defocused wavefront is created when a collimated beam of light interacts with a curved reflecting surface, like a spherical or parabolic primary mirror, or when a converging beam of light interacts with a flat reflecting surface. An approximately linear variation in the angle of incidence across the reflecting surface leads to defocus. Non-linear variations in the angle of incidence across

the surface lead to higher order aberrations that are beyond the scope of this paper. Flat reflecting surfaces with an incident collimated beam will only contribute a piston term because the angle of incidence for every ray on the mirror will be the same.

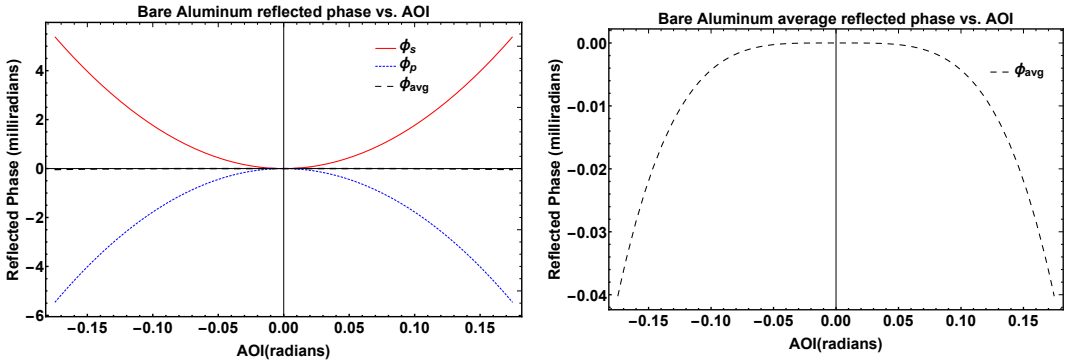
This section (1) shows that bare reflecting curved surfaces do not contribute longitudinal chromatic aberration, (2) details how the amplitude reflection coefficients from a dielectric coated curved reflecting surface give rise to defocus aberration, and (3) shows how the quadratic coefficient for the wavefront reflecting from a surface is calculated.

6.3.1 Bare Metal Reflecting Surfaces

The Fresnel reflection equations are used to calculate the reflected phase from a metal surface. Consider an incident medium with a purely real index of refraction, n_1, a reflecting substrate with complex index of refraction, $n_2 - ik_2$, and angle of incidence, θ. The complex reflection coefficients for s-polarized and p-polarized light are given by [49]:

$$r_s = \frac{n_1 \cos\theta - \sqrt{(-ik_2 + n_2)^2 - n_1^2 \sin\theta^2}}{n_1 \cos\theta + \sqrt{(-ik_2 + n_2)^2 - n_1^2 \sin\theta^2}} \tag{6.6}$$

$$r_p = \frac{(-ik_2 + n_2)^2 \cos\theta - n_1\sqrt{(-ik_2 + n_2)^2 - n_1^2 \sin\theta^2}}{(-ik_2 + n_2)^2 \cos\theta + n_1\sqrt{(-ik_2 + n_2)^2 - n_1^2 \sin\theta^2}} \tag{6.7}$$



(a) (b)

Figure 6.3: Reflected phase from aluminum as a function of angle of incidence at $\lambda = 500$ nm, n = 0.666 - 5.57i. (a) shows the s-polarized phase, p-polarized phase, and average of s- and p-polarized phases. (b) zooms in on the average of the s- and p-polarized phases, showing the absence of a quadratic term but presence of higher even ordered terms.

It is important to note that the Fresnel equations, eqs. 6.6 and 6.7 are defined for a right-handed coordinate system [49] such that, upon reflection, the p-polarized reflection coefficient is given a negative sign due to the change of basis (p-direction flips after reflection). This will show up as a π phase shift that is not related to the actual reflected phase. In figure 6.3a, the reflected phases are *not* π apart because this phase change on reflection has been accounted for.

Figure 6.3a shows the phase of the Fresnel reflection coefficients for s-polarized light, p-polarized light, and the average of s- and p-polarized light reflecting off of aluminum at 500 nm (n = 0.666 - 5.57i) as a function of angle of incidence. Figure 6.3b is a zoomed in view of the average reflected phase. While the s- and p-polarized phases both have significant quadratic terms of opposite sign, the average reflected phase has a fourth and higher even order terms but no quadratic contribution. To prove there is no quadratic term in the average reflected phase, eq. 6.6 and eq. 6.7 are expanded in a Taylor series to second

order around the origin.

$$r_s \approx -1 + \frac{2ik_2n_1 + 2n_1^2 + 2n_1n_2}{k_2^2 + (n_1+n_2)^2} + \left(\frac{ik_2n_1 + n_1n_2}{k_2^2 + n_2^2} - \frac{2ik_2n_1 + 2n_1^2 + 2n_1n_2}{k_2^2 + (n_1+n_2)^2} \right) \theta^2 \tag{6.8}$$

$$r_p \approx 1 - \frac{2ik_2n_1 + 2n_1^2 + 2n_1n_2}{k_2^2 + (n_1+n_2)^2} + \left(\frac{ik_2n_1 + n_1n_2}{k_2^2 + n_2^2} - \frac{2ik_2n_1 + 2n_1^2 + 2n_1n_2}{k_2^2 + (n_1+n_2)^2} \right) \theta^2 \tag{6.9}$$

To calculate the phase change on reflection, the arctangent of the ratio of the imaginary part to the real part of the reflection coefficient, η, is calculated. To first order, the Taylor series of the arctangent function is

$$ArcTan[\eta] \approx \eta \tag{6.10}$$

The ratio of imaginary part to real part of the s- and p-reflection coefficients are

$$\eta_s = \frac{Im[r_s]}{Re[r_s]} = \frac{k_2n_1((-2+\theta^2)k_2^2 - \theta^2n_1^2 - 2\theta^2n_1n_2 + (-2+\theta^2)n_2^2)}{k_2^4 + n_2(\theta^2n_1 + n_2)(-n_1^2 + n_2^2) + k_2^2((-1+2\theta^2)n_1^2 + \theta^2n_1n_2 + 2n_2^2)} \tag{6.11}$$

$$\eta_p = \frac{Im[r_p]}{Re[r_p]} = -\frac{k_2n_1((2+\theta^2)k_2^2 - \theta^2n_1^2 - 2\theta^2n_1n_2 + (2+\theta^2)n_2^2)}{k_2^4 - k_2^2((1+2\theta^2)n_1^2 + \theta^2n_1n_2 + 2n_2^2) + \theta^2n_1 - n_2)n_2(n_1^2 - n_2^2)} \tag{6.12}$$

Expanding eq. 6.11 and eq. 6.12 in a Taylor series to second order yields

$$\phi_s \approx -\frac{2k_2n_1}{k_2^2 - n_1^2 + n_2^2} + \frac{k_2n_1(k_2^4 + 2k_2^2 + n_1^4 + 2k_2^2n_2^2 - 2n_1^2n_2^2 + n_2^4)}{(k_2^2 + n_2^2)(k_2^2 - n_1^2 + n_2^2)^2}\theta^2 \quad (6.13)$$

$$\phi_p \approx -\frac{2k_2n_1}{k_2^2 - n_1^2 + n_2^2} - \frac{k_2n_1(k_2^4 + 2k_2^2 + n_1^4 + 2k_2^2n_2^2 - 2n_1^2n_2^2 + n_2^4)}{(k_2^2 + n_2^2)(k_2^2 - n_1^2 + n_2^2)^2}\theta^2 \quad (6.14)$$

The coefficient of the quadratic term in eq. 6.13 and eq. 6.14 are equal in magnitude but opposite in sign and will cancel with one another when added together. Taking the average of eq. 6.13 and eq. 6.14, the average reflected phase to second order is

$$\boxed{\phi_{avg} = \frac{\phi_s + \phi_p}{2} \approx -\frac{2k_2n_1}{k_2^2 - n_1^2 + n_2^2}} \quad (6.15)$$

This demonstrates the important result that there is no angle dependent quadratic component to the average reflected phase. Thus there is no defocus imparted by bare reflecting surfaces. With no defocus, there is no chromatic change of focus from bare metals. *Thus, bare reflecting surfaces show no longitudinal chromatic aberration.*

6.3.2 Dielectric Thin Film Coatings on Metal-Coated Surfaces

While uncoated reflecting surfaces do not generate longitudinal chromatic aberration, dielectric coated reflecting surfaces generate longitudinal chromatic aberration. Figures 6.4a, 6.4b, and 6.4c show a few examples of the reflected phase from multilayer coatings versus angle of incidence, calculated at a wavelength of 500 nm from a polarization ray trace. These example multilayer

coatings are: (1) Macleod 188, a five layer dielectric stack[100], (2) FSS99-600, an enhanced reflection coating on silver for use in space-based telescopes like HabEx [67], and (3) protected aluminum which uses enhanced reflection coatings of three separate quarter wave stacks of hafnium oxide and silicon dioxide[101]. The prescription of the Macleod 188 stack assumes no dispersion in each material (index of refraction is constant across wavelength). These coating prescriptions are provided in Appendix C.

The defocus contribution of a parabolic or spherical mirror and its coating in the paraxial regime can be seen as the quadratic phase imparted by the reflecting surface. As opposed to the average reflected phase from bare aluminum (figure 6.3) which had the quadratic contribution from s- and p- cancel one another, these three multilayer thin films all show average reflected phases with a non-zero quadratic phase.

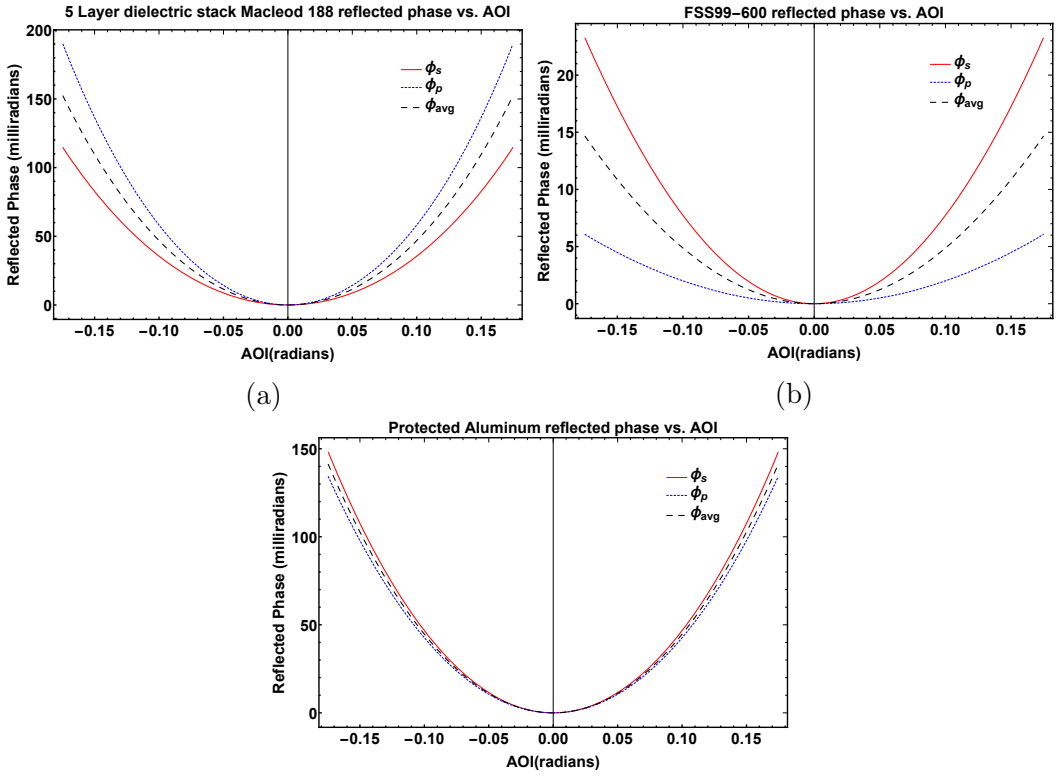


Figure 6.4: Reflected phase at 500 nm for (a) Macleod 188, (b) FSS99-600, and (c) protected aluminum multilayer thin film coatings. The solid red curve is the s-polarized reflected phase, the dotted blue curve is the p-polarized reflected phase, and the dashed black curve is the average of s- and p-polarized reflections. Across this range of angles of incidence, both s- and p-polarized phases are predominately quadratic which contributes defocus to the wavefront.

We define the quadratic coefficient of the reflected phase from a multilayer coating as:

$$\Phi_2(\lambda) = \lim_{\theta_0 \to 0} \frac{\frac{\phi_s(\theta_0;\lambda)+\phi_p(\theta_0;\lambda)}{2} - \phi(0;\lambda)}{\theta_0^2} \tag{6.16}$$

Φ_2 is the calculated quadratic coefficient, θ_0 is the angle of incidence, $\phi_s(\theta_0)$ and $\phi_p(\theta_0)$ are the s-polarized and p-polarized reflected phases at θ_0, and $\phi(0)$ is the reflected phase at normal incidence. In practice, Φ_2 is evaluated as a finite difference by replacing the limit in eq. 6.16 with a small angle for θ_0.

The defocus magnitude in radians at each wavelength is calculated as the av-

erage phase change that the reflecting surface induces on the marginal ray[90],

$$\Delta W_{defocus,radians} = \Phi_2 \theta_{marg}^2 \tag{6.17}$$

The chromatic aberration for a dielectric coated reflecting surface is seen as variations in the quadratic coefficient with wavelength. Using eq. 6.16 with a small angle of incidence, $\theta_0 = 0.1$, the quadratic coefficient is plotted against wavelength in figures 6.5a-6.5c. Larger fluctuations in the quadratic coefficient indicate more chromatic aberration while smaller changes in the quadratic coefficient across wavelength indicate less chromatic aberration. Of particular interest in these plots of quadratic coefficients are where the areas of relatively small changes in the quadratics occur. For Macleod 188, this region occurs for wavelengths shorter than 500 nm and between 550 nm and 750 nm, while for FSS99, the variations are very small after about 450 nm. For the protected aluminum, this region occurs between about 550 nm and 750 nm. These regions line up with the regions of highest reflectivity for each coating, shown in figures 6.6a-6.6c.

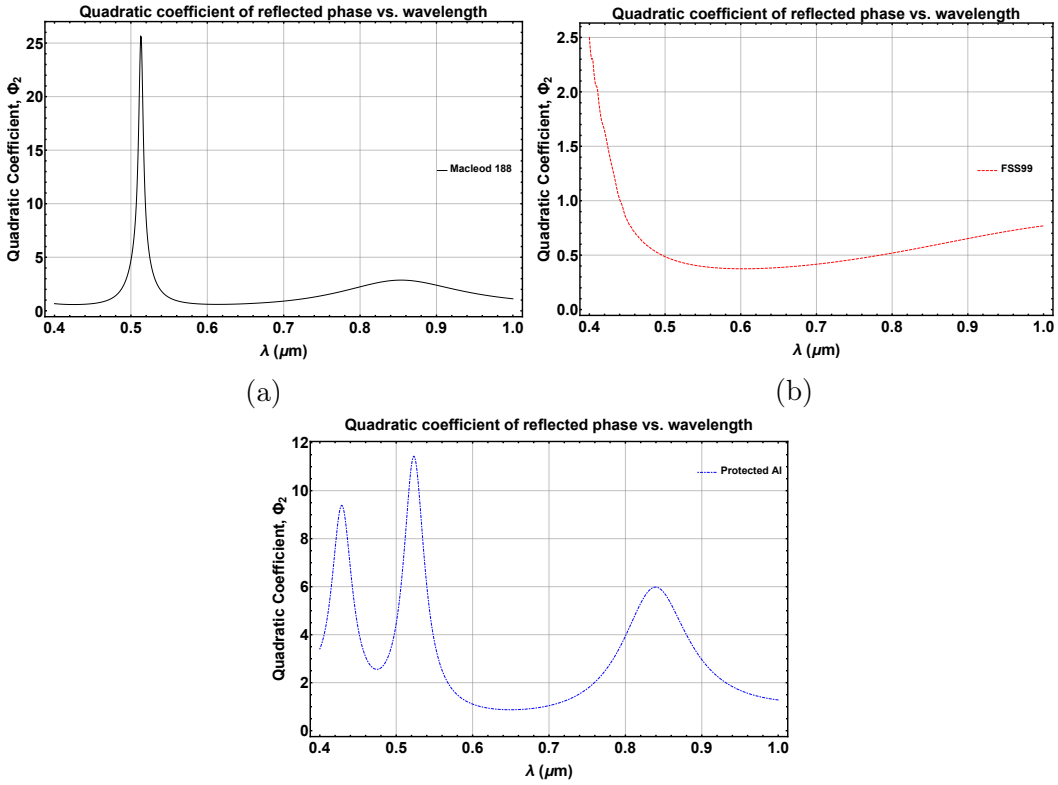


Figure 6.5: Quadratic coefficients of the average reflected phase from (a) Macleod 188, (b) FSS99-600, and (c) protected aluminum versus wavelength.

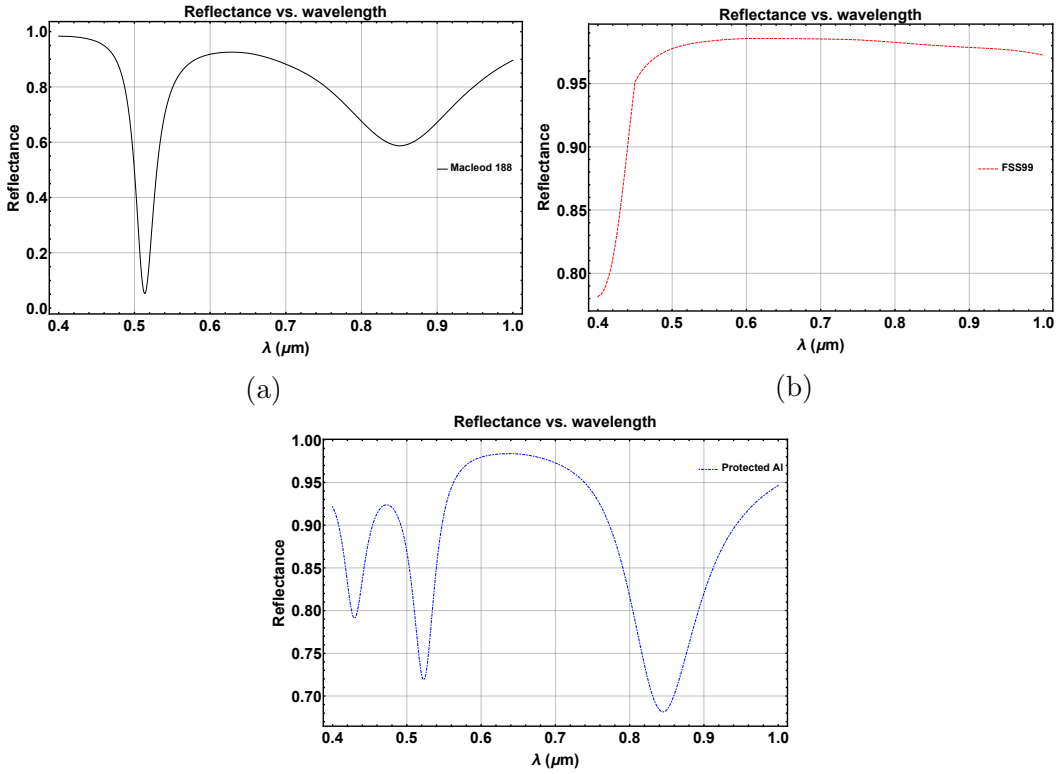


Figure 6.6: Normal incidence reflectance of (a) Macleod 188, (b) FSS99-600, and (c) protected aluminum versus wavelength. Regions of high reflectance correspond to regions of small changes in the quadratic coefficient.

The wavelength regions of lower chromatic aberration are observed to be the same as the regions of higher reflectivity for these thin films. This is fortunate for thin film designers because thin films designed for high reflection will naturally have low chromatic aberration in the highly reflecting regions. But many optical systems are illuminated with wavelengths not in the highly reflecting band, and operating at these wavelengths will lead to larger amounts of chromatic aberration. This is because in regions of low reflectance, the constructive and destructive interference between the partial waves changes rapidly with angle, and the phase changes (and thus coating induced aberrations) tend to be larger[102].

6.4 Physical Shifts from Defocus

The longitudinal chromatic aberration of these coatings will cause a shift in the focal point as a function of wavelength. The relationship between the magnitude of the defocus aberration and the physical focal shift, ϵ_z, is given by[103]:

$$\Delta W_{defocus,\mu m} = -\frac{\epsilon_z}{8(f\#)^2} \tag{6.18}$$

ΔW is the defocus calculated from eq. 6.17, reported in physical units. To convert the aberration magnitude from radians to physical units, the magnitude in radians is divided by 2π and multiplied by the wavelength.

$$\Delta W = \frac{\Phi_2(\lambda)\theta_{marg}^2\lambda}{2\pi} \tag{6.19}$$

ϵ_z is the physical shift along the optical axis from ideal focus, and $f\#$ is the f-number of the system, the ratio of the system focal length to the entrance pupil diameter. Rearranging eq. 6.18, the physical shift associated with the defocus aberrations calculated is given by:

$$\epsilon_z = -8(f\#)^2\Delta W = \frac{-8(f\#)^2\Phi_2\theta_{marg}^2\lambda}{2\pi} \tag{6.20}$$

The units of ϵ_z are units of wavelength. Figures 6.7a-6.7c show the physical shift of focus in microns (in red) and the corresponding defocus magnitude at the exit pupil in milliwaves (in blue) for an f/12.5 reflecting surface which has a marginal ray angle of incidence of 0.02 radians (1.15°) with various coatings.

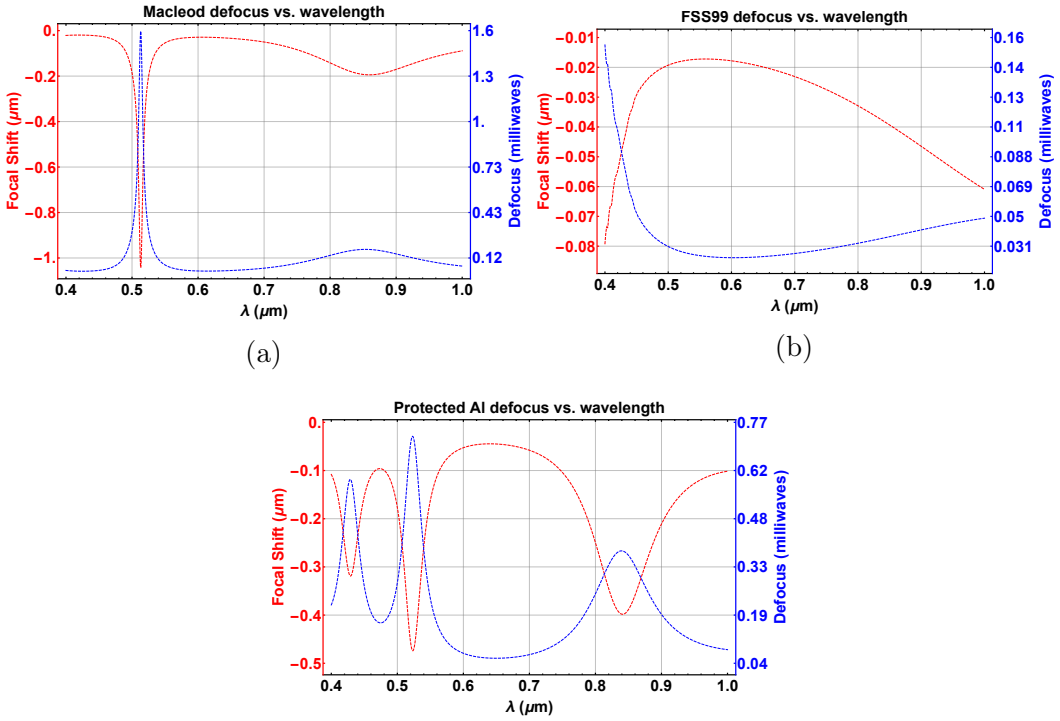


Figure 6.7: Physical focal shift in microns (red curves) and the corresponding amount of defocus in milliwaves (blue curves) as a function of wavelength for (a) Macleod 188, (b) FSS99-600, and (c) protected aluminum.

Inside the highly reflective regions of each coating, the variations in focal point shift across wavelength is less than a tenth of a micron. Outside of the highly reflective regions, the focal shifts vary more rapidly as wavelength changes, with peak variations on the order of about 0.06 microns for FSS99, about 1 micron for Macleod 188, and about 0.4 microns for protected aluminum.

The significance of these focal shifts depends on how tolerant the optical system is to defocus, which is dependent on depth of focus. Depth of focus is proportional to the f-number, so a faster system will have a smaller depth of focus while a slower system will have a larger depth of focus. Thus faster systems are generally less tolerant to defocus while slower systems are more tolerant to defocus. The depth of focus should be calculated and compared to the fo-

cal shifts for a fast system to determine whether the longitudinal chromatic aberration introduced by a surface's coatings are significant.

6.5 Conclusions

It is widely assumed that reflecting surfaces do not exhibit longitudinal chromatic aberration because the paths of light rays follow the wavelength independent law of reflection, whereas lenses do have longitudinal chromatic aberration because the refractive index in Snell's law is wavelength dependent. However, dielectric coated reflecting elements introduce chromatic aberration due to the wavelength dependent nature of the amplitude reflection coefficients. A Taylor series expansion of the Fresnel reflection coefficients showed that bare reflecting surfaces do not have a quadratic component and, therefore, cannot contribute to longitudinal chromatic aberrations.

Using finite differences, quadratic coefficients were calculated for various thin film coatings across wavelength, and the variations in these coefficients were shown to cause different wavelengths of light to have their focal points shifted by different amounts. The magnitude of these focal shifts were dependent on the specific coating, with image shifts on the order of 0.1-0.4 microns for protected aluminum and up to 0.8 microns for Macleod 188. FSS99, the highly reflective coating for spaced based applications, was shown to exhibit the lowest amount of longitudinal chromatic aberration, fluctuating by only 0.06-0.08 microns for most of its spectrum, 400 nm - 1000 nm.

Since defocus is calculated from the quadratic coefficient, which is dependent on the prescription of the dielectric coating, and marginal ray angle of inci-

dence at the mirror, which is dependent on the system's numerical aperture, the longitudinal chromatic aberration induced by a dielectric coated reflecting surface will change with the coating *and* numerical aperture of the system. Additionally, the variations in the quadratic coefficient vary significantly faster outside of the coating's passband, so operating at wavelengths not within the passband will contribute to the chromatic aberration.

CHAPTER 7

Conclusion and Future Work

7.1 Summary

The research objective of this dissertation was to seek answers to these questions: do polarization effects limit a telescope's ability to detect exoplanets and their spectra? If polarization does indeed negatively affect a telescope's ability to do science, is it the limiting factor? What can be done to compensate for or mitigate polarization effects? And are there other polarization effects that have been generally overlooked in optical systems?

In order to answer these questions, the polarization ray tracing program, Polaris-M, was used to analyze the polarization aberrations of two optical systems, HabEx and LUVOIR. Diffraction algorithms in MATLAB were used to determine how these polarization aberrations interacted with one of the proposed coronagraphs for HabEx, a vector vortex coronagraph. A commonly used metric to measure how well a telescope/coronagraph system is suppressing on-axis starlight, the contrast, was calculated without polarization aberrations, with polarization aberrations, and with polarization aberrations and form birefringence associated with coating a large primary mirror. Comparison of the contrast from these three situations revealed that, when no adaptive optics are utilized, adding in polarization aberrations degraded the contrast by several

orders of magnitude. Adding in form birefringence associated with coating a large primary mirror to the isotropic polarization aberrations, the contrast was further degraded by two orders of magnitude.

A technique for reducing/mitigating polarization aberrations was explored. By properly choosing fold mirror orientations, it was shown that the magnitude of the overall diattenuation was reduced by a factor of 1.2 while the overall retardance was reduced by a factor of 2.95. An algorithm to optimize the fold mirror orientations of an optical system was created and demonstrated.

Polarization ray tracing was also used to show that reflecting surfaces contribute longitudinal chromatic aberration to an optical system. A simple parabolic reflecting mirror was analyzed with different dielectric coatings to show how they contributed to the chromatic aberration.

7.2 Future Work

While it has been shown that the polarization aberration due to the thin film coatings and form birefringence in a telescope/coronagraph system will degrade the contrast, this analysis was done without the use of adaptive optics. Any real coronagraph system will undoubtedly utilize some form of adaptive optics. Current adaptive optics algorithms like FALCO[22] or PROPER[104] do not incorporate polarization into their calculations. One of the future goals of this work is to fully model and completely analyze the polarization effects of a coronagraph system with working adaptive optics. But to do so, either the current algorithms will need to have polarization added to them, or new, polarization-specific algorithms will need to be written.

Other future work that is soon to be undertaken at the University of Arizona is analysis of a dichroic for the HabEx optical system. In its current design, the dichroic in the HabEx layout acts as an ideal surface, *i.e.* it perfectly passes or reflects the light incident on each of its surfaces. Of course, this is not how a real dichroic acts. To better model how the dichroic will affect the polarization aberrations in HabEx, an off-the-shelf dichroic will soon be analyzed in a polarimeter at the polarization lab. That data will be incorporated into the polarization ray tracing model of HabEx. Depending on how this dichroic affects polarization aberrations, one of the next steps might include attempting to design or optimize the design of a dichroic for the HabEx system.

7.3 Conclusion

The goal of this dissertation was to determine if polarization aberrations impede a telescope/coronagraph system's ability to detect exoplanets and their spectra. The answer to this question is still open ended, though this dissertation has laid the ground work for answering it. In the absence of adaptive optics and wavefront control, the polarization aberrations degrade the contrast of a vector vortex by several orders of magnitude. However, since any real operating telescope will almost assuredly use some form of adaptive optics, more work must be done to incorporate the effects of polarization into the field of coronagraphy and adaptive optics.

APPENDIX A

Dipole/Double Pole Coordinate Systems

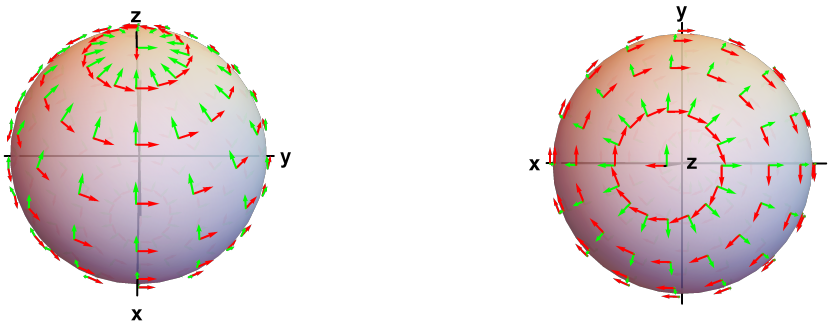
It is often of interest to convert between a three dimensional polarization ray tracing matrix function(defined in global coordinates on a sphere) into a two dimensional Jones pupil (defined in local coordinates) to better visualize the information on a computer screen or printed paper. This appendix details two of the local coordinate systems, dipole coordinates and double pole coordinates, that are commonly used to convert three dimensional polarization PRT matrices into two dimensional Jones matrices.

A.0.0.1 Dipole Coordinate System

The first coordinate system discussed here is the dipole coordinate system, like Earth's latitude and longitude system. This coordinate system is defined relative to a polar axis, $\hat{\mathbf{a}}_{loc}$, which defines the location of two singularities (poles). For an arbitrary propagation vector, $\hat{\mathbf{k}}$, the local x-coordinate and y-coordinate are defined as:

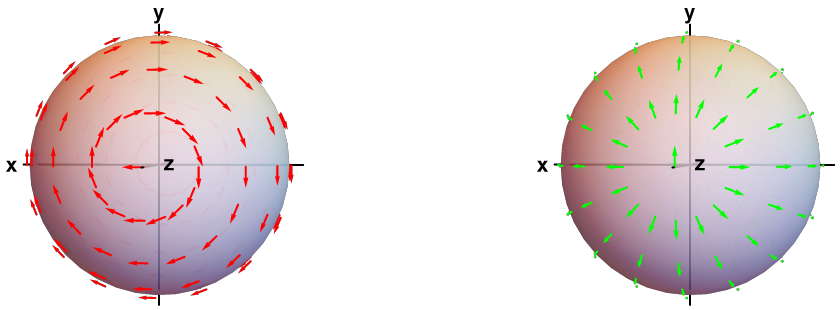
$$\hat{\mathbf{x}}_{loc} = \frac{\hat{\mathbf{a}}_{loc} \times \hat{\mathbf{k}}}{|\hat{\mathbf{a}}_{loc} \times \hat{\mathbf{k}}|}$$
$$\hat{\mathbf{y}}_{loc} = \hat{\mathbf{k}} \times \hat{\mathbf{x}}_{loc} \tag{A.1}$$

For example, if the dipole axis is chosen to be along the z-axis, $\hat{\mathbf{a}}_{loc} = (0, 0, 1)$, the two poles of this dipole coordinate system will be at (0,0,1) and (0,0,-1). Figure A.1 shows the local coordinates of a dipole system with such a dipole axis. Highlighted in figure A.2, the local coordinates vary rapidly when the propagation vector approaches one of the poles of the dipole coordinate system.



(a) (b)

Figure A.1: Dipole coordinate system with the dipole axis oriented along the z-axis shown from two different view angles. The red arrows show the local x-basis while the green arrows show the local y-basis.



(a) (b)

Figure A.2: (a) Dipole local x-coordinates and (b) dipole local y-coordinates. As the propagation vector approaches the pole, the local x and y-coordinates change rapidly.

When the propagation vector lies in the same direction as the dipole axis ($\hat{\mathbf{k}} = \pm\hat{\mathbf{a}}_{loc}$), equation A.1 becomes singular. To avoid problems using the singularity when using the dipole basis to convert between PRTs and Jones matrices, any rays with propagation vectors in the small area around the dipoles are treated as special cases and their local basis vectors are set to constants. The dipole coordinate system best depicts the polarizations of spherical wavefronts emerging from linear polarizers[105].

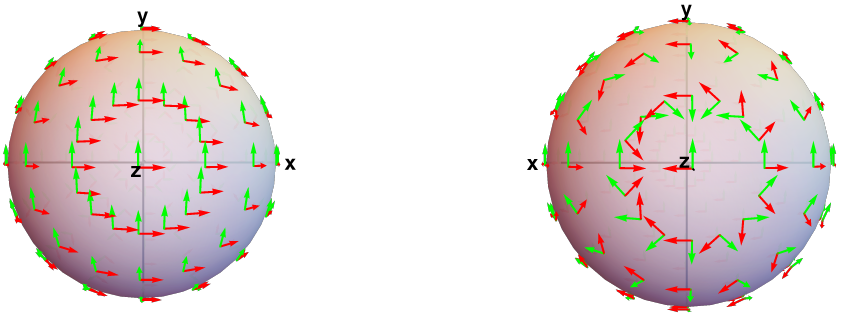
A.0.0.2 Double Pole Coordinate System

The second coordinate system is the double pole coordinate system. According to the Hairy Ball Theorem[106], any continuous tangent vector field on a sphere must disappear at at least two points, *i.e.* no set of local coordinates exist such that there is no singularity across a sphere. In the case of the dipole coordinate system, two singularities exist, one at each pole. The double pole coordinate

system, on the other hand, is constructed such that these two singularities coincide, making a double pole at one point. The double pole coordinate system is defined by the axis of the antipole, $\hat{\mathbf{a}}_{loc}$ (the direction opposite of the double pole), and the local x-coordinate at the antipole, $\hat{\mathbf{x}}_0$. With those two chosen, the double pole coordinates are:

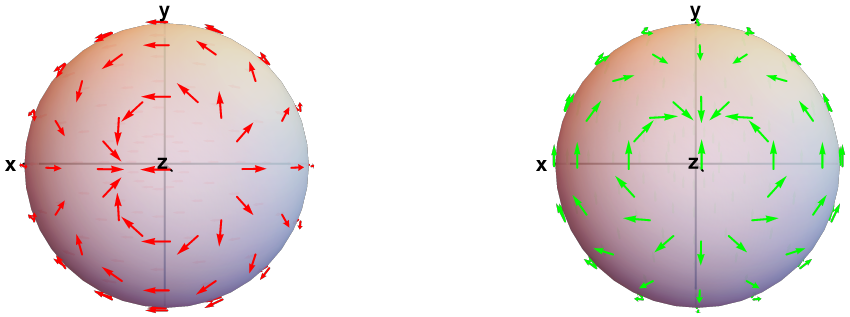
$$\begin{aligned} \hat{\mathbf{x}}_{loc} &= \mathbf{R} \cdot \hat{\mathbf{x}}_0 \\ \hat{\mathbf{y}}_{loc} &= \mathbf{R} \cdot (\hat{\mathbf{a}}_{loc} \times \hat{\mathbf{x}}_0) \end{aligned} \tag{A.2}$$

where $\mathbf{R}$ is a rotation about axis $\hat{\mathbf{r}}$ by angle θ, $\hat{\mathbf{r}} = \hat{\mathbf{k}} \times \hat{\mathbf{a}}_{loc}$, and $\theta = -cos^{-1}(\hat{\mathbf{k}} \cdot \hat{\mathbf{a}}_{loc})$. Figure A.3 shows the double pole with $\hat{\mathbf{a}}_{loc} = (0, 0, 1)$ and $\hat{\mathbf{x}}_0 = (1, 0, 0)$. This puts the double pole at (0,0,-1). The left side of the figure shows the +z hemisphere, the furthest propagation vectors from the double pole. The local coordinates are smoothly and slowly varying for these propagation vectors. The right side of the figure shows the -z hemisphere, which contains the double pole. As the propagation vectors approach the double pole, the local coordinates vary rapidly, going through 4π rotations as the propagation vector goes through a 2π rotation about the double pole. Figure A.4 highlights the rapid variation of the local x and local y coordinates around the double pole.



(a) (b)

Figure A.3: Double pole coordinate system with the double pole axis oriented along the z-axis shown from two different view angles. (a) shows the local coordinates near the anti-pole while (b) shows the local coordinates near the double pole.



(a) (b)

Figure A.4: (a) Double pole local x-coordinates and (b) double pole local y-coordinates. As the propagation vector approaches the pole, the local x and y-coordinates change rapidly. A full circle (2π rotation) around the double pole results in a 4π rotation of the local coordinate's orientation.

When the propagation vector lies in the same direction as the double pole ($\hat{\mathbf{k}} = -\hat{\mathbf{a}}_{loc}$), equation A.2 becomes singular. The double pole coordinate system best depicts the polarization of light exiting an ideal lens.

APPENDIX B

NMinimize and Minimizing Ξ in Mathematica

Mathematica's NMinimize function takes in a function and set of constraints then numerically steps through iterations of that function to locate a local or global minimum. This appendix provides the functions that were used to calculate Ξ and the Mathematica and Polaris-M code necessary to do the fold mirror rotation and optimization.

Rotate surface function

Function

rotateSurface[centerRay_,surfaceToRotate_,θ_,systemToRotate_:newmasktoend]

This function takes in ray data, surface data, and a rotation angle and returns the optical system where all succeeding elements are rotated about the mirror's normal.

Inputs

centerRay: ray traced data for a single ray through the center of the system

surfaceToRotate: which fold mirror surface that is going to be rotated

θ: the angle that the surface will be rotated (in radians)

systemToRotate: parameters of the optical system that will be rotated. Defaults to the original system with no surfaces rotated.

Outputs

rotatedHabex: the new system layout with rotated fold mirrors

Mathematica code:

```
rotateSurface[centerRay_, surfaceToRotate_, θ_, systemToRotate_: originalSys] :=
  Module[{surfBeforeray, surfray, rotaxis, rotpoint, vertexrot,
    normrot, rotsurfNorm, rotsurfvert, rotcenterVertex, rotatedHabex},

   surfBeforeray = Select[centerRay, #[[ray`surfaceID]] == surfaceToRotate - 1 &];

   surfray = Select[centerRay, #[[ray`surfaceID]] == surfaceToRotate &];

   rotaxis = surfBeforeray[[1, ray`k]];

   rotpoint = surfray[[1, ray`r]];

   vertexrot = RotationTransform[θ, rotaxis, rotpoint];
   normrot = RotationTransform[θ, rotaxis];

rotsurfNorm = Table[normrot[surfNorm[[ii]]], {ii, Length@surfNorm}];
   rotsurfvert = Table[vertexrot[surfvert[[ii]]], {ii, Length@surfNorm}];
   rotcenterVertex = Table[vertexrot[centerVertex[[ii]]], {ii, Length@surfNorm}];
rotatedHabex = systemToRotate;
   Table[rotatedHabex[[ii, sys`a]] = normrot[systemToRotate[[ii, sys`a]]];
    rotatedHabex[[ii, sys`v]] = vertexrot[systemToRotate[[ii, sys`v]]];
    rotatedHabex[[ii, sys`aperture]] = (True &);
    , {ii, surfaceToRotate, 23}];

   Return[rotatedHabex]];
```

The rotateSurface function does the following:

-selects the ray incident on the surface to be rotated

  -extracts propagation direction incident on the surface

-selects the ray exiting the surface to be rotated

  -extracts vertex of surface interaction with ray

-calculates two rotation transforms, one for rotating surface normals and one for rotating surface vertices

-applies rotation transforms (both surface normal and vertex) to the mirror being rotated and every subsequent mirror in the optical system

-updates the optical system with the newly calculated optical surfaces

-sets the apertures to be open

Pauli representation of Jones matrix function

Function

PRTtoPauli[PRT_,q_,Oin_]

This function takes in ray data and calculates the Jones matrix of that ray's interaction with the optical system. Using the weak polarization approximation, it decomposes that Jones matrix into a non-polarizing coefficient and 3 coefficients that relate to the linear horizontal, linear 45 °, and circular diattenuation and retardance.

Inputs

PRT: Polarization ray tracing matrix from a polarization ray trace

q: Parallel transport matrix from a polarization ray trace

Oin: Unitary matrix that transforms between the local coordinates and global coordinates for the ray entering the system

Output

c0: non-polarizing term of the Jones matrix

f1: linear horizontal diattenuation and retardance term

f2: linear 45° diattenuation and retardance term

f3: Circular diattenuation and retardance term

Mathematica code:

```
PRTtoPauli[PRT_, q_, Oin_] := Module[{Oout, j, c0, c1, c2, c3, f1, f2, f3},
  Oout = q.Oin;
  j = (Inverse[Oout].PRT.Oin)[[1 ;; 2, 1 ;; 2]];
  c0 = (j[[1, 1]] + j[[2, 2]])/2;
  c1 = (j[[1, 1]] - j[[2, 2]])/2;
  c2 = (j[[1, 2]] + j[[2, 1]])/2;
  c3 = (i (j[[1, 2]] - j[[2, 1]]))/2;

  f1 = c1/c0;
  f2 = c2/c0;
  f3 = c3/c0;

  Return[{c0, f1, f2, f3}]];
```

Function

f[θ4_,θ5_,θ10_,θ11_,θ12_]

This function takes in the five rotation angles of the fold mirrors and returns the value of the merit function Ξ. The ?NumericQ that appears in the actual code is there to tell Mathematica that the inputs to the function are numeric quantities and not symbolic. This makes it easier for the optimizer to process the function.

Inputs
$\theta 4$: Rotation angle of the first fold mirror (the fourth surface in HabEx)
$\theta 5$: Rotation angle of the second fold mirror (the fifth surface in HabEx)
$\theta 10$: Rotation angle of the third fold mirror (the tenth surface in HabEx)
$\theta 11$: Rotation angle of the fourth fold mirror (the eleventh surface in HabEx)
$\theta 12$: Rotation angle of the fifth fold mirror (the twelth surface in HabEx)

Output
Ξ: merit function for RMS diattenuation and retardance magnitude

Mathematica code:

Merit function

```
f[θ4_?NumericQ, θ5_?NumericQ, θ10_?NumericQ, θ11_?NumericQ, θ12_?NumericQ] :=
  Module[{rot4, rot5, rot10, rot11, rot12, surface4out, surface5out, surface10out,
    surface11out, numrays, xsize, ysize, raysin, raysout, lastrays, PRT, Q, pauli, Ξ},
   rot4 = rotateSurface[singleout, 4, θ4 °, originalSys];
   surface4out = TraceRays[singlein, rot4];
   rot5 = rotateSurface[surface4out, 5, θ5 °, rot4];
   surface5out = TraceRays[singlein, rot5];
   rot10 = rotateSurface[surface5out, 10, θ10 °, rot5];
   surface10out = TraceRays[singlein, rot10];
   rot11 = rotateSurface[surface10out, 11, θ11 °, rot10];
   surface11out = TraceRays[singlein, rot11];
   rot12 = rotateSurface[surface11out, 12, θ12 °, rot11];
   config`rayID = 1;
   numrays = 11;
   xsize = 2 * size[[1]];
   ysize = xsize;
   raysin = CreateCollimatedRectRayGrid[xsize, ysize, numrays, numrays,
     ray`λ → .5, ray`k → rayk, ray`r → rayr, ray`surfaceOrder →
      {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 20, 22, 23}];
   raysout = TraceRays[raysin, rot12];

   lastrays = Select[raysout, #[[ray`surfaceID]] == 23 &];
   PRT = lastrays[[All, ray`PRTCumulative]];
   Q = lastrays[[All, ray`QCumulative]];
   pauli = Table[PRTtoPauli[PRT[[ii]], Q[[ii]], sin, pin, Oin], {ii, Length@lastrays}];

   Ξ = Sqrt[1/Length@lastrays Total[(Norm /@ pauli[[All, 2 ;; 4]])^2]];

   Return[Ξ]];
```

Execute optimization

Using all of the previously defined functions (some of which depend on Polaris-M functionality), the following code is used to call Mathematica's optimization routines for the five fold mirror angles that minimize the merit function, Ξ. The absolute timing function used here helps to keep track of how long the optimization process takes. The NMinimize function is constrained so that the allowable angles for surface rotations are confined to the unit circle, i.e. every angle can only go from 0 to 360°.

```
AbsoluteTiming[
 NMinimize[{f[θ1, θ2, θ3, θ4, θ5], 0 ≤ θ1 ≤ 360 && 0 ≤ θ2 ≤ 360 && 0 ≤ θ3 ≤ 360 &&
    0 ≤ θ4 ≤ 360 && 0 ≤ θ5 ≤ 360 }, {θ1, θ2, θ3, θ4, θ5}]]
```

Additional options can be used with the NMinimize:
The EvaluationMonitor option can keep count of the number of evaluations

The StepMonitor option can be set to print out the current values of the five angles being optimized as

well as the value of the merit function at each step in the process.

These two options were not used in the code because the minimization process took approximately 18 hours, so the list printed out by EvaluationMonitor and StepMonitor would be extremely long.

APPENDIX C

Chromatic Aberration

This appendix details the prescription and calculations used to create figures 6.1 and 6.2, the exact thin film coatings for FSS99-600, Macleod 188, and protected aluminum, and shows the astigmatism that appears when there is a quadratic difference between the phase of s- and p-polarized light reflecting from a surface. Table C.1 lists the surface parameters of the example reflecting surface that was geometrically ray traced in Zemax to create OPD plots and polarization ray traced in Polaris-M to create reflected phase plots showing the coating induced phase contribution. Table C.2 lists the coating parameters for FSS99-600, Macleod 188, and protected aluminum.

Radius of curvature	25 mm
Conic	$\kappa = -1$
Aperture Size (diameter)	1 mm
Focal length	12.5 mm
Coating	Various (detailed below)
Marginal Ray AOI	0.02 radians

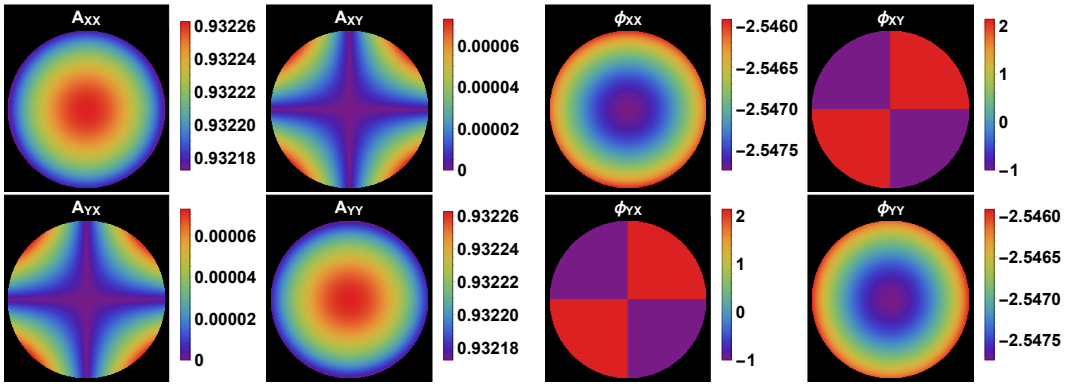
Table C.1: Parameters for the example parabolic reflecting surface

Coating	FSS99-600	Macleod 188	Protected Al
Layer 1	65 nm HfO_2	50 nm, n = 2.3	80 nm HfO_2
Layer 2	60 nm SiO_2	83.3 nm, n = 1.38	107.5 nm SiO_2
Layer 3	15 nm HfO_2	50 nm, n = 2.3	80 nm HfO_2
Layer 4	0.101 nm Cr	83.3 nm, n = 1.38	107.5 nm SiO_2
Layer 5	-	50 nm, n = 2.3	80 nm HfO_2
Layer 6	-	-	107.5 nm SiO_2
Substrate	Ag	Al	Al

Table C.2: Coating specifications for FSS99-600, Macleod 188, and protected aluminum

The first layer in table C.2 is the top layer of the coating, furthest from the substrate. The increasing layers progress towards the substrate.

Figure C.1 shows the Jones pupil (eq.6.2) for the example parabolic reflecting surface with a protected aluminum coating created by polarization ray tracing an on-axis grid of 40401 rays (201x201 square grid) with a wavelength of 500 nm in Polaris-M. The reference sphere for this pupil is a sphere of radius 12.5 mm, centered on the optical axis 12.5 mm away from the focal point of the mirror. The left side of the figure shows the amplitude of the Jones pupil and the right side of the figure shows the phase of the Jones pupil in radians. The Jones pupil is relatively close to an identity matrix, indicating relatively low amounts of polarization aberration. The sharp edges in the off-diagonal phases are π phase jumps due to the zero crossings in the off-diagonal amplitudes. The phase of the xx component and the yy component are nearly identical on this colorbar. This is because the s- and p- reflected phases for protected aluminum are very similar to one another (see figure 6.4c at max angle of incidence equal to the marginal ray's angle of incidence of 0.02 radians).

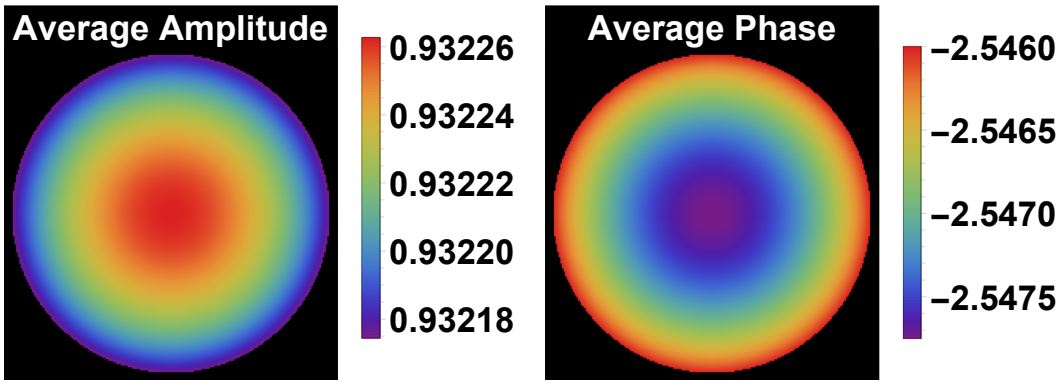


(a) (b)

Figure C.1: Jones pupil for the protected aluminum parabolic reflecting surface, detailed in table C.1, at 500 nm. (a) is the amplitude of the Jones pupil and (b) is the phase of the Jones pupil in radians.

The average amplitude reflected from this parabolic mirror surface, shown in figure C.2a, is the average of the on-axis amplitude components, A_{xx} and A_{yy}. This corresponds to the amplitude transmission term, $A(x, y, \lambda)$, in eq. 6.3. The average reflected wavefront, shown in figure C.2b, is the average of the on-axis phase components, ϕ_{xx} and ϕ_{yy}. This corresponds to the wavefront aberration term, $\frac{2\pi W(x,y,\lambda)}{\lambda}$, in eq. 6.3.

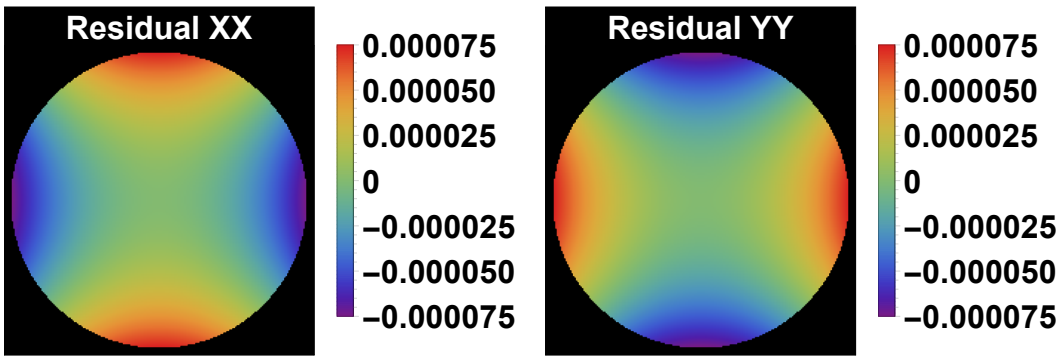
A horizontal slice through the average reflected wavefront was used to create the 500 nm curve in figure 6.2, with an additional step of subtracting off the piston term such that the phase at the center of the pupil was zero. The 400 nm and 600 nm curves were created by polarization ray tracing a grid of rays at the 400 nm and 600 nm, calculating Jones pupils for those wavelengths, extracting the average reflected wavefront, and taking a horizontal slice through that wavefront.



(a) (b)

Figure C.2: Average amplitude (a) and average wavefront in radians (b) of the protected aluminum parabolic reflecting surface at 500 nm.

Figure C.3 shows the phase of the on-diagonal elements of the Jones pupil after the average phase is factored out. These correspond to the phase of the on-diagonal terms of the Jones matrix term, $\mathbf{J}(x, y, \lambda)$ in eq. 6.3. This astigmatism results from the quadratic variation of retardance for the coating on the parabolic surface, which arises from the difference in the s- and p-reflected quadratic phase as seen in figure 6.4c.



(a) (b)

Figure C.3: Astigmatism in the on-diagonal phase terms after the average phase is subtracted out. (a) shows the astigmatism from the XX element, (b) shows the astigmatism from the YY element. The legend for these figures is in radians.

APPENDIX D

Polaris-M Code for Ray Tracing HabEx

Ray Trace Setup and trace

System Parameters

```
surfacedata =
 Import["D:\\Public\\Documents\\Updated optical system HabEx\\surface parameters.dat",
  "Data", CharacterEncoding → "Unicode", HeaderLines → 20];
(*import surface data from a file created from Zemax*)

surfaces = {"surface number", "surface name", "surface physical radius"};
(*contains data regarding which surfaces in the Zemax file are optical surfaces
 rather than dummy surfaces. Also contains data regarding physical size of
 optical surfaces. Actual data can't be shown due to ITAR restrictions.*)

radii = surfacedata[[surfaces[[All, 1]], 3]];

κ = surfacedata[[surfaces[[All, 1]], -2]];

size = surfaces[[All, 3]];
```

Surf Vertex

```
surfvertandnorm =
  Import["D:\\Public\\Documents\\Updated optical system HabEx\\global surface vertices
     and normals.dat", "Data", CharacterEncoding → "Unicode", HeaderLines → 23];
surfvert = Table[surfvertandnorm[[ii ;; ii + 2, 4]], {ii, 1, 67, 3}];
(*surface vertices from Zemax*)
```

In[15]:=

Surf Norm

```
surfvertangle = Table[surfvertandnorm[[ii ;; ii + 2, 5]], {ii, 1, 67, 3}];
tilt[α_,β_,γ_]:=RotationMatrix[α,{1,0,0}].
   RotationMatrix[β,{0,1,0}].RotationMatrix[γ,{0,0,1}].{0,0,1};
surfNorm = Table[tilt[surfvertangle[[ii,1]]°,surfvertangle[[ii,2]]°,surfvertangle[[ii,3]]°],
   {ii,Length@surfvertangle}];

pos = (Position[radii, n_ /; n < 0])[[All, 1]];
Table[surfNorm[[ii]] = -surfNorm[[ii]]; radii[[ii]] = -radii[[ii]], {ii, pos}];

(*surface normals from Zemax. Zemax gives the surface normals as rotations angles,
so this code applies those rotation angles to the {0,0,1} vector to
 calculate the surface normals. The table at the end changes the sign on
 the radius of curvature so that the surfaces curve the correct way.*)
```

Center Vertex

```
vertexdata =
  Import["D:\\Public\\Documents\\Updated optical system HabEx\\center ray trace.dat",
   "Data", CharacterEncoding → "Unicode", HeaderLines → 22];
centerVertex = Table[vertexdata[[surf, 2 ;; 4]], {surf, surfaces[[All, 1]]}];

(*Since many of the surfaces HabEx are OAPs,
the center of the surface may not align with the surface vertex. centerVertex is
 the position data for a ray that was traced through the center of the system,
giving locations of the center of the optical surfaces*)
```

Mirror coatings MgF_2

```
coatindexMgF2 = {"USERISOTROPIC_MGF2jeff", "USERISOTROPIC_ALjeff"};
coatthicknessMgF2 = {25 * 10^-3, .1};

(*prescription for magnesium fluoride on aluminum. 25 nm of mgf2 on 100 nm of aluminum*)
```

Mirror coatings FSS99

```
coatthicknessAg = 10^-3 {65, 60, 15, .101, 150, 100 * 10^3};(*thickness of metal coating"*)
coatindexAg = {"USERISOTROPIC_HFO2", "USERISOTROPIC_SIO2", "USERISOTROPIC_HFO2",
   "USERISOTROPIC_CR", "USERISOTROPIC_AG", "USERISOTROPIC_SIO2"};

(*prescription for FSS99-600
  65 nm hfo2, 60 nm sio2, 15 nm hfo2, .101 nm cr,150 nm silver, 100 micron sio2*)
```

System setup

```
habexToMask = Table[
   CreateSurface[
    sys`surfID → ii,
    sys`surfaceLabel → surfaces[[ii, 2]],
    sys`shape`type → If[κ[[ii]] == 0, "Plane", "AnamorphicAsphere"],
    sys`a → surfNorm[[ii]],
    sys`v → surfvert[[ii]],
    sys`material1 → If[ii == 7, "UserIsotropic_FusedSilicaJeff", "Air"],
    sys`material2 → If[ii == 6 ∨ ii == 14, "UserIsotropic_FusedSilicaJeff", "Air"],
    sys`mode → If[ii == 14, {"Absorb"}, If[ii == 6 ∨ ii == 7, {"Refract"}, {"Reflect"}]],
    sys`coating`type → If[ii == 6 ∨ ii == 7, "Perfect", "Isotropic"],
    sys`coating`thickness →
     If[ii < 3, coatthicknessMgF2, If[ii == 6 ∨ ii == 7 ∨ ii == 14, {}, coatthicknessAg]],
    sys`coating`indices → If[ii < 3, coatindexMgF2,
      If[ii == 6 ∨ ii == 7 ∨ ii == 14, {}, coatindexAg]],
    sys`aperture → (Evaluate[Norm[{#1 - centerVertex[[ii, 1]], #2 - centerVertex[[ii, 2]],
          #3 - centerVertex[[ii, 3]]}] ≤ Norm[size[[ii]]] ] &),
    sys`shape`curvX → If[κ[[ii]] == 0, 0, 1/radii[[ii]]],
    sys`shape`curvY → If[κ[[ii]] == 0, 0, 1/radii[[ii]]],
    sys`shape`conicKX → If[κ[[ii]] == 0, 0, κ[[ii]]],
    sys`shape`conicKY → If[κ[[ii]] == 0, 0, κ[[ii]]]
   ], {ii, 14}];
(*habexToMask creates the first half of the habex system,
going from primary mirror to coronagraph mask plane. The table
 above sets up the optical system by assigning the proper shape, vertex,
surface normal, material, coating, and aperture to each optical surface.*)
```

In[35]:=
```
habexMaskToEnd = Table[
    CreateSurface[
     sys`surfID → ii - 14,
     sys`surfaceLabel → surfaces[[ii, 2]],
     sys`shape`type → If[κ[[ii]] == 0, "Plane", "AnamorphicAsphere"],
     sys`a → surfNorm[[ii]],
     sys`v → surfvert[[ii]],
     sys`material1 → If[ii == 20, "UserIsotropic_FusedSilicaJeff", "Air"],
     sys`material2 → If[ii == 21 , "UserIsotropic_FusedSilicaJeff", "Air"],
     sys`mode → If[ii == 23, {"Absorb"},
       If[ii == 16 ∨ ii == 18 ∨ ii == 20 ∨ ii == 21 ∨ ii == 23, {"Refract"}, {"Reflect"}]],
     sys`coating`type → If[ii == 16 ∨ ii == 18 ∨ ii == 20 ∨ ii == 21 ∨ ii == 23,
       "Perfect", "Isotropic"],
     sys`coating`thickness → If[ii == 16 ∨ ii == 18 ∨ ii == 20 ∨ ii == 21 ∨ ii == 23,
       {}, coatthicknessAg],
     sys`coating`indices →  If[ii == 16 ∨ ii == 18 ∨ ii == 20 ∨ ii == 21 ∨ ii == 23,
       {}, coatindexAg],
     sys`aperture → (Evaluate[Norm[{#1 - centerVertex[[ii, 1]], #2 - centerVertex[[ii, 2]],
           #3 - centerVertex[[ii, 3]]}] ≤ Norm[size[[ii]]] ] &),
     sys`shape`curvX → If[κ[[ii]] == 0, 0, 1/radii[[ii]]],
     sys`shape`curvY → If[κ[[ii]] == 0, 0, 1/radii[[ii]]],
     sys`shape`conicKX → If[κ[[ii]] == 0, 0, κ[[ii]]],
     sys`shape`conicKY → If[κ[[ii]] == 0, 0, κ[[ii]]]
    ], {ii, 15, 23}];

(*habexMaskToEnd sets up the second half of the system*)
```

In[36]:=
```
newmasktoend = Prepend[habexMaskToEnd, habexToMask[[-1]]];
newmasktoend[[1, sys`material2]] = "Air";
newmasktoend[[1, sys`mode]] = "Refract";
newmasktoend[[1, sys`coating]] = {"Perfect", ""};

(*newmasktoend puts the two parts of the system together*)
```

□ Starting position and k-vector

```
rayk = vertexdata[[1, 5 ;; 7]];
rayr = vertexdata[[1, 2 ;; 4]];
```

Ray Trace

```
wavelengths = 1/1000 {450, 475, 500, 525, 550};
```

```
(*trace wavelengths .450-.550 μm*)
Do[
 λ = wavelengths[[ii]];

 dir = CreateDirectory[
   "D:\\Public\\Documents\\Updated optical system HabEx\\" <> ToString[λ * 10^3]];

 config`rayID = 1;
 numrays = 101;
 xsize = 2 size[[1]];
 ysize = xsize;
 rays = CreateCollimatedRectRayGrid[xsize, ysize, numrays, numrays,
   ray`r → rayr, ray`λ → λ, ray`k → rayk, ray`surfaceOrder → Range[14]];
 (*This creates an input grid of rays that will be traced through the system*)

 raysout = AbsoluteTiming[TraceRays[rays, habexToMask]];

 (*traces the grid of rays through the first part of the system*)

sysAXP = -15864.33;
 (*distance from coronagraph mask plane to exit pupil of system A*)
 sysBXP = -6654.086;
 (*distance from final image plane to exit pupil of system B*)
{sysAXPrays, jp} = myJonesPupil[habexToMask, raysout[[2]], sysAXP, 14, numrays];
 (*myJonesPupil is functionally the same as the Polaris-
  M function JonesPupil. The only difference is JonesPupil returns
   the Jones pupil of the system while myJonesPupil returns the
   Jones pupil and the rays at the exit pupil of the system.*)

 Export[dir <> "\\jpA " <> ToString[λ * 10^3] <> " nm.h5",
  {Re@jp, Im@jp}, {"Datasets", {"real", "imaginary"}}];
 (*export the Jones pupil data to a file*)

gridA = myRebuildGrid[sysAXPrays, numrays];
 (*myRebuildGrid rearranges the flattened list of rays back into a grid of rays*)
 config`rayID = 1;
 sysBrays = CreateCollimatedRectRayGrid[1, 1, numrays, numrays, ray`r → {0, 0, 0},
   ray`λ → λ, ray`k → {0, 0, 1}, ray`surfaceOrder → {14, 1, 2, 3, 4, 5, 7, 6, 8, 9}];
 sysBinput = Table[ReplacePart[sysBrays[[ii, jj]],
    {ray`r → gridA[[ii, jj, ray`r]],
     ray`k → gridA[[ii, jj, ray`k]],
     ray`S → gridA[[ii, jj, ray`S]],
     ray`status → If[gridA[[ii, jj, ray`status]] == -8, 1, -1]}],
   {ii, numrays}, {jj, numrays}];
 (*this part essentially resets all the ray data besides the position,
 k-vector, and status so that the rays at the exit pupil of the
  first system can be traced through the second part of the system.*)

 sysBraysout = TraceRays[sysBinput, newmasktoend];
 {sysBXPrays, jpB} = myJonesPupil[habexMaskToEnd, sysBraysout, sysBXP, 9, numrays];
 Export[dir <> "\\jpB " <> ToString[λ * 10^3] <> " nm.h5",
```

```
  {Re@jpB, Im@jpB}, {"Datasets", {"real", "imaginary"}}];
 (*trace rays through the second part of the system,
 get the Jones pupil, export Jones pupil data to file*)

 , {ii, Length@wavelengths}];
```

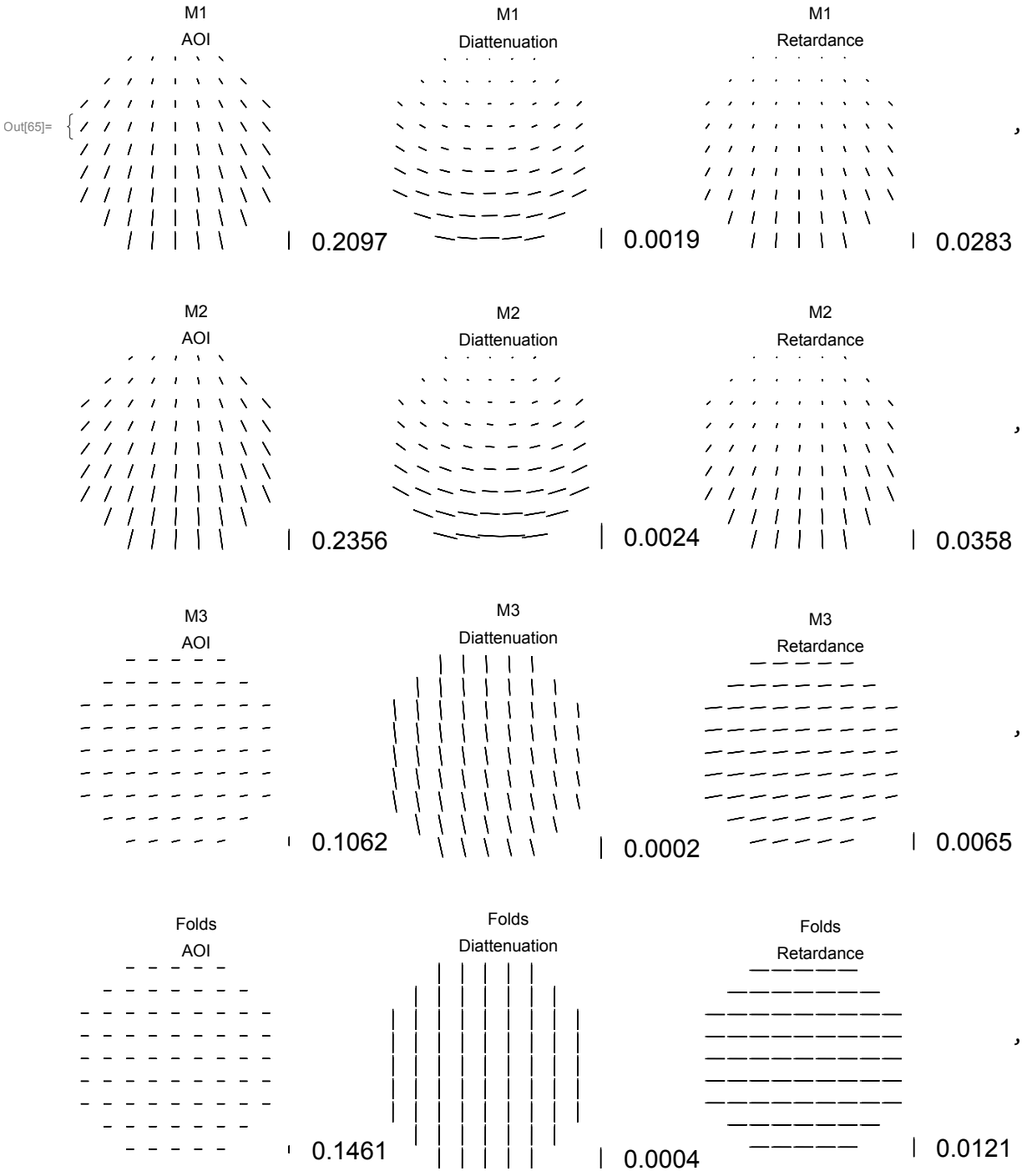
Smaller ray trace for polarization vector maps

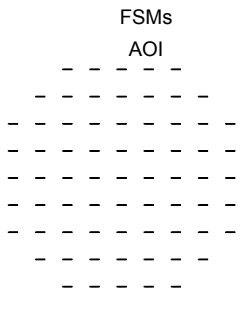
```
(*PVMs need fewer rays, otherwise they would take far too long to calculate and
 plot. This part redoes the full system ray trace with a sparse grid of rays*)
totalSys = Join[habexToMask, habexMaskToEnd];
totalSys[[14, sys`material2]] = "Air";
totalSys[[14, sys`mode]] = {"Refract"};
Table[totalSys[[ii, sys`surfID]] = totalSys[[ii, sys`surfID]] + 14, {ii, Range[15, 23]}];

λ = 500 / 1000;

config`rayID = 1;
numrays = 11;
xsize = 2 size[[1]];
ysize = xsize;
rays = CreateCollimatedRectRayGrid[xsize, ysize, numrays,
    numrays, ray`r → rayr, ray`λ → λ, ray`k → rayk, ray`surfaceOrder →
     {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 20, 22, 23}];
raysout = AbsoluteTiming[TraceRays[rays, totalSys]];
```

☐ surface by surface maps (AOI, dia, ret)

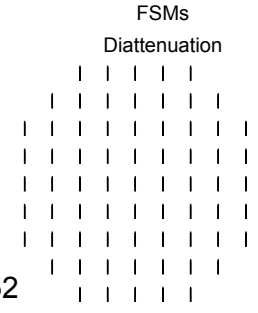
In[62]:=
```
scale = 100 {8, 8, 8, 5, 3, 1, 2, 5, 5, 5, 5, 5, 5, .5, 5, 5, 10, 10, 10, 10, 10, 10};
diascale =
  1000000 {.1, .1, 1, .5, .1, .5, .5, .5, .5, .5, .5, .2, .2, .5, .2, .5, .5, .5, .5, .5, .5, 1};
retscale = 5000 {1, 1, 5, 3, 1, 1, 2, 5, 5, 1, 1, 1, 1, .5, 1, 5, 1, 10, 5, 10, 10, 10};

vectormaps = Table[Row[{
    Graphics[PolarizationVectorMap[raysout[[2]], ii, "AOI", scale[[ii]]],
     PlotLabel → surfaces[[ii, 2]] <> "\nAOI", ImageSize → 200],
    Graphics[PolarizationVectorMap[raysout[[2]], ii, "Diattenuation", diascale[[ii]]],
     PlotLabel → surfaces[[ii, 2]] <> "\nDiattenuation", ImageSize → 200],
    Graphics[PolarizationVectorMap[raysout[[2]], ii, "Retardance", retscale[[ii]]],
     PlotLabel → surfaces[[ii, 2]] <> "\nRetardance", ImageSize → 200]}],
  {ii, {1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 15, 17, 19, 22}}]
```

Out[65]=

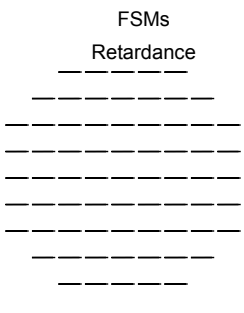

M1
AOI
M1
Diattenuation
M1
Retardance
0.2097
0.0019
0.0283
M2
AOI
M2
Diattenuation
M2
Retardance
0.2356
0.0024
0.0358
M3
AOI
M3
Diattenuation
M3
Retardance
0.1062
0.0002
0.0065
Folds
AOI
Folds
Diattenuation
Folds
Retardance
0.1461
0.0004
0.0121



0.2552

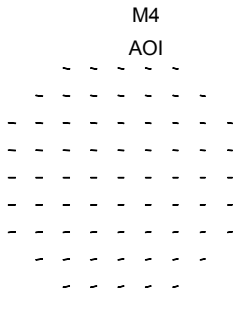


0.0011

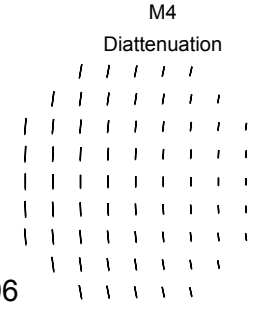


0.0357

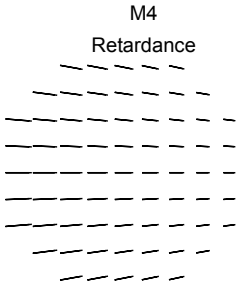
,



0.1196

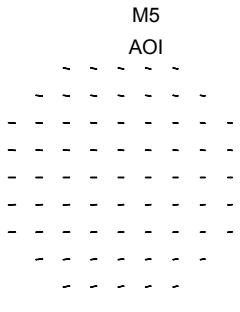


0.0002

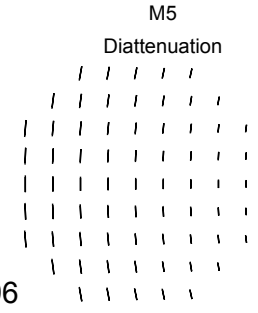


0.0082

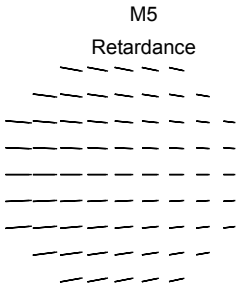
,



0.1196

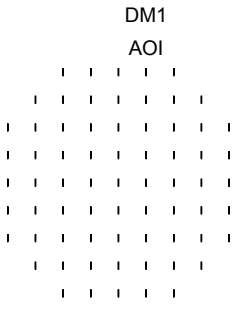


0.0002

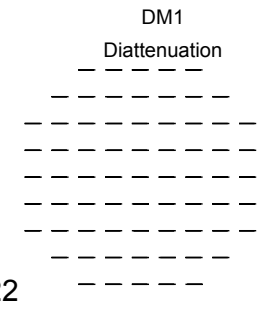


0.0082

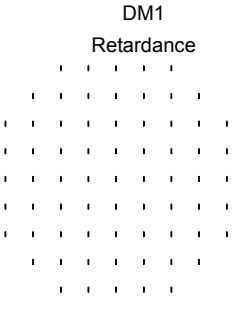
,



0.1222

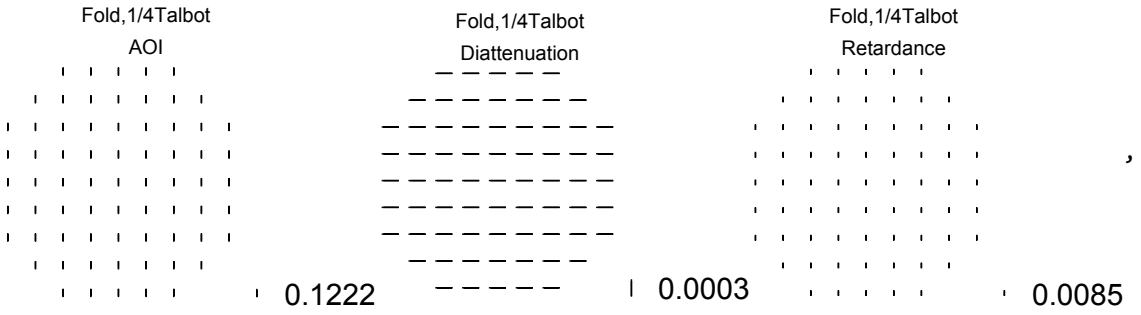


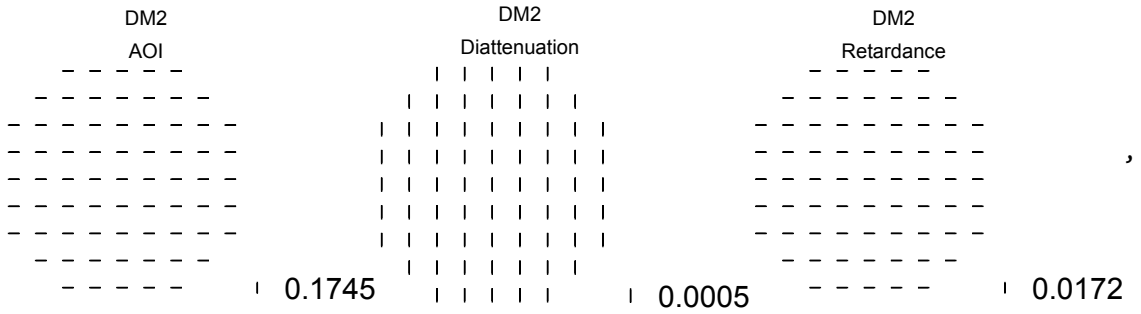
0.0003

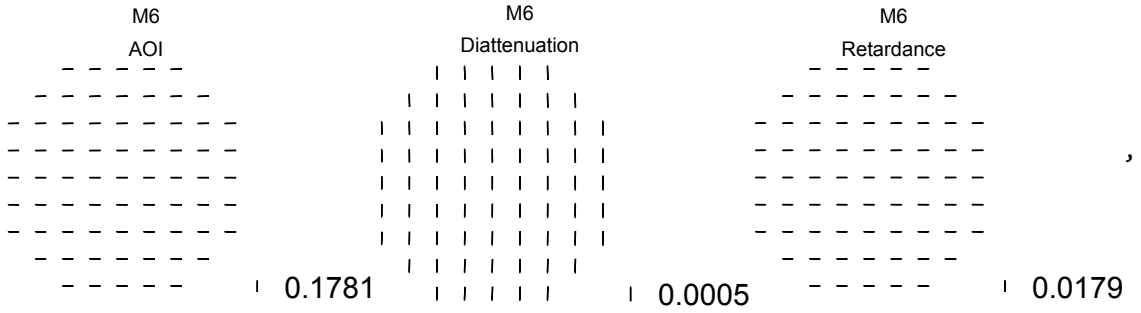


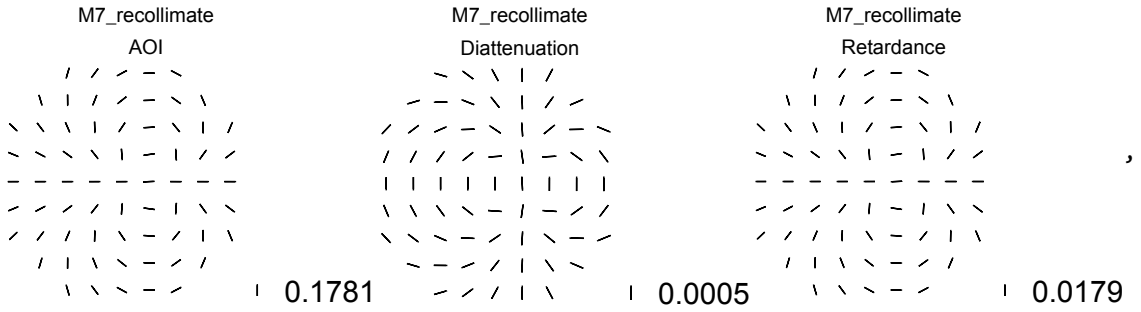
0.0085

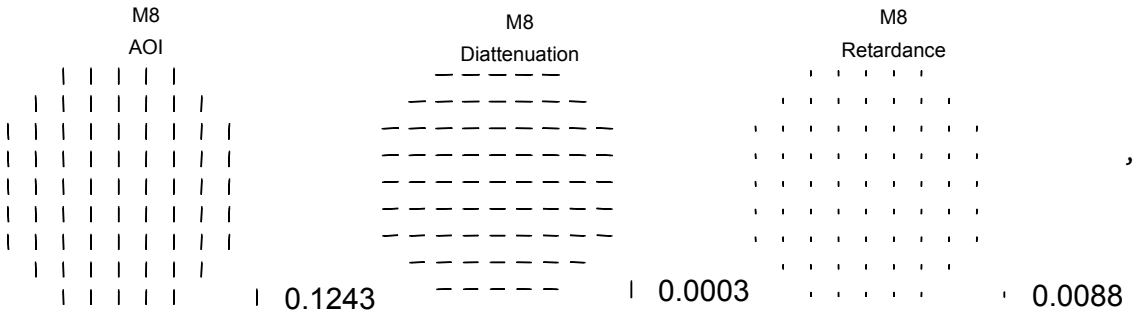
,

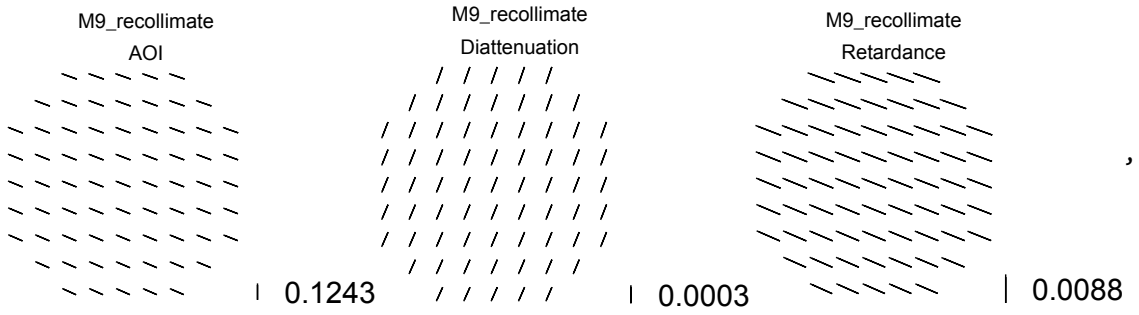

Fold,1/4Talbot
AOI
0.1222
Fold,1/4Talbot
Diattenuation
0.0003
Fold,1/4Talbot
Retardance
0.0085

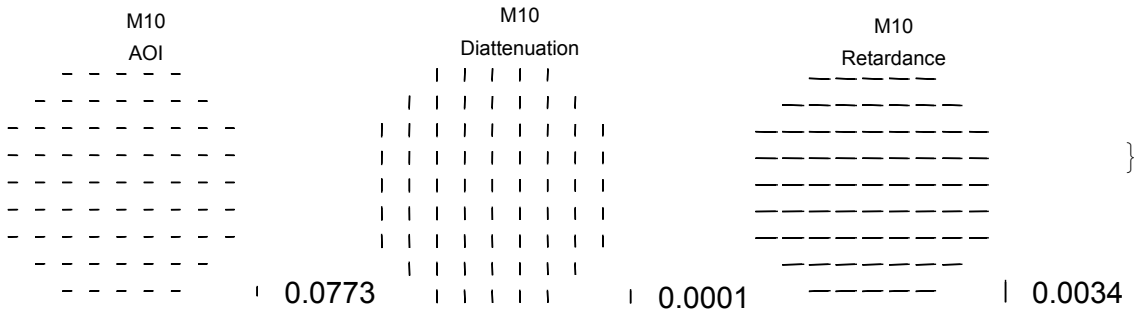

DM2
AOI
0.1745
DM2
Diattenuation
0.0005
DM2
Retardance
0.0172


M6
AOI
0.1781
M6
Diattenuation
0.0005
M6
Retardance
0.0179


M7_recollimate
AOI
0.1781
M7_recollimate
Diattenuation
0.0005
M7_recollimate
Retardance
0.0179



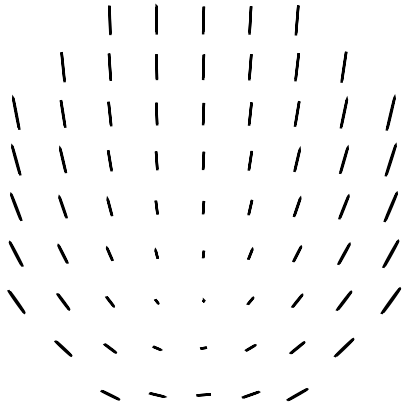




surface by surface cumulative maps

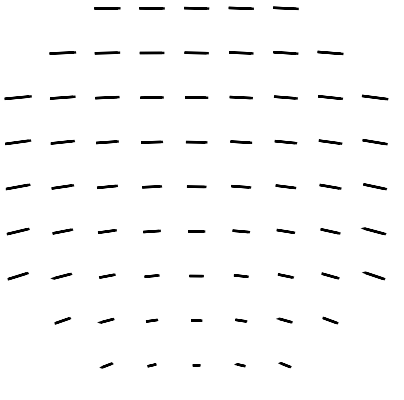
```
cumulativeDia = 100 000 * .5;
cumulativeRet = 5000 * .3;
cumulativemaps = Row[{
   Graphics[PolarizationVectorMap[raysout[[2]],
     22, "Diattenuation", cumulativeDia, Cumulative → True],
    PlotLabel → Style["Cumulative Diattenuation", Bold, 16], ImageSize → 300],
   Graphics[PolarizationVectorMap[raysout[[2]], 22, "Retardance",
     cumulativeRet, Cumulative → True],
    PlotLabel → Style["Cumulative Retardance", Bold, 16], ImageSize → 300]}]
```

Cumulative Diattenuation



0.003

Cumulative Retardance



0.0876

APPENDIX E

Matlab Code for Coronagraph

Table of Contents

Generate Tukey window

```
function w = generateTukeyWindow( Nwindow, RHO, alpha )

Nlut = round(10*Nwindow);
p = linspace(-Nwindow/2,Nwindow/2,Nlut);
lut = tukeywin(Nlut,alpha);

w = interp1(p,lut,RHO,'linear',0);
end
```

Semi-analytic method code

```
function OUT = vortexCoronagraph_Pup2Pup( IN, FPM, apRad, lambdaOverD,
 RHO, N, algo, operation, inVal, outVal, useGPU )
%vortexCoronagraph_Pup2Pup Propagates from the input pupil to output
 pupil
%   Uses either an FFT or local DFT

    showPlots2debug = false;

    if(useGPU)
        IN = gpuArray(IN);
    end
    if( strcmp(algo,'fft') && ~strcmp(operation,'adj') )

        EP = IN;
        OUT = myifft2(myfft2(EP).*FPM);

 elseif( strcmp(algo,'fft') && strcmp(operation,'adj') )

        LP = IN;
        OUT = myifft2(myfft2(LP).*conj(FPM));

    elseif( strcmp(algo,'dft') )

        cut_rad1 = inVal*lambdaOverD;
        cut_rad2 = outVal*lambdaOverD;

        windowKnee = 1-cut_rad1/cut_rad2;
```

```
        D = 2*apRad;

        NA = 2^(nextpow2(D))+2;
        crop = N/2-NA/2+1:N/2+NA/2;


        % DFT vectors
        x = ((0:NA-1)-NA/2)/D;
        u1 = ((0:N-1)-N/2)/lambdaOverD;
        u2 = ((0:N-1)-N/2)*2*cut_rad2/N;

        windowMASK1 = generateTukeyWindow( 2*cut_rad2*lambdaOverD,
 RHO, windowKnee ) ;
        windowMASK2 = generateTukeyWindow( N, RHO, windowKnee ) ;

        if(useGPU)
            x = gpuArray(x);
            u1 = gpuArray(u1);
            u2 = gpuArray(u2);
            windowMASK1 = gpuArray(windowMASK1);
            windowMASK2 = gpuArray(windowMASK2);
        end
        if(~strcmp(operation,'adj'))

            EP = IN;
            EP = EP(crop,crop);
            if showPlots2debug;
 figure;imagesc(abs(EP));axis image;colorbar; title('Cropped
 pupil'); end;

            %%%%%%% Large scale DFT

            FP1 = (N/lambdaOverD)/
(D*N)*exp(-1i*2*pi*u1'*x)*EP*exp(-1i*2*pi*x'*u1);
            if showPlots2debug;
 figure;imagesc(log10(abs(FP1).^2));axis image;colorbar; title('Large
 scale DFT'); end;
            LP1 = (N/lambdaOverD)/
(D*N)*exp(1i*2*pi*x'*u1)*(FP1.*FPM.*(1-
windowMASK1))*exp(1i*2*pi*u1'*x);

            %%%%%%% Fine sampled DFT

            FP2 = 2*cut_rad2/
(D*N)*exp(-1i*2*pi*u2'*x)*EP*exp(-1i*2*pi*x'*u2);
            if showPlots2debug;
 figure;imagesc(log10(abs(FP2).^2));axis image;colorbar; title('Fine
 sampled DFT'); end;
            LP2 = 2*cut_rad2/
(D*N)*exp(1i*2*pi*x'*u2)*(FP2.*FPM.*windowMASK2)*exp(1i*2*pi*u2'*x);

            OUT = padarray_centered(LP1+LP2,NA,NA,N);
            %disp('Propagating through vortex with forward DFT.');
```

```
            if showPlots2debug;
 figure;imagesc(abs(LP1+LP2));axis image;colorbar; title('Lyot
 plane'); end;
            if showPlots2debug; figure;imagesc(abs(LP1+LP2-
EP));axis image;colorbar; title('Lyot plane - Entrance Pupil'); end;
        elseif(strcmp(operation,'adj') )

            LP = IN(crop,crop);

            %%%%%%% Large scale DFT

            FP1 = (N/lambdaOverD)/
(D*N)*exp(-1i*2*pi*u1'*x)*LP*exp(-1i*2*pi*x'*u1);
            EP1 = (N/lambdaOverD)/
(D*N)*exp(1i*2*pi*x'*u1)*(FP1.*conj(FPM).*(1-
windowMASK1))*exp(1i*2*pi*u1'*x);


            %%%%%%% Fine sampled DFT

            FP2 = 2*cut_rad2/
(D*N)*exp(-1i*2*pi*u2'*x)*LP*exp(-1i*2*pi*x'*u2);
            EP2 = 2*cut_rad2/
(D*N)*exp(1i*2*pi*x'*u2)*(FP2.*conj(FPM).*windowMASK2)*exp(1i*2*pi*u2'*x);

            OUT = padarray_centered(EP1+EP2,NA,NA,N);
        end
    else
        error('Error. \nChoose algo = fft or dft. \nChoose forward or
 adj operation.')
    end
    if(useGPU)
        OUT = gather(OUT);
    end
end
```

Vector vortex function

```
function [ FPM,RHO,THETA ] = VVR( N,charge,xvals)

[X,Y] = meshgrid(xvals,xvals); % Grids with Cartesian (x,y)
 coordinates
[THETA,RHO] = cart2pol(X,Y);  % Grids with polar (rho,theta)
 coordinates

FPM = zeros(N,N,2,2);
FPM(:,:,1,1) = cos(charge*THETA);
FPM(:,:,2,2) = -cos(charge*THETA);
FPM(:,:,1,2) = sin(charge*THETA);
FPM(:,:,2,1) = sin(charge*THETA);

FPM(N/2+1,N/2+1,1,1) = 0;
FPM(N/2+1,N/2+1,1,2) = 0;
```

```
FPM(N/2+1,N/2+1,2,1) = 0;
FPM(N/2+1,N/2+1,2,2) = 0;

end
```

Function to read in pupil data and interpolate to scaled pupil size

```
function [ PartA,PartB ] = ReadPupilsInterp( N,Np,lambda)


pupilB = pupilFromh5(['C:\Users\jeffdavis\Documents\HabEx Pupils
 white light\10-30-18 prescription\Correct JP\jpB ' num2str(lambda) '
 nm.h5']);

pupilA = pupilFromh5(['C:\Users\jeffdavis\Documents\HabEx Pupils
 white light\10-30-18 prescription\Correct JP\jpA ' num2str(lambda) '
 nm.h5']);

[Xp,Yp]=meshgrid(linspace(1,101,Np),linspace(1,101,Np));
PartA=zeros(N,N,2,2);
PartB=zeros(N,N,2,2);

for n=1:2
    for m=1:2
temp =
 interp2(squeeze(real(pupilA(:,:,n,m))),Xp,Yp)+1i*interp2(squeeze(imag(pupilA(:,:,
temp = (padarray(temp,[((N-Np)+1)/2 ((N-Np)+1)/2],'pre'));
temp = (padarray(temp,[((N-Np)-1)/2 ((N-Np)-1)/2],'post'));
PartA(:,:,n,m) = temp;
temp =
 interp2(squeeze(real(pupilB(:,:,n,m))),Xp,Yp)+1i*interp2(squeeze(imag(pupilB(:,:,
temp = (padarray(temp,[((N-Np)+1)/2 ((N-Np)+1)/2],'pre'));
temp = (padarray(temp,[((N-Np)-1)/2 ((N-Np)-1)/2],'post'));
PartB(:,:,n,m) = temp;
    end
end


end
```

Apply VVR isotropic

```
function [ JP_A_masked,PartA,PartB,dx,apRad ] =
 ApplyVVRisotropic(lambda,charge)

D=[];%diameter of primary in meters, redacted due to ITAR restrictions


N = 2^12;% Computational grid
Np= 2^10;% Number of points we interpolate to for the pupil diameter
```

```
R = N/Np;%Number of aperature diameters that fit across the
 computational grid

lambdaOverD = (lambda*10^-9)/D;
lambdaOverDmean = (500*10^-9)/D;
xvals=linspace(-R/lambdaOverDmean,R/lambdaOverDmean,N+1);%R*D/lambda
 range
xvals=xvals(1:(end-1));
dx=abs(xvals(1)-xvals(2));%[# of D/lambda per pixel]
apRad = round(1/lambdaOverD*1/dx); % # pixels across aperature radius


[ FPM,~,~ ] = VVR(N,charge,xvals);



[ PartA,PartB ] = ReadPupilsInterp(N,2*apRad+1,lambda);

% % Defines the coordinate systems
[X,Y] = meshgrid(-N/2:N/2-1); % Grids with Cartesian (x,y) coordinates
[~,RHO] = cart2pol(X,Y);  % Grids with polar (rho,theta) coordinates

% Computes the masked PartA
JP_A_masked=zeros(N,N,2,2);

for n=1:2
    for m=1:2
        JP_A_masked(:,:,m,n) =
 vortexCoronagraph_Pup2Pup( squeeze(PartA(:,:,m,n)),
 squeeze(FPM(:,:,m,n)), apRad, R, RHO, N, 'dft', 'forward',0.1/2,
 1.5*4, false );
    end
end

end
```

Main script

reads in Jones pupils, interpolates them to a scaled pupil size, applies the vector vortex mask to the electric field distribution at the coronagraph mask plane and DFTs to get the masked Jones pupil, calculates the end-to-end Jones pupil with and without the vector vortex mask applied, and calculates the ARM^2 for each wavelength with and without the vortex mask. The sum of the 4 components of the ARM^2 yields the PSF.

```
% waveset = [450,475,500,525,550]

% D=4;

% N = 2^12;
% Np = 2^10;
% isotropic_psf_no_mask = zeros(N,N,2,2);
% isotropic_psf_with_mask = zeros(N,N,2,2);
% isotropic_contrast_temp = cell(1,length(waveset));
% charge = 6;
```

```
% for ii = 1:length(waveset)

% lambda = waveset(ii);

% [maskedA,partA,partB,dxtemp,apRad]=
 ApplyVVRisotropic(lambda,charge);

% crop = (N-2*apRad)/2 + 1: (N+2*apRad+2)/2;
% cropMask = maskedA(crop,crop,:,:);
% cropA = partA(crop,crop,:,:);
% cropB = partB(crop,crop,:,:);

% total_no_mask = cell2pupil(pupilmultiply(cropB,cropA));
% total_with_mask = cell2pupil(pupilmultiply(cropB,cropMask));

% padsize = (N-2*apRad)/2;

% total_no_mask = padarray(total_no_mask,[padsize padsize],'pre');
% total_no_mask = padarray(total_no_mask,[padsize-1
 padsize-1],'post');

% total_with_mask = padarray(total_with_mask,[padsize padsize],'pre');
% total_with_mask = padarray(total_with_mask,[padsize-1
 padsize-1],'post');

% for n = 1:2
% for m = 1:2
% isotropic_psf_no_mask(:,:,m,n) =
 abs(fftshift(fft2(fftshift(total_no_mask(:,:,m,n))))).*dxtemp.^2).^2;
% isotropic_psf_with_mask(:,:,m,n) =
 abs(fftshift(fft2(fftshift(total_with_mask(:,:,m,n))))).*dxtemp.^2).^2;
% end
% end

% hdf5write(['C:\Users\jeffdavis\Documents\Dissertation\Figures
\Chapter 4\HabEx psfs\isotropic_psf_no_mask_' num2str(lambda)
 '.h5'],'real',real(isotropic_psf_no_mask((N-Np)/2:(N+Np)/2,(N-Np)/2:
(N+Np)/2,:,:)))
% hdf5write(['C:\Users\jeffdavis\Documents\Dissertation\Figures
\Chapter 4\HabEx psfs\isotropic_psf_with_mask_' num2str(lambda)
 '.h5'],'real',real(isotropic_psf_with_mask((N-Np)/2:(N+Np)/2,(N-
Np)/2:(N+Np)/2,:,:)))
```

Published with MATLAB® R2018a

APPENDIX F

Chapter 3 Figure Creation

Chapter 3 Figure Creation

The following code was used to create all the figures shown in chapter 3.

Parabolic mirror setup, Jones pupil, and ARM

The following code setups up the system for a parabolic mirror, traces rays that reflect off the parabolic mirror, and calculates the Jones pupil and ARM/PSF for the system

Mirror Setup

```
In[•]:= parabolicMirror = {
          CreateSurface[
           sys`surfID → 1,
           sys`surfaceLabel → "Parabolic Mirror",
           sys`shape`type → "Conic",
           sys`shape`curv → -1/25,
           sys`shape`conicK → -1,
           sys`v → {0, 0, 0},
           sys`a → {0, 0, 1},
           sys`material1 → "Air",
           sys`material2 → "Air",
           sys`mode → {"Reflect"},
           sys`aperture → (Norm[{#1, #2}] < 10 &),
           sys`coating`type → "Perfect",
           sys`coating`thickness → {1000},
           sys`coating`indices → {"UserIsotropic_aljeff"}],
          CreateSurface[
           sys`surfID → 2,
           sys`surfaceLabel → "Detector",
           sys`shape`type → "Plane",
           sys`v → {0, 0, -12.5},
           sys`a → {0, 0, -1},
           sys`aperture → (Norm[{#1, #2}] < 1 &),
           sys`mode → {"Absorb"}]};
```

Ray trace

```
In[•]:= config`rayID = 1;
        rays = CreateCollimatedRectRayGrid[20, 20, 51, 51, ray`λ → .5,
           ray`k → {0, 0, 1}, ray`surfaceOrder → {1, 2}, ray`r → {0, 0, -5}];
```

```
In[•]:= raysout = TraceRays[rays, parabolicMirror];
```

Jones pupil

```
In[•]:= {jp, xloc, yloc} = JonesPupil[{raysout, parabolicMirror, -12.5}];

        pupil = jp[[1]];
        absjones = Abs[pupil];
        argjones = Arg[pupil];
        labels = {{"XX", "XY"}, {"YX", "YY"}};

In[•]:= pupilticks = Subdivide[-10, 10, 4];

In[•]:= ampgrid =
         Grid[Table[ArrayPlot[absjones[[All, All, ii, jj]], PlotLegends → BarLegend[Automatic,
               LabelStyle → Directive[Black, Bold, 30]], (*Background→ Black,*)
            ColorFunction → GrayLevel, PlotLabel → Style["A"labels[[ii,jj]], Bold, Black, 40],
            ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
            Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
             {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 30]
           ], {ii, 2}, {jj, 2}], Alignment → Left]
```

Spacing in Jones pupil and PSF

```
        (*p606 in pl&os, spacing in psf*)
        (*Δs = λ*effl / (n Δx) = wavelength * focal length / (number of samples in exit pupil * exit pupil spacing)*)

        Δx = 0.4 * 10^-3;
        n = 511;
        λ = 500 * 10^-9;
        effl = 12.5(*mm*);
        Δs = λ * effl / ((n - 1) Δx) (*mm*);
```

PSF

```
In[•]:= psf = JonesPSF[jp, FTRatio → 10];

In[•]:= l = Length[psf[[1]]];

In[•]:= ξ = Range[-(l - 1)/2, (l - 1)/2];
        psflength = ξ * Δs;

In[•]:= ticks = Round[Subdivide[1, Length@psflength, 6]];
```

```
psfplotxx = ArrayPlot[Log10@(Abs[psf[[1, All, All, 1, 1]]]^2),
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}}, PlotLabel → Style["ARM_XX", Black, Bold, 40]]
```

In[•]:=
```
circ = Graphics[{Red, Thickness[.01], Circle[{0, 0}, 1.22 λ effl / .02]}]
```

```
psfplotxy = ArrayPlot[Chop@Abs[psf[[1, All, All, 1, 2]]]^2,
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → Automatic, PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["ARM_XY", Black, Bold, 40]]
```

In[•]:=
```
psfplotyx = ArrayPlot[Chop@Abs[psf[[1, All, All, 2, 1]]]^2,
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → Automatic, PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["ARM_YX", Black, Bold, 40]]
```

```
psfplotyy = ArrayPlot[Log10@(Abs[psf[[1, All, All, 2, 2]]]^2),
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}}, PlotLabel → Style["ARM_YY", Black, Bold, 40]]
```

```
psfgrid = Grid[{{Show[psfplotxx, circ], psfplotxy},
   {psfplotyx, Show[psfplotyy, circ]}}, Alignment → Left]
```

Use VV6 mask

The following code sets up the vector vortex charge 6 mask and applies it to the ARM at the coronagraph mask plane

In[•]:=
```
v11 = Table[Cos[6 * ArcTan[x, y]], {x, Subdivide[-1, 1, 510]}, {y, Subdivide[-1, 1, 510]}];
```

In[•]:=
```
v12 = Table[Sin[6 * ArcTan[x, y]], {x, Subdivide[-1, 1, 510]}, {y, Subdivide[-1, 1, 510]}];
v21 = v12;
v22 = -v11;
```

```
vv6 = Transpose[{{v11, v12}, {v21, v22}}, {3, 4, 1, 2}];
```

In[•]:=
```
masked = psf[[1]] * vv6;
```

In[•]:=
```
maskedlabel = {{"Masked ARM_XX", "Masked ARM_XY"}, {"Masked ARM_YX", "Masked ARM_YY"}};
maskedplot = Grid[Table[ArrayPlot[Chop@Abs@masked[[All, All, ii, jj]],
    DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
    Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
    FrameTicks → {{Round[psflength[[ticks]], 10^-3] // N, None},
      {Round[psflength[[ticks]], 10^-3] // N, None}},
    FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
    PlotRange → {{-.004, .004}, {-.004, .004}},
    PlotLabel → Style[maskedlabel[[ii, jj]], Black, Bold, 40]],
   {ii, 2}, {jj, 2}], Alignment → Left]
```

In[•]:=
```
masklabel = {{"VV6_XX", "VV6_XY"}, {"VV6_YX", "VV6_YY"}};
```

In[•]:=
```
maskplot =
  Grid[Table[ArrayPlot[vv6[[All, All, ii, jj]], DataRange → {{Min@psflength, Max@psflength},
      {Min@psflength, Max@psflength}}, Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
    FrameTicks → {{Round[psflength[[ticks]], 10^-3] // N, None},
      {Round[psflength[[ticks]], 10^-3] // N, None}},
    FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
    PlotRange → {{-.004, .004}, {-.004, .004}},
    PlotLabel → Style[masklabel[[ii, jj]], Black, Bold, 40]],
   {ii, 2}, {jj, 2}], Alignment → Left];
```

Masked JP_A, total system Jones pupil, and total system $|\text{ARM}|^2$

This code imports data for and makes plots of the Jones pupil distribution with the vector vortex applied, the total system Jones pupil, and the total system ARM

□ JP_A masked

In[•]:=
```
{real, imag} =
  Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 3\\maskedA.h5",
   {"Datasets", {"real", "imaginary"}}];
```

In[•]:=
```
jpAmasked = Transpose[real + ⅈ imag, {3, 4, 1, 2}];
```

In[•]:=
```
sub = Length[jpAmasked]/2;
```

```
δ = sub/4;
```

```
In[•]:= pupilticks2 = Subdivide[-20, 20, 4];

In[•]:= jpm = Grid[Table[ArrayPlot[Abs@jpAmasked[[sub - δ ;; sub + δ, sub - δ ;; sub + δ, ii, jj]],
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     (*Background→ Black,*)ColorFunction → GrayLevel,
     PlotLabel → Style["JP masked"labels[[ii,jj]], Bold, Black, 40],
     ImageSize → Large, DataRange → {{-20, 20}, {-20, 20}},
     Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
      {{pupilticks2, None}, {pupilticks2, None}}, FrameStyle → Directive[Black, Bold, 30]
    ], {ii, 2}, {jj, 2}], Alignment → Left]
```

Total Jones pupil with no mask

```
{real, imag} = Import[
   "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 3\\jp_no_mask.h5",
   {"Datasets", {"real", "imaginary"}}];
nomask = Transpose[real + i imag, {3, 4, 1, 2}];

jptotalnomask =
 Grid[Table[ArrayPlot[Abs@nomask[[sub - δ/2 ;; sub + δ/2, sub - δ/2 ;; sub + δ/2, ii, jj]],
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
    (*Background→ Black,*)ColorFunction → GrayLevel,
    PlotLabel → Style["Total JP no mask"labels[[ii,jj]], Bold, Black, 26],
    ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
    Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
     {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 20]
   ], {ii, 2}, {jj, 2}], Alignment → Left]
```

Total Jones pupil with mask

```
{real, imag} = Import[
   "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 3\\jp_with_mask.h5",
   {"Datasets", {"real", "imaginary"}}];
withmask = Transpose[real + i imag, {3, 4, 1, 2}];

In[•]:= jptotalwithmask =
  Grid[Table[ArrayPlot[Abs@withmask[[sub - δ/2 ;; sub + δ/2, sub - δ/2 ;; sub + δ/2, ii, jj]],
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     (*Background→ Black,*)ColorFunction → GrayLevel,
     PlotLabel → Style["Total JP with mask"labels[[ii,jj]], Bold, Black, 40],
     ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
     Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
      {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 30]
    ], {ii, 2}, {jj, 2}], Alignment → Left];
```

Setup spacing for ARM plots

```
R = 4;
lambdaOverD = (500 * 10^-9) / 0.02;
xvals = Subdivide[-R / lambdaOverD, R / lambdaOverD, 2^12];
xvals = xvals[[1 ;; -2]];
dx = Abs[xvals[[1]] - xvals[[2]]];
xi = Subdivide[-1 / (2 * dx), 1 / (2 * dx), 2^12];
xi = xi[[1 ;; -2]];
nomaskticks = Round[Subdivide[-10 λ effl / (.02 * .9), 10 λ effl / (.02 * .9), 4], 10^-4] // N
nomaskpsflabel =
  {{"|ARM|^2 No Mask_XX", "|ARM|^2 No Mask_Xy"}, {"|ARM|^2 No Mask_YX", "|ARM|^2 No Mask_YY"}};
withmaskpsflabel = {{"|ARM With Mask_XX|^2", "|ARM With Mask_Xy|^2"},
   {"|ARM With Mask_YX|^2", "|ARM With Mask_YY|^2"}};
maxnomask = Max[Abs[psfnomask]]
```

ARM with no mask

```
{real, imag} = Import[
   "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 3\\psf_no_mask.h5",
   {"Datasets", {"real", "imaginary"}}];
psfnomask = Transpose[real + i imag, {3, 4, 1, 2}];

nomaskpsfplot = Grid[Table[ArrayPlot[Chop@ Abs@psfnomask[[All, All, ii, jj]] / maxnomask, Frame → True,
    FrameLabel → {"y (mm)", "x (mm)"}, DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
    PlotRange → 10 λ effl / (.02 * .9) {{-1, 1}, {-1, 1}},
    FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
    FrameStyle → Directive[Black, Bold, 20], ImageSize → Large, ColorFunction → GrayLevel,
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
    PlotLabel → Style[nomaskpsflabel[[ii, jj]], Black, Bold, 20]],
   {ii, 2}, {jj, 2}], Alignment → Left]
```

ARM with mask

```
In[•]:= {real, imag} = Import[
   "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 3\\psf_with_mask.h5",
   {"Datasets", {"real", "imaginary"}}];
psfwithmask = Transpose[real + i imag, {3, 4, 1, 2}];
```

```
withmaskpsfplot =
 Grid[Table[ArrayPlot[Chop@ Abs@psfwithmask[[All, All, ii, jj]] / maxnomask, Frame → True,
    FrameLabel → {"y (mm)", "x (mm)"}, DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
    PlotRange → 10 λ effl / (.02 * .9) {{-1, 1}, {-1, 1}},
    FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
    FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
    PlotLabel → Style[withmaskpsflabel[[ii, jj]], Black, Bold, 40]],
   {ii, 2}, {jj, 2}], Alignment → Left]
```

Off-axis planet

This code recalculates the Jones pupils and ARMs (with and without the vector vortex mask) for an off-axis planet, simulated by rotating the incoming light's propagation vector by 5λ/D relative to the star's propagation vector.

☐ Setup rotation angle

```
tiltang = 5 λ / .02
```

```
tiltk = RotationMatrix[tiltang, {0, 1, 0}].{0, 0, 1}
```

☐ Trace tilted rays

```
config`rayID = 1;
tiltrays = CreateCollimatedRectRayGrid[20, 20, 51, 51,
   ray`λ → .5, ray`k → tiltk, ray`surfaceOrder → {1, 2}, ray`r → {0, 0, -5}];
tiltout = TraceRays[tiltrays, parabolicMirror];
```

☐ Jones pupil for tilted rays

The 'myJonesPupilWithExitPupil' function traces the tilted rays to the star's reference sphere instead of making a reference sphere for the off-axis planet. XP is the surface representing the star's reference sphere.

```
{tiltrays, tiltjp} = myJonesPupilWithExitPupil[parabolicMirror, tiltout, -12.5, 2, 51, XP];
```

```
In[•]:= ampgridtilt =
 Grid[Table[ArrayPlot[Round[Abs[tiltjp[[All, All, ii, jj]]]], PlotLegends → BarLegend[
       Automatic, LabelStyle → Directive[Black, Bold, 20]], (*Background→ Black,*)
    ColorFunction → GrayLevel, PlotLabel → Style["A"labels[[ii,jj]], Bold, Black, 26],
    ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
    Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
     {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 20]
   ], {ii, 2}, {jj, 2}], Alignment → Left]

In[•]:= arggridtilt =
 Grid[Table[ArrayPlot[argtilt[[All, All, ii, jj]], PlotLegends → BarLegend[Automatic,
       LabelStyle → Directive[Black, Bold, 20]], (*Background→ Black,*)
    ColorFunction → Hue, PlotLabel → Style["ϕ"labels[[ii,jj]], Bold, Black, 26],
    ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
    Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
     {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 20]
   ], {ii, 2}, {jj, 2}], Alignment → Left]
```

ARM for tilted rays

```
In[•]:= psftilt = JonesPSF[{tiltjp, 0.37354440480612294`,
     {0.`, 0.`, -1.7763568394002505`*^-15}, {0.`, 0.`, -1.`}, 0.5`}, FTRatio → 10];
```

```
psfplottiltxx = ArrayPlot[Log10@(Abs[psftilt[[1, All, All, 1, 1]]]),
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}}, PlotLabel → Style["ARM_XX", Black, Bold, 40]]
psfplottiltyy = ArrayPlot[Log10@(Abs[psftilt[[1, All, All, 2, 2]]]),
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}}, PlotLabel → Style["ARM_YY", Black, Bold, 40]]
psfplottiltxy = ArrayPlot[Chop@Abs[psftilt[[1, All, All, 1, 2]]],
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → Automatic, PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["ARM_XY", Black, Bold, 40]]
psfplottiltyx = ArrayPlot[Chop@Abs[psftilt[[1, All, All, 2, 1]]],
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → Automatic, PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["ARM_YX", Black, Bold, 40]]
psftiltgrid = Grid[{{psfplottiltxx, psfplottiltxy}, {psfplottiltyx, psfplottiltyy}},
  Alignment → Left]
```

Compare on-axis star PSF with off-axis planet PSF with differing flux ratios

```
In[ ]:= starplanetPSF =
 ArrayPlot[Log10[Abs[psf[[1, All, All, 1, 1]]]^2 + Abs[psftilt[[1, All, All, 1, 1]]]^2],
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["Star + Planet", Black, Bold, 30]]
```

```
In[•]:= starplanetPSF1 =
 ArrayPlot[Log10[Abs[psf[[1, All, All, 1, 1]]]^2 + .1 Abs[psftilt[[1, All, All, 1, 1]]]^2],
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["Star + Planet", Black, Bold, 30]]

In[•]:= starplanetPSF3 =
 ArrayPlot[Log10[Abs[psf[[1, All, All, 1, 1]]]^2 + .001 Abs[psftilt[[1, All, All, 1, 1]]]^2],
  DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
  Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
   {{Round[psflength[[ticks]], 10^-3] // N, None}, {Round[psflength[[ticks]], 10^-3] // N, None}},
  FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  PlotRange → {{-.004, .004}, {-.004, .004}},
  PlotLabel → Style["Star + Planet", Black, Bold, 30]]
```

Apply vector vortex

```
In[•]:= maskedtilt = psftilt[[1]] * vv6;

In[•]:= maskedtiltplot = Grid[Table[ArrayPlot[Chop@Abs@maskedtilt[[All, All, ii, jj]],
     DataRange → {{Min@psflength, Max@psflength}, {Min@psflength, Max@psflength}},
     Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
     FrameTicks → {{Round[psflength[[ticks]], 10^-3] // N, None},
       {Round[psflength[[ticks]], 10^-3] // N, None}},
     FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     PlotRange → {{-.004, .004}, {-.004, .004}},
     PlotLabel → Style[maskedlabel[[ii, jj]], Black, Bold, 40]],
    {ii, 2}, {jj, 2}], Alignment → Left]
```

JP_A masked for tilted rays

```
In[•]:= {real, imag} = Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
      3\\jpa_mask_tilted.h5", {"Datasets", {"real", "imaginary"}}];

In[•]:= jpAmaskedtilt = Transpose[real + i imag, {3, 4, 1, 2}];

In[•]:= jpmtilt = Grid[Table[ArrayPlot[Chop@Abs@jpAmaskedtilt[[All, All, ii, jj]],
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     (*Background→ Black,*)ColorFunction → GrayLevel,
     PlotLabel → Style["Planet JP masked"labels[[ii,jj]], Bold, Black, 40],
     ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
     Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
      {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 30]
    ], {ii, 2}, {jj, 2}], Alignment → Left]
```

End-to-end Jones pupil

```
In[•]:= {real, imag} = Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
          3\\totl_mask_tilted.h5", {"Datasets", {"real", "imaginary"}}];
        withmask = Transpose[real + i imag, {3, 4, 1, 2}];
```

```
In[•]:= jptotalwithmasktilted = Grid[Table[ArrayPlot[Chop@Abs@withmask[[All, All, ii, jj]],
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          (*Background→ Black,*)ColorFunction → GrayLevel,
          PlotLabel → Style["Planet Total JP with mask"labels[[ii,jj]], Bold, Black, 40],
          ImageSize → Large, DataRange → {{-10, 10}, {-10, 10}},
          Frame → True, FrameLabel → {"y (mm)", "x (mm)"}, FrameTicks →
           {{pupilticks, None}, {pupilticks, None}}, FrameStyle → Directive[Black, Bold, 30]
         ], {ii, 2}, {jj, 2}], Alignment → Left]
```

PSF with mask

```
In[•]:= {real, imag} = Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
          3\\total_psf_tilted.h5", {"Datasets", {"real", "imaginary"}}];
        psfwithmasktilt = Transpose[real + i imag, {3, 4, 1, 2}];
```

```
In[•]:= withmaskpsfplot =
         Grid[Table[ArrayPlot[If[ii + jj == 3, Chop@ Abs@psfwithmasktilt[[All, All, ii, jj]] / maxnomask,
           Chop@Log10@Reverse[Transpose[ Abs@psfwithmasktilt[[All, All, ii, jj]] / maxnomask], {2}]],
          Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
          DataRange → {{xi[[sub - δ]], xi[[sub + δ]]}, {xi[[sub - δ]], xi[[sub + δ]]}},
          PlotRange → 10 λ effl / .02 {{-1, 1}, {-1, 1}},
          FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
          FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style[withmaskpsflabel[[ii, jj]], Black, Bold, 40]],
         {ii, 2}, {jj, 2}], Alignment → Left]
```

Compare on-axis star with VV6 and off-axis planet with VV6 for differing flux ratios

```
In[•]:= starPlanetpsf0 = ArrayPlot[
          Log10[ 1/maxnomask (Chop@Reverse[Transpose[Abs@psfwithmasktilt[[All, All, 1, 1]]], {2}] +
             psfwithmask[[sub - δ ;; sub + δ, sub - δ ;; sub + δ, 1, 1]]) ],
          Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
          DataRange → {{xi[[sub - δ]], xi[[sub + δ]]}, {xi[[sub - δ]], xi[[sub + δ]]}},
          PlotRange → 10 λ effl/.02 {{-1, 1}, {-1, 1}},
          FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
          FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style["Star + Planet", Black, Bold, 40]]

In[•]:= starPlanetpsf4 = ArrayPlot[
          Log10[ 1/maxnomask (10^-4 Chop@Reverse[Transpose[Abs@psfwithmasktilt[[All, All, 1, 1]]], {2}] +
             psfwithmask[[sub - δ ;; sub + δ, sub - δ ;; sub + δ, 1, 1]]) ],
          Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
          DataRange → {{xi[[sub - δ]], xi[[sub + δ]]}, {xi[[sub - δ]], xi[[sub + δ]]}},
          PlotRange → 10 λ effl/.02 {{-1, 1}, {-1, 1}},
          FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
          FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style["Star + Planet", Black, Bold, 40]]

In[•]:= starPlanetpsf7 = ArrayPlot[
          Log10[ 1/maxnomask (10^-7 Chop@Reverse[Transpose[Abs@psfwithmasktilt[[All, All, 1, 1]]], {2}] +
             psfwithmask[[sub - δ ;; sub + δ, sub - δ ;; sub + δ, 1, 1]]) ],
          Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
          DataRange → {{xi[[sub - δ]], xi[[sub + δ]]}, {xi[[sub - δ]], xi[[sub + δ]]}},
          PlotRange → 10 λ effl/.02 {{-1, 1}, {-1, 1}},
          FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
          FrameStyle → Directive[Black, Bold, 30], ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style["Star + Planet", Black, Bold, 40]]
```

```
In[ ]:= starPlanetpsf10 = ArrayPlot[Log10[ 1/maxnomask
          (10^-10 Chop@Reverse[Transpose[Abs@psfwithmasktilt[[All, All, 1, 1]]], {2}] +
            psfwithmask[[sub - δ ;; sub + δ, sub - δ ;; sub + δ, 1, 1]])],
        Frame → True, FrameLabel → {"y (mm)", "x (mm)"},
        DataRange → {{xi[[sub - δ]], xi[[sub + δ]]}, {xi[[sub - δ]], xi[[sub + δ]]}},
        PlotRange → 10 λ effl/.02 {{-1, 1}, {-1, 1}},
        FrameTicks → {{nomaskticks, None}, {nomaskticks, None}},
        FrameStyle → Directive[Black, Bold, 20], ImageSize → Large, ColorFunction → GrayLevel,
        PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
        PlotLabel → Style["|Star_xx|^2 + |Planet_xx|^2", Black, Bold, 20]]
```

APPENDIX G

Chapter 4 Figure Creation

Chapter 4 Figure Creation

The following code was used to create the figures shown in chapter 4.

The first set of code creates the figures for each of these using 500 nm light:
-the ARM at the coronagraph mask plane
-the product of the vector vortex and ARM at the coronagraph mask plane
-the Jones pupil distribution with the vector vortex applied (JPA_masked)
-the Jones pupil for system A (fore-optics)
-the Jones pupil for system B (post-optics)
-total Jones pupil with vector vortex
-total Jones pupil without vector vortex
-PSF with vector vortex
-PSF without vector vortex

The second set of code creates the figures for each of the following for polychromatic light ([450, 475, 500, 525, 550] nm light):
-PSF with no mask and ideal apertures (no aberrations)
-PSF with VV6 mask and ideal apertures (no aberrations)
-PSF with no mask and isotropic coatings
-PSF with VV6 mask and isotropic coatings
-PSF with no mask and isotropic coatings and form birefringence on primary mirror
-PSF with VV6 mask and isotropic coatings and form birefringence on primary mirror
-2D contrast for the ideal, isotropic, and anisotropic cases
-Horizontal and vertical slices through the 2D contrast plots

500 nm light

Setup the object spacing angle on sky

```
n = 2^12;
np = 2^10;
r = n / np;
lambdaOverD = (*omitted due to ITAR*)
xvals = Subdivide[-r / lambdaOverD, r / lambdaOverD, n];
xvals = xvals[[1 ;; -2]];
dx = Abs[xvals[[1]] - xvals[[2]]];
xi = Subdivide[-1 / (2 * dx), 1 / (2 * dx), n];
xi = xi[[1 ;; -2]];
xi = xi * 206 265;
xi = xi[[(n - np)/2 ;; (n + np)/2]];
```

ARM at coronagraph mask plane

```
In[•]:= {re, im} = Import[
          "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\arma_500.h5",
          {"Datasets", {"real", "imaginary"}}];
        arma = Transpose[re + i im, {3, 4, 1, 2}];


In[•]:= ticksh = Subdivide[-.5, .5, 4];

In[•]:= armlabel = {{"ARM_XX", "ARM_XY"}, {"ARM_YX", "ARM_YY"}};

In[•]:= range3 = {{{-6, 0}, {-8, -2}}, {{-8, -2}, {-6, 0}}};

In[•]:= armaplot = Grid[Table[ArrayPlot[Transpose[Reverse[Log10[Abs[arma[[All, All, ii, jj]]]/Max[Abs[arma]]]]],
            Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
            DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
            PlotRange → {{-.5, .5}, {-.5, .5}, range3[[ii, jj]]},
            FrameTicks → {{ticksh, None}, {ticksh, None}},
            FrameStyle → Directive[Black, Bold, 30], ColorFunction → GrayLevel,
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 500, PlotLabel → Style[armlabel[[ii, jj]], Black, Bold, 40]],
          {ii, 2}, {jj, 2}], Alignment → Left]
```

Vector vortex charge 6

```
In[•]:= vv6data = Table[
          {{Cos[6 ArcTan[x, y]], Sin[6 ArcTan[x, y]]}, {Sin[6 ArcTan[x, y]], Cos[6 ArcTan[x, y]]}},
          {x, Subdivide[-1, 1, 1024]}, {y, Subdivide[-1, 1, 1024]}];
```

```
vv6data[[513, 513]] = {{0, 0}, {0, 0}};

vv6data2 = RotateRight[vv6data[[All, All, 1, 1]], {1, 1}];
vv6data3 = RotateRight[vv6data[[All, All, 1, 2]], {1, 1}];
vv6data4 = RotateRight[vv6data[[All, All, 2, 1]], {1, 1}];
vv6data5 = RotateRight[vv6data[[All, All, 2, 2]], {1, 1}];

vv6total = {{vv6data2, vv6data3}, {vv6data4, vv6data5}};

vv6arm = arma * Transpose[vv6total, {3, 4, 1, 2}];

maskedarmlabel = {{"Masked ARM_XX", "Masked ARM_XY"}, {"Masked ARM_YX", "Masked ARM_YY"}};

vv6armaplot =
  Grid[Table[ArrayPlot[Log10[Transpose[Reverse[Abs[vv6arm[[All, All, ii, jj]]] / Max[Abs[vv6arm]]]]] /.
      Indeterminate → None, Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
     DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
     PlotRange → {{-.5, .5}, {-.5, .5}(*,range3[[ii,jj]]*)},
     FrameTicks → {{ticksh, None}, {ticksh, None}},
     FrameStyle → Directive[Black, Bold, 30], ColorFunction → GrayLevel,
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     ImageSize → 500, PlotLabel → Style[maskedarmlabel[[ii, jj]], Black, Bold, 40]],
    {ii, 2}, {jj, 2}], Alignment → Left];
```

Jones pupil distribution with the vector vortex

```
{re, im} = Import[
   "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\masked_jp_500.h5",
   {"Datasets", {"real", "imaginary"}}];
ring = Transpose[re + i im, {3, 4, 1, 2}];

absringlabel = {{"A_XX", "A_XY"}, {"A_YX", "A_YY"}};
argringlabel = {{"φ_XX", "φ_XY"}, {"φ_YX", "φ_YY"}};

absringplot =
  Grid[Table[ArrayPlot[Transpose[Reverse[Abs[ring[[All, All, ii, jj]]]]], Frame → False,
     ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     ImageSize → 300, PlotLabel → Style[absringlabel[[ii, jj]], White, Bold, 40],
     Background → Black], {ii, 2}, {jj, 2}], Alignment → Left];

argringplot =
  Grid[Table[ArrayPlot[Transpose[Reverse[Arg[ring[[All, All, ii, jj]]]]], Frame → False,
     ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
     PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
     ImageSize → 300, PlotLabel → Style[argringlabel[[ii, jj]], White, Bold, 40],
     Background → Black], {ii, 2}, {jj, 2}], Alignment → Left];
```

Jones pupil A without vector vortex

```
In[•]:= {re, im} = Import["C:\\Users\\jeffdavis\\Documents\\HabEx Pupils white light\\10-30-18
         prescription\\Correct JP\\jpA 500 nm.h5", {"Datasets", {"real", "imaginary"}}];

In[•]:= pupilA = re + i im;

In[•]:= absA = Replace[Abs[pupilA], t_ /; t == 0. → None, {4}];
        argA = Replace[Arg[pupilA], t_ /; t == 0. → None, {4}];

In[•]:= absAplot = Grid[Table[ArrayPlot[absA[[All, All, ii, jj]], Frame → False,
            ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 300, PlotLabel → Style[absringlabel[[ii, jj]], White, Bold, 40],
            Background → Black], {ii, 2}, {jj, 2}], Alignment → Left]

In[•]:= argAplot = Grid[Table[ArrayPlot[argA[[All, All, ii, jj]], Frame → False,
            ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 300, PlotLabel → Style[argringlabel[[ii, jj]], White, Bold, 40],
            Background → Black], {ii, 2}, {jj, 2}], Alignment → Left]
```

Jones pupil B

```
In[•]:= {re, im} = Import["C:\\Users\\jeffdavis\\Documents\\HabEx Pupils white light\\10-30-18
         prescription\\Correct JP\\jpB 500 nm.h5", {"Datasets", {"real", "imaginary"}}];

In[•]:= pupilB = re + i im;

In[•]:= absB = Replace[Abs[pupilB], t_ /; t == 0. → None, {4}];
        argB = Replace[Arg[pupilB], t_ /; t == 0. → None, {4}];

In[•]:= absBplot = Grid[Table[ArrayPlot[absB[[All, All, ii, jj]], Frame → False,
            ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 300, PlotLabel → Style[absringlabel[[ii, jj]], White, Bold, 40],
            Background → Black], {ii, 2}, {jj, 2}], Alignment → Left]

In[•]:= argBplot = Grid[Table[ArrayPlot[argB[[All, All, ii, jj]], Frame → False,
            ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 300, PlotLabel → Style[argringlabel[[ii, jj]], White, Bold, 40],
            Background → Black], {ii, 2}, {jj, 2}], Alignment → Left]
```

Total system Jones pupil with vector vortex

```
In[•]:= {re, im} = Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
         4\\total_pupil_with_mask.h5", {"Datasets", {"real", "imaginary"}}];

In[•]:= jptotMask = Transpose[re + i im, {3, 4, 1, 2}];
```

```
In[•]:= abstot = Replace[Abs[jptotMask], t_ /; t == 0. → None, {4}];
        argtot = Replace[Arg[jptotMask], t_ /; t == 0. → None, {4}];

In[•]:= abstotplot =
          Grid[Table[ArrayPlot[Transpose[Reverse[abstot[[All, All, ii, jj]]]], Frame → False,
              ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
              PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
              ImageSize → 300, PlotLabel → Style[absringlabel[[ii, jj]], White, Bold, 40],
              Background → Black], {ii, 2}, {jj, 2}], Alignment → Left];

In[•]:= argtotplot =
          Grid[Table[ArrayPlot[Transpose[Reverse[argtot[[All, All, ii, jj]]]], Frame → False,
              ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
              PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
              ImageSize → 300, PlotLabel → Style[argringlabel[[ii, jj]], White, Bold, 40],
              Background → Black], {ii, 2}, {jj, 2}], Alignment → Left];
```

Total system Jones pupil without vector vortex

```
In[•]:= jptotnomask = Table[pupilB[[ii, jj]].pupilA[[ii, jj]], {ii, 101}, {jj, 101}];

In[•]:= abstotnomask = Replace[Abs[jptotnomask], t_ /; t == 0. → None, {4}];
        argtotnomask = Replace[Arg[jptotnomask], t_ /; t == 0. → None, {4}];

In[•]:= abstotnomaskplot = Grid[Table[ArrayPlot[abstotnomask[[All, All, ii, jj]], Frame → False,
            ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 300, PlotLabel → Style[absringlabel[[ii, jj]], White, Bold, 40],
            Background → Black], {ii, 2}, {jj, 2}], Alignment → Left]

In[•]:= argtotnomaskplot = Grid[Table[ArrayPlot[argtotnomask[[All, All, ii, jj]], Frame → False,
            ColorFunction → (Blend[{Black, Blue, Green, Red, Purple}, #] &),
            PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
            ImageSize → 300, PlotLabel → Style[argringlabel[[ii, jj]], White, Bold, 40],
            Background → Black], {ii, 2}, {jj, 2}], Alignment → Left]
```

PSF with vector vortex mask

```
In[•]:= psftotMask = Transpose[
           Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
              psfs\\isotropic_psf_with_mask_500.h5", {"Datasets", {"real"}}], {3, 4, 1, 2}];

In[•]:= range4 = {{{-5, 0}, {-9, -3}}, {{-9, -3}, {-5, 0}}};

In[•]:= totlabel = {{"|Total Masked ARM_XX|^2", "|Total Masked ARM_XY|^2"},
           {"|Total Masked ARM_YX|^2", "|Total Masked ARM_YY|^2"}};
```

```
psftotMaskPlot =
 Grid[Table[ArrayPlot[Transpose[Reverse[Log10[psftotMask[[All, All, ii, jj]] / Max[psftotMask]]]],
    Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
    DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
    PlotRange → {{-.5, .5}, {-.5, .5}, range4[[ii, jj]]},
    FrameTicks → {{ticksh, None}, {ticksh, None}},
    FrameStyle → Directive[Black, Bold, 30], ColorFunction → GrayLevel,
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
    ImageSize → 500, PlotLabel → Style[totlabel[[ii, jj]], Black, Bold, 30]],
   {ii, 2}, {jj, 2}], Alignment → Left]
```

```
sum = psftotMask[[All, All, 1, 1]] + psftotMask[[All, All, 1, 2]] +
   psftotMask[[All, All, 2, 1]] + psftotMask[[All, All, 2, 2]];
```

```
sumpsftotMaskPlot = ArrayPlot[Transpose[Reverse[Log10[sum / Max[sum]]]],
  Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
  DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
  PlotRange → {{-.5, .5}, {-.5, .5}, {-6, 0}},
  FrameTicks → {{ticksh, None}, {ticksh, None}},
  FrameStyle → Directive[Black, Bold, 30], ColorFunction → GrayLevel,
  PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
  ImageSize → 500, PlotLabel → Style["PSF with Mask", Black, Bold, 40]]
```

PSF with no vector vortex mask

```
psftotnoMask = Transpose[
   Import["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
      psfs\\isotropic_psf_no_mask_500.h5", {"Datasets", {"real"}}], {3, 4, 1, 2}];
```

```
range5 = {{{-12, 0}, {-15, -4}}, {{-15, -4}, {-12, 0}}};
```

```
ticksh = Subdivide[-.5, .5, 4];
```

```
totnomasklabel =
  {{"|No Mask ARM_XX|^2", "|No Mask ARM_XY|^2"}, {"|No Mask ARM_YX|^2", "|No Mask ARM_YY|^2"}};
```

```
psftotnoMaskPlot =
 Grid[Table[ArrayPlot[Transpose[Reverse[Log10[psftotnoMask[[All, All, ii, jj]] / Max[psftotnoMask]]]],
    Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
    DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
    PlotRange → {{-.5, .5}, {-.5, .5}, range5[[ii, jj]]},
    FrameTicks → {{ticksh, None}, {ticksh, None}},
    FrameStyle → Directive[Black, Bold, 30], ColorFunction → GrayLevel,
    PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
    ImageSize → 500, PlotLabel → Style[totnomasklabel[[ii, jj]], Black, Bold, 40]],
   {ii, 2}, {jj, 2}], Alignment → Left]
```

```
sumnomask = psftotnoMask[[All, All, 1, 1]] + psftotnoMask[[All, All, 1, 2]] +
   psftotnoMask[[All, All, 2, 1]] + psftotnoMask[[All, All, 2, 2]];
```

```
In[•]:= sumpsftotnoMaskPlot = ArrayPlot[Transpose[Reverse[Log10[sumnomask / Max[sumnomask]]]],
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {-12, 0}},
          FrameTicks → {{ticksh, None}, {ticksh, None}},
          FrameStyle → Directive[Black, Bold, 30], ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          ImageSize → 500, PlotLabel → Style["PSF No Mask", Black, Bold, 40]]
```

Polychromatic light

Function to plot and return psf for a single wavelength

```
In[•]:= psfplots[filepath_, label_, range_] := Module[{n, np, r, psf, lambdaOverD,
           data, xvals, dx, xi, ticks, plot, xxplot, yyplot, xyplot, yxplot},
         n = 2^12;
         np = 2^10;
        r = n / np;
        lambdaOverD = (500 * 10^-9) / 4;
        xvals = Subdivide[-r / lambdaOverD, r / lambdaOverD, n];
        xvals = xvals[[1 ;; -2]];
        dx = Abs[xvals[[1]] - xvals[[2]]];
        xi = Subdivide[-1 / (2 * dx), 1 / (2 * dx), n];
        xi = xi[[1 ;; -2]];
        xi = xi * 206 265;
         xi = xi[[(n - np)/2 ;; (n + np)/2]];
         ticks = Subdivide[-.5, .5, 4] // N;
         data = Import[filepath, {"Datasets", {"real"}}];
         data = Transpose[data, {3, 4, 1, 2}];
         psf =
          data[[All, All, 1, 1]] + data[[All, All, 1, 2]] + data[[All, All, 2, 1]] + data[[All, All, 2, 2]];
         plot = ArrayPlot[Log10@psf, Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
           DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
           PlotRange → {{-.5, .5}, {-.5, .5}, range},
           FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 20],
           ImageSize → Large, ColorFunction → GrayLevel,
           PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
           PlotLabel → Style[label, Black, Bold, 20]];
         Return[{psf, plot}];
        ]
```

Ideal no mask

```
In[•]:= {data450, plot450} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
             psfs\\ideal_psf_no_mask_450.h5", "Ideal PSF\nλ = 450 nm", {16, 28}];

In[•]:= {data475, plot475} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
             psfs\\ideal_psf_no_mask_475.h5", "Ideal PSF\nλ = 475 nm", {16, 28}];

In[•]:= {data500, plot500} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
             psfs\\ideal_psf_no_mask_500.h5", "Ideal PSF\nλ = 500 nm", {16, 28}];

In[•]:= {data525, plot525} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
             psfs\\ideal_psf_no_mask_525.h5", "Ideal PSF\nλ = 525 nm", {16, 28}];

In[•]:= {data550, plot550} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
             psfs\\ideal_psf_no_mask_550.h5", "Ideal PSF\nλ = 550 nm", {16, 28}];

In[•]:= polyidealpsf = data450 + data475 + data500 + data525 + data550;
        polyidealplot =
          ArrayPlot[Log10@polyidealpsf, Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
           DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
           PlotRange → {{-.5, .5}, {-.5, .5}, {16, 29}},
           FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 30],
           ImageSize → Large, ColorFunction → GrayLevel,
           PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
           PlotLabel → Style["Ideal PSF\n[450-550] nm", Black, Bold, 40]];
```

Ideal with mask

```
In[•]:= {data450, plot450} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
             4\\HabEx psfs\\ideal_psf_with_mask_450.h5",
           "Ideal Apertures with VV6\nλ = 450 nm", {12, 20}];
```

```
In[•]:= {data475, plot475} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\ideal_psf_with_mask_475.h5",
            "Ideal Apertures with VV6\nλ = 475 nm", {12, 20}];
        {data500, plot500} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\ideal_psf_with_mask_500.h5",
            "Ideal Apertures with VV6\nλ = 500 nm", {12, 20}];
        {data525, plot525} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\ideal_psf_with_mask_525.h5",
            "Ideal Apertures with VV6\nλ = 525 nm", {12, 20}];
        {data550, plot550} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\ideal_psf_with_mask_550.h5",
            "Ideal Apertures with VV6\nλ = 550 nm", {12, 20}];
        polyidealmaskpsf = data450 + data475 + data500 + data525 + data550;

In[•]:= polyidealplot = ArrayPlot[Log10@polyidealmaskpsf,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {12, 20}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 30],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style["Ideal Apertures with VV6\n[450-550] nm", Black, Bold, 40]]
```

□ Isotropic no mask

```
In[•]:= {data450, plot450} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter 4\\HabEx
              psfs\\isotropic_psf_no_mask_450.h5", "Isotropic Surfaces\nλ = 450 nm", {18, 30}];

In[•]:= {data475, plot475} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_no_mask_475.h5",
            "Isotropic Surfaces\nλ = 475 nm", {18, 30}];
        {data500, plot500} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_no_mask_500.h5",
            "Isotropic Surfaces\nλ = 500 nm", {18, 30}];
        {data525, plot525} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_no_mask_525.h5",
            "Isotropic Surfaces\nλ = 525 nm", {18, 30}];
        {data550, plot550} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_no_mask_550.h5",
            "Isotropic Surfaces\nλ = 550 nm", {18, 30}];
        polyisotropicpsf = data450 + data475 + data500 + data525 + data550;
```

```
In[•]:= polyidealplot = ArrayPlot[Log10@polyisotropicpsf,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {18, 30}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 20],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
          PlotLabel → Style["Isotropic Surfaces\n[450-550] nm", Black, Bold, 20]]
```

□ Isotropic with mask

```
In[•]:= {data450, plot450} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_with_mask_450.h5",
           "Isotropic Surfaces with VV6\nλ = 450 nm", {16, 25}];
```

```
In[•]:= {data475, plot475} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_with_mask_475.h5",
           "Isotropic Surfaces with VV6\nλ = 475 nm", {16, 25}];
        {data500, plot500} = psfplots[
           "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_with_mask_500.h5",
           "Isotropic Surfaces with VV6\nλ = 500 nm", {16, 25}];
        {data525, plot525} = psfplots[
           "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_with_mask_525.h5",
           "Isotropic Surfaces with VV6\nλ = 525 nm", {16, 25}];
        {data550, plot550} = psfplots[
           "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\isotropic_psf_with_mask_550.h5",
           "Isotropic Surfaces with VV6\nλ = 550 nm", {16, 25}];
        polyisotropicmaskpsf = data450 + data475 + data500 + data525 + data550;
```

```
In[•]:= polyidealplot = ArrayPlot[Log10@polyisotropicmaskpsf,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {16, 25}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 20],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
          PlotLabel → Style["Isotropic Surfaces with VV6\n[450-550] nm", Black, Bold, 20]]
```

□ AnIsotropic no mask

```
In[•]:= {data450, plot450} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_no_mask_450.h5",
           "Anisotropic Surfaces\nλ = 450 nm", {16, 29}];
```

```
In[•]:= {data475, plot475} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_no_mask_475.h5",
            "Anisotropic Surfaces\nλ = 475 nm", {16, 29}];
        {data500, plot500} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_no_mask_500.h5",
            "Anisotropic Surfaces\nλ = 500 nm", {16, 29}];
        {data525, plot525} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_no_mask_525.h5",
            "Anisotropic Surfaces\nλ = 525 nm", {16, 29}];
        {data550, plot550} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_no_mask_550.h5",
            "Anisotropic Surfaces\nλ = 550 nm", {16, 29}];
        polyanisotropicpsf = data450 + data475 + data500 + data525 + data550;

In[•]:= polyidealplot = ArrayPlot[Log10@polyanisotropicpsf,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {16, 29}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 20],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
          PlotLabel → Style["Anisotropic Surfaces\n[450-550] nm", Black, Bold, 20]]
```

□ AnIsotropic with mask

```
In[•]:= {data450, plot450} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_with_mask_450.h5",
            "Anisotropic Surfaces with VV6\nλ = 450 nm", {15, 25}];

In[•]:= {data475, plot475} =
          psfplots["C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_with_mask_475.h5",
            "Anisotropic Surfaces\nλ = 475 nm", {15, 25}];
        {data500, plot500} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_with_mask_500.h5",
            "Anisotropic Surfaces\nλ = 500 nm", {15, 25}];
        {data525, plot525} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_with_mask_525.h5",
            "Anisotropic Surfaces\nλ = 525 nm", {15, 25}];
        {data550, plot550} = psfplots[
            "C:\\Users\\jeffdavis\\Documents\\Dissertation\\Figures\\Chapter
              4\\HabEx psfs\\anisotropic_psf_with_mask_550.h5",
            "Anisotropic Surfaces\nλ = 550 nm", {15, 25}];
        polyanisotropicmaskpsf = data450 + data475 + data500 + data525 + data550;
```

```
In[•]:= polyidealplot = ArrayPlot[Log10@polyanisotropicmaskpsf,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {15, 25}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 20],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 20]],
          PlotLabel → Style["Anisotropic Surfaces\n[450-550] nm", Black, Bold, 20]]
```

Contrast 2D

```
In[•]:= idealcontrast = polyidealmaskpsf / Max[polyidealpsf];
```

```
In[•]:= idealcontrastplot =
         ArrayPlot[Log10@idealcontrast, Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {-14, -3}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 30],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]], PlotLabel →
           Style["Ideal Contrast\n[450-550] nm", Black, Bold, 40], ClippingStyle → Black]
```

```
In[•]:= isotropiccontrast = polyisotropicmaskpsf / Max[polyisotropicpsf];
```

```
In[•]:= isotropiccontrastplot = ArrayPlot[Log10@isotropiccontrast,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {-14, -3}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 30],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style["Isotropic Contrast\n[450-550] nm", Black, Bold, 40]]
```

```
In[•]:= anisotropiccontrast = polyanisotropicmaskpsf / Max[polyanisotropicpsf];
```

```
In[•]:= anisotropiccontrastplot = ArrayPlot[Log10@anisotropiccontrast,
          Frame → True, FrameLabel → {"y (arcsec)", "x (arcsec)"},
          DataRange → {{xi[[1]], xi[[-1]]}, {xi[[1]], xi[[-1]]}},
          PlotRange → {{-.5, .5}, {-.5, .5}, {-14, -3}},
          FrameTicks → {{ticks, None}, {ticks, None}}, FrameStyle → Directive[Black, Bold, 30],
          ImageSize → Large, ColorFunction → GrayLevel,
          PlotLegends → BarLegend[Automatic, LabelStyle → Directive[Black, Bold, 30]],
          PlotLabel → Style["Anisotropic Contrast\n[450-550] nm", Black, Bold, 40]]
```

Contrast 1D slice

```
In[•]:= hcontrast1D =
  ListPlot[{Transpose[{xi, Log10@idealcontrast[[Length@idealcontrast + 1/2, All]]}],
    Transpose[{xi, Log10@isotropiccontrast[[Length@idealcontrast + 1/2, All]]}],
    Transpose[{xi, Log10@anisotropiccontrast[[Length@idealcontrast + 1/2, All]]}]},
   Frame → True, FrameStyle → Directive[Black, Bold, 30], PlotLegends →
    Placed[{Style["Ideal Apertures", Bold, 30, Black], Style["Isotropic Surfaces",
       Bold, 30, Black], Style["Anisotropic Primary", Bold, 30, Black]}, Top],
   ImageSize → 750, Joined → True, FrameLabel → {"x (arcsec)", "Contrast"},
   PlotRange → {{0, .5}, {-13, -3}}, PlotMarkers → {"x", "o", "+"},
   PlotStyle → {Directive[Black, Thick], Directive[Red, Thick], Directive[Blue, Thick]}]

In[•]:= vcontrast1D =
  ListPlot[{Transpose[{xi, Log10@idealcontrast[[All, Length@idealcontrast + 1/2]]}],
    Transpose[{xi, Log10@isotropiccontrast[[All, Length@idealcontrast + 1/2]]}],
    Transpose[{xi, Log10@anisotropiccontrast[[All, Length@idealcontrast + 1/2]]}]},
   Frame → True, FrameStyle → Directive[Black, Bold, 30], PlotLegends →
    Placed[{Style["Ideal Apertures", Bold, 30, Black], Style["Isotropic Surfaces",
       Bold, 30, Black], Style["Anisotropic Primary", Bold, 30, Black]}, Above],
   ImageSize → 750, Joined → True, FrameLabel → {"y (arcsec)", "Contrast"},
   PlotRange → {{0, .5}, {-13, -3}}, PlotMarkers → {"x", "o", "+"},
   PlotStyle → {Directive[Black, Thick], Directive[Red, Thick], Directive[Blue, Thick]}]
```

REFERENCES

[1] Big Questions. https://science.nasa.gov/astrophysics/big-questions. Accessed: 2019-04-24.

[2] 2020 Decadal Survey. https://science.nasa.gov/astrophysics/2020-decadal-survey-planning. Accessed: 2019-04-24.

[3] NASA Exoplanet Archive. https://exoplanetarchive.ipac.caltech.edu/. Accessed: 2019-04-24.

[4] Andrew Cumming. Detectability of extrasolar planets in radial velocity surveys. *Monthly Notices of the Royal Astronomical Society*, 354(4):1165–1176, 2004.

[5] Jason T WrightDr and B Scott GaudiAssociate Professor. Exoplanet detection methods. *Planets, Stars and Stellar Systems: Volume 3: Solar and Stellar Planetary Systems*, pages 489–540, 2013.

[6] A Reiners, JL Bean, KF Huber, S Dreizler, A Seifahrt, and S Czesla. Detecting planets around very low mass stars with the radial velocity method. *The Astrophysical Journal*, 710(1):432, 2010.

[7] Francesco Pepe, David Ehrenreich, and Michael R Meyer. Instrumentation for the detection and characterization of exoplanets. *Nature*, 513(7518):358, 2014.

[8] RL Akeson, X Chen, D Ciardi, M Crane, J Good, M Harbut, E Jackson, SR Kane, AC Laity, S Leifer, et al. The nasa exoplanet archive: data and tools for exoplanet research. *Publications of the Astronomical Society of the Pacific*, 125(930):989, 2013.

[9] Michael Perryman, Joel Hartman, Gáspár Á Bakos, and Lennart Lindegren. Astrometric exoplanet detection with gaia. *The Astrophysical Journal*, 797(1):14, 2014.

[10] J-P Beaulieu, David P Bennett, Pascal Fouqué, Andrew Williams, Martin Dominik, UG Jørgensen, Daniel Kubas, Arnaud Cassan, Christian Coutures, John Greenhill, et al. Discovery of a cool planet of 5.5 earth masses through gravitational microlensing. *Nature*, 439(7075):437, 2006.

[11] Wesley A Traub and Ben R Oppenheimer. Direct imaging of exoplanets. *Exoplanets*, pages 111–156, 2010.

[12] John E Krist, Bijan Nemati, and Bertrand P Mennesson. Numerical modeling of the proposed wfirst-afta coronagraphs and their predicted performances. *Journal of Astronomical Telescopes, Instruments, and Systems*, 2(1):011003, 2015.

[13] John T Trauger, Dwight C Moody, John E Krist, and Brian L Gordon. Hybrid lyot coronagraph for wfirst-afta: coronagraph design and performance metrics. *Journal of Astronomical Telescopes, Instruments, and Systems*, 2(1):011013, 2016.

[14] Michael Galvin, Yunjong Kim, N Jeremy Kasdin, Dan Sirbu, Robert Vanderbei, Dan Echeverri, Giuseppe Sagolla, Andreas Rousing, Kunjithapatham Balasubramanian, Daniel Ryan, et al. Design and construction of a 76m long-travel laser enclosure for a space occulter testbed. In *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II*, volume 9912, page 99126N. International Society for Optics and Photonics, 2016.

[15] Christopher C Stark, Eric J Cady, Mark Clampin, Shawn Domagal-Goldman, Doug Lisman, Avi M Mandell, Michael W McElwain, Aki Roberge, Tyler D Robinson, Dmitry Savransky, et al. A direct comparison of exoearth yields for starshades and coronagraphs. In *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave*, volume 9904, page 99041U. International Society for Optics and Photonics, 2016.

[16] Kunjithapatham Balasubramanian, AJ Eldorado Riggs, Eric Cady, Victor White, Karl Yee, Daniel Wilson, Pierre Echternach, Richard Muller, Camilo Mejia Prada, Byoung-Joon Seo, et al. Fabrication of coronagraph masks and laboratory scale star-shade masks: characteristics, defects, and performance. In *Techniques and Instrumentation for Detection of Exoplanets VIII*, volume 10400, page 104000C. International Society for Optics and Photonics, 2017.

[17] Siddharth Hegde, Ivan G Paulino-Lima, Ryan Kent, Lisa Kaltenegger, and Lynn Rothschild. Surface biosignatures of exo-earths: remote detection of extraterrestrial life. *Proceedings of the National Academy of Sciences*, 112(13):3886–3891, 2015.

[18] James B Breckinridge, Wai Sze T Lam, and Russell A Chipman. Polarization aberrations in astronomical telescopes: the point spread function. *Publications of the Astronomical Society of the Pacific*, 127(951):445, 2015.

[19] Russell A Chipman and Wai Sze Tiffany Lam. The polaris-m ray tracing program. In *Polarization Science and Remote Sensing VII*, volume 9613, page 96130J. International Society for Optics and Photonics, 2015.

[20] Garam Yun. *Polarization ray tracing.* PhD thesis, The University of Arizona, 2011.

[21] Airy Optics. `http://www.airyoptics.com/`. Accessed: 2019-02-22.

[22] AJ Eldorado Riggs, Garreth Ruane, Erkin Sidick, Carl Coker, Brian D Kern, and Stuart B Shaklan. Fast linearized coronagraph optimizer (falco) i: a software toolbox for rapid coronagraphic design and wavefront

correction. In *Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave*, volume 10698, page 106982V. International Society for Optics and Photonics, 2018.

[23] Fast linearized coronagraph optimizer (falco). `https://github.com/ajeldorado/falco-matlab`. Accessed: 2019-04-25.

[24] Eugene Hecht. *Optics*. Addison-Wesley, 4th edition, 2002.

[25] Masud Mansuripur. *Field, Force, Energy and Momentum in Classical Electrodynamics*. Bentham Science Publishers Ltd., 2011.

[26] R Clark Jones. A new calculus for the treatment of optical systems i. description and discussion of the calculus. *Josa*, 31(7):488–493, 1941.

[27] Henry Hurwitz and R Clark Jones. A new calculus for the treatment of optical systems ii. proof of three general equivalence theorems. *JOSA*, 31(7):493–499, 1941.

[28] R Clark Jones. A new calculus for the treatment of optical systems iii. the sohncke theory of optical activity. *Josa*, 31(7):500–503, 1941.

[29] R Clark Jones. A new calculus for the treatment of optical systems iv. *Josa*, 32(8):486–493, 1942.

[30] R Clark Jones. A new calculus for the treatment of optical systems v. a more general formulation, and description of another calculus. *JOSA*, 37(2):107–110, 1947.

[31] R Clark Jones. A new calculus for the treatment of optical systems vi. experimental determination of the matrix. *JOSA*, 37(2):110–112, 1947.

[32] R Clark Jones. A new calculus for the treatment of optical systems vii. properties of the n-matrices. *Josa*, 38(8):671–685, 1948.

[33] R Clark Jones. New calculus for the treatment of optical systems viii. electromagnetic theory. *Josa*, 46(2):126–131, 1956.

[34] Garam Yun. *Polarization Ray Tracing*. PhD thesis, The University of Arizona, 2011.

[35] Shih-Yau Lu and Russell A Chipman. Homogeneous and inhomogeneous jones matrices. *JOSA A*, 11(2):766–773, 1994.

[36] Nicholas J Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986.

[37] Stephen H. Friedberg, Arnold J. Insel, and Lawrence E. Spence. *Linear Algebra*. Prentice Hall, 1997.

[38] Russell A Chipman, Wai Sze Tiffany Lam, and Garam Young. *Polarized Light and Optical Systems*. CRC Press, 2018.

[39] Garam Yun, Karlton Crabtree, and Russell A Chipman. Three-dimensional polarization ray-tracing calculus i: definition and diattenuation. *Applied optics*, 50(18):2855–2865, 2011.

[40] Johannes Ruoff and Michael Totzeck. Orientation zernike polynomials: a useful way to describe the polarization effects of optical imaging systems. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 8(3):031404, 2009.

[41] Russell A Chipman. Polarization ray tracing. In *Recent Trends in Optical Systems Design and Computer Lens Design Workshop*, volume 766, pages 61–69. International Society for Optics and Photonics, 1987.

[42] Brian Daugherty. *Form Birefringence in Optical Systems*. PhD thesis, The University of Arizona, 2019.

[43] M.W. McCall, I.J. Hodgkinson, and Q. Wu. *Birefringent Thin Films And Polarizing Elements (2nd Edition)*. World Scientific Publishing Company, 2014.

[44] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[45] James P McGuire and Russell A Chipman. Diffraction image formation in optical systems with polarization aberrations. i: Formulation and example. *JOSA A*, 7(9):1614–1626, 1990.

[46] Garam Yun, Stephen C McClain, and Russell A Chipman. Three-dimensional polarization ray-tracing calculus ii: retardance. *Applied optics*, 50(18):2866–2874, 2011.

[47] Rudolf Kingslake and R Barry Johnson. *Lens design fundamentals*. academic press, 2009.

[48] David G Voelz. *Computational Fourier Optics: A MATLAB Tutorial (SPIE Tutorial Texts Vol. TT89)*. SPIE press, 2011.

[49] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[50] Robert Tyson. *Principles of adaptive optics*. CRC press, 2010.

[51] D Mawet, E Serabyn, K Liewer, Ch Hanot, S McEldowney, D Shemo, and N O'Brien. Optical vectorial vortex coronagraphs using liquid crystal polymers: theory, manufacturing and laboratory demonstration. *Optics Express*, 17(3):1902–1918, 2009.

[52] Anand Sivaramakrishnan, Christopher D Koresko, Russell B Makidon, Thomas Berkefeld, and Marc J Kuchner. Ground-based coronagraphy with high-order adaptive optics. *The astrophysical journal*, 552(1):397, 2001.

[53] Dimitri Mawet, Laurent Pueyo, Dwight Moody, John Krist, and Eugene Serabyn. The vector vortex coronagraph: sensitivity to central obscuration, low-order aberrations, chromaticism, and polarization. In *Modern Technologies in Space-and Ground-based Telescopes and Instrumentation*, volume 7739, page 773914. International Society for Optics and Photonics, 2010.

[54] Garreth Ruane, Dimitri Mawet, Bertrand Mennesson, Jeffrey Jewell, and Stuart Shaklan. Vortex coronagraphs for the habitable exoplanet imaging mission concept: theoretical performance and telescope requirements. *Journal of Astronomical Telescopes, Instruments, and Systems*, 4(1):015004, 2018.

[55] Scott C McEldowney, David M Shemo, and Russell A Chipman. Vortex retarders produced from photo-aligned liquid crystal polymers. *Optics express*, 16(10):7295–7308, 2008.

[56] John Krist, Ruslan Belikov, Dimitri Mawet, Dwight Moody, Laurent Pueyo, Stuart Shaklan, and John Trauger. Assessing the performance limits of internal coronagraphs through end-to-end modeling. Technical report, National Aeronautics and Space Administration, 2013.

[57] Kelsey L Miller. *Development and Demonstration of new Focal Plane Wavefront Sensing Techniques for High-Contrast Direct Imaging of Exoplanets*. PhD thesis, The University of Arizona, 2018.

[58] Jeffrey Davis, Meredith K Kupinski, Russell A Chipman, and James B Breckinridge. Habex polarization ray trace and aberration analysis. In *Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave*, volume 10698, page 106983H. International Society for Optics and Photonics, 2018.

[59] Remi Soummer, Laurent Pueyo, Anand Sivaramakrishnan, and Robert J Vanderbei. Fast computation of Lyot-style coronagraph propagation. *Optics Express*, 15(24):15935–15951, 2007.

[60] Mary L Boas. *Mathematical methods in the physical sciences*. John Wiley & Sons, 2006.

[61] Habitable Exoplanet Observatory (HabEx). `https://www.jpl.nasa.gov/habex/`. Accessed: 2019-03-18.

[62] Large UV Optical IR Surveyor (LUVOIR). `https://asd.gsfc.nasa.gov/luvoir/`. Accessed: 2019-03-18.

[63] Bertrand Mennesson, Scott Gaudi, Sara Seager, Kerri Cahoy, Shawn Domagal-Goldman, Lee Feinberg, Olivier Guyon, Jeremy Kasdin, Christian Marois, Dimitri Mawet, et al. The habitable exoplanet (habex) imaging mission: preliminary science drivers and technical requirements. In *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave*, volume 9904, page 99040L. International Society for Optics and Photonics, 2016.

[64] Paul A Scowen, Daniel Stern, Rachel Somerville, Mayer Rud, Stefan Martin, and Matthew Beasley. Science and architecture drivers for the habex ultraviolet spectrograph (uvs). In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 1039807. International Society for Optics and Photonics, 2017.

[65] Manan Arya, David Webb, James McGown, P Douglas Lisman, Stuart Shaklan, S Case Bradford, John Steeves, Evan Hilgemann, Brian Trease, Mark Thomson, et al. Starshade mechanical design for the habitable exoplanet imaging mission concept (habex). In *Techniques and Instrumentation for Detection of Exoplanets VIII*, volume 10400, page 104001C. International Society for Optics and Photonics, 2017.

[66] Stefan Martin, Mayer Rud, Dimitri Mawet, Joel Nissen, Stuart Shaklan, and Luis Marchen. Habex space telescope exoplanet instruments. In *Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave*. Society of Photo-optical Instrumentation Engineers (SPIE), 2018.

[67] Stefan Martin, Mayer Rud, Paul Scowen, Daniel Stern, Joel Nissen, and John Krist. Habex space telescope optical system. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 1039805. International Society for Optics and Photonics, 2017.

[68] Fredric J Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[69] Olivier Guyon. High performance coronagraphy for direct imaging of exoplanets. In *EPJ Web of Conferences*, volume 16, page 03001. EDP Sciences, 2011.

[70] John Krist, AJ Riggs, James McGuire, Hong Tang, Nikta Amiri, Gary Gutt, Luis Marchen, David Marx, Bijan Nemati, Navtej Saini, et al. Wfirst coronagraph optical modeling. In *Techniques and Instrumentation for Detection of Exoplanets VIII*, volume 10400, page 1040004. International Society for Optics and Photonics, 2017.

[71] Amir Give'on, Brian Kern, Stuart Shaklan, Dwight C Moody, and Laurent Pueyo. Broadband wavefront correction algorithm for high-contrast imaging systems. In *Astronomical Adaptive Optics Systems and Applications III*, volume 6691, page 66910A. International Society for Optics and Photonics, 2007.

[72] OpticStudio. `https://www.zemax.com/products/opticstudio`. Accessed: 2019-02-06.

[73] Matthew R Bolcar, Steve Aloezos, Vincent T Bly, Christine Collins, Julie Crooke, Courtney D Dressing, Lou Fantano, Lee D Feinberg, Kevin

France, Gene Gochar, et al. The large uv/optical/infrared surveyor (luvoir): Decadal mission concept design update. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 1039809. International Society for Optics and Photonics, 2017.

[74] Kevin France, Brian Fleming, Garrett West, Stephan R McCandliss, Matthew R Bolcar, Walter Harris, Leonidas Moustakas, John M O'Meara, Ilaria Pascucci, Jane Rigby, et al. The luvoir ultraviolet multi-object spectrograph (lumos): instrument definition and design. In *UV, X-ray, and gamma-ray space instrumentation for astronomy XX*, volume 10397, page 1039713. International Society for Optics and Photonics, 2017.

[75] Kevin C France, Brian T Fleming, and Keri Hoadley. Chisl: the combined high-resolution and imaging spectrograph for the luvoir surveyor. *Journal of Astronomical Telescopes, Instruments, and Systems*, 2(4):041203, 2016.

[76] Laurent Pueyo, Neil Zimmerman, Matthew Bolcar, Tyler Groff, Christopher Stark, Garreth Ruane, Jeffrey Jewell, R Soummer, K St Laurent, J Wang, et al. The luvoir architecture" a" coronagraph instrument. In *UV/Optical/IR Space Telescopes and Instruments: Innovative Technologies and Concepts VIII*, volume 10398, page 103980F. International Society for Optics and Photonics, 2017.

[77] Matthew R Bolcar, Julie Crooke, Jason E Hylan, Ginger Bronke, Christine Collins, James Corsetti, Joe Generie, Qian Gong, Tyler Groff, William Hayden, et al. The large uv/optical/infrared surveyor (luvoir): decadal mission study update. In *Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave*, volume 10698, page 106980O. International Society for Optics and Photonics, 2018.

[78] Jason E Hylan, Matthew R Bolcar, Julie Crooke, Ginger Bronke, Christine Collins, James Corsetti, Joe Generie, Qian Gong, Tyler Groff, William Hayden, et al. The large uv/optical/infrared surveyor (luvoir): Decadal mission concept study update. 2019.

[79] Natalie Clark and James B Breckinridge. Polarization compensation of fresnel aberrations in telescopes. In *Uv/Optical/Ir Space Telescopes and Instruments: Innovative Technologies and Concepts V*, volume 8146, page 81460O. International Society for Optics and Photonics, 2011.

[80] Peter William Maymon and Russell A Chipman. Linear polarization sensitivity specifications for spaceborne instruments. In *Polarization Analysis and Measurement*, volume 1746, pages 148–157. International Society for Optics and Photonics, 1992.

[81] Wai Sze Tiffany Lam and Russell Chipman. Balancing polarization aberrations in crossed fold mirrors. *Applied optics*, 54(11):3236–3245, 2015.

[82] Russell A Chipman, Wai Sze Tiffany Lam, and Garam Young. *Polarized Light and Optical Systems*, chapter 14. CRC Press, 2018.

[83] Wolfram mathematica. `https://www.wolfram.com/mathematica/`. Accessed: 2019-04-17.

[84] Numerical nonlinear global optimization. `https://reference.wolfram.com/language/tutorial/ConstrainedOptimizationGlobalNumerical.html`. Accessed: 2019-04-17.

[85] José Sasián. *Introduction to aberrations in optical imaging systems*. Cambridge University Press, 2013.

[86] Daniel Malacara. *Optical shop testing*. John Wiley & Sons, 2007.

[87] W.J. Smith. *Modern Optical Engineering, 4th Ed.* McGraw Hill professional. McGraw-Hill Education, 2007.

[88] Veronica M Reading and RA Weale. Macular pigment and chromatic aberration. *JOSA*, 64(2):231–234, 1974.

[89] John E Greivenkamp. *Field guide to geometrical optics*, volume 1. SPIE Press Bellingham, WA, 2004.

[90] Daniel J Reiley and Russell A Chipman. Coating-induced wavefront aberrations: on-axis astigmatism and chromatic aberration in all-reflecting systems. *Applied optics*, 33(10):2002–2012, 1994.

[91] Robert R Shannon. *The art and science of optical design*. Cambridge University Press, 1997.

[92] Russell A Chipman. Polarization analysis of optical systems. *Optical engineering*, 28(2):280290, 1989.

[93] Russel A Chipman. *Polarization Aberrations*. PhD thesis, University of Arizona, 1987.

[94] Russell A Chipman. Polarization analysis of optical systems, ii. In *Polarization Considerations for Optical Systems II*, volume 1166, pages 79–95. International Society for Optics and Photonics, 1990.

[95] James P McGuire and Russell A Chipman. Polarization aberrations. 1. rotationally symmetric optical systems. *Applied optics*, 33(22):5080–5100, 1994.

[96] James P McGuire and Russell A Chipman. Polarization aberrations. 2. tilted and decentered optical systems. *Applied optics*, 33(22):5101–5107, 1994.

[97] Russell Atwood Chipman. *Polarization aberrations*. PhD thesis, The University of Arizona, 1987.

[98] Virendra N Mahajan. *Optical Imaging and Aberrations: Ray Geometrical Optics*, volume 45. SPIE press, 1998.

[99] Helmut KV Lotsch. Reflection and refraction of a beam of light at a plane interface. *JOSA*, 58(4):551–561, 1968.

[100] H Angus Macleod. *Thin-film optical filters*. CRC press, 2010.

[101] Minghong Yang, Alexandre Gatto, and Norbert Kaiser. Optical thin films with high reflectance, low thickness and low stress for the spectral range from vacuum uv to near ir. *Journal of Optics A: Pure and Applied Optics*, 8(3):327, 2006.

[102] Russell A Chipman, Wai Sze Tiffany Lam, and Garam Young. *Polarized Light and Optical Systems*, chapter 13. CRC Press, 2018.

[103] James C Wyant and Katherine Creath. Basic wavefront aberration theory for optical metrology. *Applied optics and optical engineering*, 11(s 29):2, 1992.

[104] John E Krist. Proper: an optical propagation library for idl. In *Optical Modeling and Performance Predictions III*, volume 6675, page 66750P. International Society for Optics and Photonics, 2007.

[105] Wai Sze Tiffany Lam. *Anisotropic ray trace*. PhD thesis, The University of Arizona, 2015.

[106] Murray Eisenberg and Robert Guy. A proof of the hairy ball theorem. *The American Mathematical Monthly*, 86(7):571–574, 1979.