JATIS Journal of Astronomical Telescopes, Instruments, and Systems

RESEARCH PAPER

Hybrid propagation physics for the design and modeling of astronomical observatories: a coronagraphic example

Jaren N. Ashcraft,[a,*] Ewan S. Douglas,[b,*] Daewook Kim,[a,b,c] and A. J. Eldorado Riggs[d]

[a]University of Arizona, James C. Wyant College of Optical Sciences, Tucson, Arizona, United States
[b]University of Arizona, Department of Astronomy and Steward Observatory, Tucson, Arizona, United States
[c]University of Arizona, Large Binocular Telescope Observatory, Tucson, Arizona, United States
[d]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, United States

ABSTRACT. For diffraction-limited optical systems, an accurate physical optics model is necessary to properly evaluate instrument performance. Astronomical observatories outfitted with coronagraphs for direct exoplanet imaging require physical optics models to simulate the effects of misalignment and diffraction. Accurate knowledge of the observatory's point-spread function (PSF) is integral for the design of high-contrast imaging instruments and simulation of astrophysical observations. The state of the art is to model the misalignment, ray aberration, and diffraction across multiple software packages, which complicates the design process. Gaussian beamlet decomposition (GBD) is a ray-based method of diffraction calculation that has been widely implemented in commercial optical design software. By performing the coherent calculation with data from the ray model of the observatory, the ray aberration errors can be fed directly into the physical optics model of the coronagraph, enabling a more integrated model of the observatory. We develop a formal algorithm for the transfer-matrix method of GBD and evaluate it against analytical results and a traditional physical optics model to assess the suitability of GBD for high-contrast imaging simulations. Our GBD simulations of the observatory PSF, when compared to the analytical Airy function, have a sum-normalized RMS difference of $\approx 10^{-6}$. These fields are then propagated through a Fraunhofer model of an exoplanet imaging coronagraph where the mean residual numerical contrast is 4×10^{-11}, with a maximum near the inner working angle at 5×10^{-9}. These results show considerable promise for the future development of GBD as a viable propagation technique in high-contrast imaging. We developed this algorithm in an open-source software package and outlined a path for its continued development to increase the accuracy and flexibility of diffraction simulations using GBD.

Keywords: physical optics modeling; coronagraphs; exoplanets; high-contrast imaging; diffraction; Gaussian beamlet decomposition

Paper 23009G received Jan. 18, 2023; revised Oct. 2, 2023; accepted Oct. 23, 2023; published Nov. 16, 2023.

*Address all correspondence to Jaren N. Ashcraft, jashcraft@arizona.edu; Ewan S. Douglas, douglase@arizona.edu

1 Introduction

1.1 Astrophysical Motivation

Integrated models of optical observatories are highly beneficial to their design and use.[1,2] Accurate observatory models permit powerful insight into predicting the as-built performance of a given instrument. However, the accuracy of these models is fundamentally limited by the assumptions made. To facilitate high-yield scientific observations, astronomical observatories are nominally designed to operate in the diffraction limit where wavefront aberrations are small. Diffraction-limited optical observatories are necessarily modeled with diffraction integrals derived from the Huygens–Fresnel principle to support the wave-like behavior of light. The paraxial and scalar assumptions that angles of incidence are small and that polarization is negligible[3] are made to ease the computational burden on the model. The resultant Fresnel and Fraunhofer diffraction integrals are accurate, providing these conditions are met. If the performance of the observatory is limited by a factor outside the assumptions made, and then the model will be ignorant of it. An example of this is the linear and shift-invariant assumption imposed on diffraction models of astronomical observatories. Ray aberrations (e.g., coma and astigmatism) have a field dependence and consequently change across an observatory's field of view. However, diffraction integrals assume shift invariance. This means that the aberrations do not change across the field of view and a separate ray trace model must be used to capture this effect.

Integrating optical models from different regimes in physics has become a popular method by which to overcome this limitation. Linking ray trace models to diffraction models in particular can overcome the paraxial and scalar assumption imposed by the Fresnel and Fraunhofer diffraction integrals. In the prior example, to capture the influence of optical aberrations, a new ray trace must be performed and the optical path difference of the rays must be translated to a diffraction model for each point of interest in the field of view. Similarly, diffraction integrals are incapable of determining the effects of optical polarization. For example, the Daniel K. Inoyue Solar Telescope supports a suite of polarimetric instrumentation that is sensitive to the influence of optical polarization. To support this regime of optical physics, the scalar assumption is not sufficient, so the polarization state is propagated along geometric ray paths using polarization ray tracing[4,5] to determine the influence of polarization aberrations on the optical beam.[6] Modern space telescopes also require integrated models to accurately predict the instrument behavior. For example, the optical models of the James Webb Space Telescope incorporate the influence of dynamic thermal, structural, and optical effects simultaneously to produce an accurate library of the observatory's jitter.[7] High-contrast imaging instruments that use coronagraphs, designed to separate exoplanets from diffracted starlight, are in dire need of integrated physical optics modeling from their inception. These coronagraphs aim to discern targets that are orders of magnitude dimmer than their host star.[8–10] Understanding all sources of error is of paramount importance to the functionality of the instrument.

High-contrast imaging instruments have been successfully deployed on the ground (e.g., SCExAO,[11] MagAO-X,[12] NIRC2,[13] GPI,[14] and SPHERE[15]) and in space (e.g., NICMOS[16] and NIRCam[17,18]) to pursue the direct detection of extrasolar planets, debris, and protoplanetary disks. The Decadal Survey on Astronomy and Astrophysics 2020 (Astro2020) recommends pursuing these instruments for a future 6 meter diameter infrared/optical/visible (IROUV) flagship observatory for the progression of astrophysical sciences.[19]

Presently, the optical design of observatories is done in a ray-tracing engine[20,21] (e.g., CODE V and Zemax OpticStudio) because it is more suitable to optimizing the shapes of observatory mirrors. Upon reaching a diffraction-limited optical design, the system is then assumed to be well-represented by a paraxial diffraction model. The wavefront maps produced by the ray trace model of the observatory and the contributions from the imperfect polishing of the observatory mirrors are sent to a linearized physical optics propagator to examine the image plane electric field in the presence of diffraction from structure in the beam and phase errors on the optics.

Many tools have been developed to simulate the performance of high-contrast imaging instrumentation. Tiny Tim is one of the first of these widely used packages used to simulate the Hubble Space Telescope (HST) instrument point-spread functions (PSFs).[22] The tool generates aberrated PSFs based on the instrument, observation, and dynamic aberrations for a given observing scenario, enabling highly accurate simulations of the observatory performance.

However, it only considers aberrations that are conjugate to the exit pupil of the observatory. This limits the model's ability to capture out-of-pupil effects, such as the Talbot effect and speckles from optical surfaces.[3] To capture these effects, optical propagation packages integrate optical models of observatories by adding Fresnel diffraction to the PSF simulation, enabling the modeling of plane-to-plane diffraction effects. Open-source packages that currently support Fresnel diffraction include: PROPER,[23] Physical Optics Propagation in PYthon (POPPY),[24–26] High-Contrast Imaging in Python (HCIPy),[27] AOTools,[28] and prysm.[2,29] Using these tools, near-field diffraction that limits high-contrast imaging can be modeled, and focal-plane wavefront sensing methods can be tested.

These open-source physical optics propagation tools form the cornerstone of high-contrast imaging instrument modeling and design. The open-source framework means that the codes are accessible to anyone, so the physics are completely verifiable by the scientific community.[30] It is in the scientific community's best interest to continue to develop open-source propagation physics modules to increase the scope of and further integrate our observatory models.

Commercial optical design codes offer the ability to make diffraction calculations based on ray data, but their physical optics simulation techniques are not as transparent or versatile as the open-source propagation codes that are used to design coronagraphs for astronomical observatories. The current open-source physical optics codes used for observatory modeling are also limited in their scope because of the Fresnel approximation, which is incapable of accurately modeling the field after fast-focusing and highly aspheric surfaces.[31,32] As observatories get larger, their optics may become faster (i.e., lower $F\#$) and more aspheric to fit within an available volume. Some coronagraph architectures capable of Earth-like exoplanet detection (e.g., phase-induced amplitude apodization or PIAA[33]) employ mirrors that apodize the pupil with highly aspheric mirrors and require tailored propagators in order to be included in physical optics models.[31] In the regime where the contribution of these surfaces is best represented by a ray trace, a diffraction calculation must be made to appropriately model the optical field at the image plane. To continue the development of integrated optical models, exploring the possibilities and limitations of new propagation techniques is desirable.

An example of the typical integrated modeling pipeline for astronomical observatories outfitted with coronagraphs is shown in Fig. 1. The observatory is typically designed and modeled in ray trace software (e.g., CODE V and Zemax OpticStudio) to accurately model the wavefront in the observatory's exit pupil. Upon finalizing the design, the complex-valued exit pupil is decomposed into a functional representation (e.g., a set of polynomial coefficients, such as the Zernikes[35]) and passed to the entrance pupil of a coronagraph model constructed in an open-source, Fourier-based physical optics propagator (e.g., POPPY, PROPER, and HCIPy). The front-end model computes the complex field distribution at the coronagraph mask and then propagates the field past the mask to the image plane. The field is taken by a model of the detector (e.g., EMCCD Detect[36] and Pyxel[37]) to create a simulated raw science image that can be post-processed (e.g., PyKLIP[38] and NMF imaging[39]).

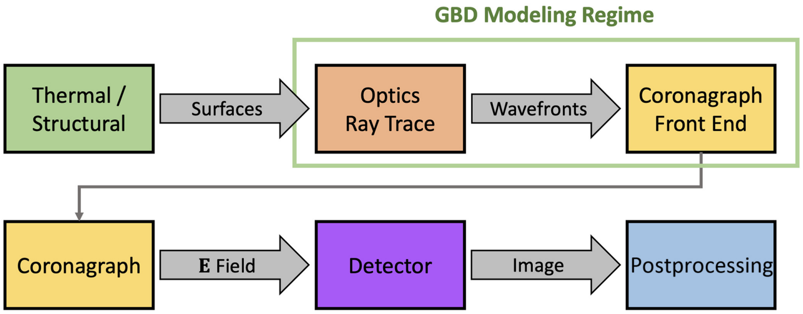

Fig. 1 Modeling flow inspired by the structural thermal optical performance (STOP) modeling process for the Roman Coronagraph.[34] This diagram illustrates the different modeling regimes required to create a simulated image of an observatory. We aim to further integrate this modeling pipeline by creating an open-source GBD platform to unify the ray trace model of the observatory with the diffraction model of the coronagraph.

To bridge the gap between commercial ray tracing engines and open-source physical optics propagation codes, we investigate the viability of a ray-based diffraction calculation called Gaussian beamlet decomposition (GBD) for modeling observatories with coronagraphs. Traditionally, GBD operates using the complex ray tracing algorithm described in the works by Greynolds[40] and Harvey et al.[41] This technique has been previously implemented in FRED,[41,42] and possibly in CODE V,[43] but an exact method of its implementation in these software packages is not clearly available in the literature. An alternative approach called the transfer matrix algorithm was recently developed to improve GBD's viability for precision diffraction simulation by Worku and Gross.[44–46] However, their implementation is not public and has not yet been formally evaluated as a tool to augment the modeling of astronomical observatories or high-contrast imaging instrumentation. To formally evaluate GBD as a modeling tool for astronomical instrumentation, a complete algorithm for its implementation is derived in this article.

1.2 Gaussian Beamlet Decomposition

GBD is a method of physical optics propagation that approximates the propagated field as a finite sum of coherent Gaussian beams that each propagate along a ray path. This method has been implemented in optical design packages[47] to perform coherent calculations on non-paraxial systems. The operating principle of GBD is to decompose the field in the entrance pupil of an optical system [Fig. 2(a)] into a finite set of Gaussian beams [Fig. 2(b)]. Their coherent sum [Figs. 2(c) and 2(d)] approximates the initial field decomposition and can be propagated anywhere in the optical system along geometric ray paths.

Undersampling the field in the entrance pupil leads to artifacts in the decomposition. A characteristic amplitude ripple based on the period of the beamlet decomposition remains in the field. Due to the soft edges of the Gaussian beams, they cannot completely reconstruct the field of a sharp aperture edge. Using a larger number of smaller beamlets decreases the beamlet distribution period and increases the slope of the Gaussian beams, allowing for the mitigation of both of these effects [shown in Fig. 2(d)].

Upon decomposing the initial wavefront into a sufficient set of Gaussian beams, we can use their analytical linear propagation laws[48] (described in Sec. 2) to compute the coherent field at any arbitrary plane in the optical system. Fourier transform-based propagation methods derived from the Huygens–Fresnel principle typically assume that the field is scalar and the optical system is paraxial.[3] This is usually appropriate for stellar coronagraphs, which operate on slowly focusing beams with diffraction-limited optics, but may not be for next-generation observatories whose large apertures may necessitate relatively fast telescope optics.

GBD computes the same complex optical field without making the paraxial assumption across the observatory. Rather, the coherent field of Gaussian beams is derived from the ray data directly. Doing so imposes the paraxial assumption about a single beamlet instead of the

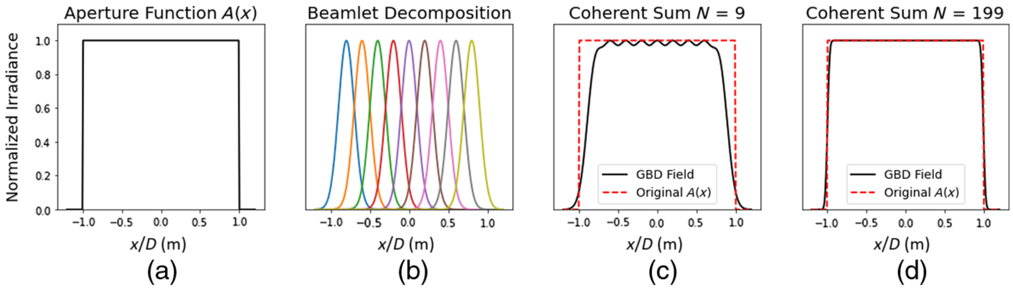

Fig. 2 Illustration of the operating principle of GBD in one dimension. (a) The aperture function for traditional imaging systems as a top-hat function; (b) the decomposition of this function into a discrete set of Gaussian beams; (c) nine evenly spaced Gaussian profiles before their coherent summation. The coherent sum (black) of these nine beamlets shows that the beamlets are incapable of perfectly reconstructing the original aperture function (dashed red), specifically the sharp edge and uniform amplitude. (d) Using more beamlets results in a more accurate reconstruction of the aperture function. The amplitude ripple has virtually vanished, and the field near the aperture edges is almost entirely recovered.

entire observatory, which is a much less stringent approximation. Gaussian beams are technically infinite in extent, but extremely localized around the beam waist. Consequently, the contribution of the field very far from the Gaussian is negligible. This locality enables the simulation of the optical system to generally be non-paraxial. By making the diffraction calculation directly from ray data, GBD circumvents the need for translating the wavefront to a physical optics propagator and imposing the paraxial assumption on the optical system. Instead, the ray trace model can be directly integrated into the diffraction model.

There are two main approaches that exist in the literature to implement GBD: the complex ray tracing method and the transfer matrix method. The complex ray tracing method was recently described by Harvey et al.[41] in their seminal paper about implementing GBD in photon engineering's non-sequential ray tracing software FRED. This method traces waist and divergence rays to compute the complex field at the plane of interest using Arnaud's method of complex ray tracing.[40,49] Through FRED, the complex ray tracing method has seen widespread use for non-paraxial coherent beam analysis.

Another GBD approach, the transfer matrix method was developed by Worku and Gross[44–46] to mitigate GBD's inability to simulate sharp-edge diffraction and add new utility to the technique. This formulation of GBD works by computing the differential ray transfer matrix for a given ray path and then using that data to solve the Gaussian beam solution to the general Collins integral.[50] Worku and Gross have leveraged the general Collins integral to provide alternative conditions to the Gaussian beam solution to modify the decomposition, such as truncated[46] and pulsed[51] beamlet decomposition. The option of modifying the beamlets to overcome the limitations of GBD makes the transfer matrix method extremely attractive for use in high-contrast imaging where preservation of high-spatial frequency content is important. Of particular interest are mirror segment gaps and optomechanical structures that obscure the primary mirror. Therefore, we elect to investigate Worku and Gross's transfer matrix method of GBD for the work presented in this article. Our goal is to publicize the transfer matrix method by developing an algorithm for its implementation and then use it to characterize GBD's suitability for high-contrast imaging simulation. Our work can then be used as a platform with which to study the suitability of Worku and Gross' modified GBD in future investigations.

1.3 Hybrid Propagation Physics

The ray-based nature of GBD introduces problems in modeling the electric field when the rays are vignetted. Structure in the field where the initial decomposition occurs (typically, the entrance pupil) can be well-represented by Gaussian beams as long as the structure of interest is larger than the beamlets used in decomposition. Diffraction from structure in interemediate planes (between the pupil and focal planes) is challenging to represent if the beamlets diverge considerably. However, secondary beamlets can be traced from these intermediate structures to aid in the accuracy of the simulation.[52] At the focal plane of a diffraction-limited system, the rays are highly concentrated while the diffracted field spreads out considerably (e.g., the Airy disk). Because of GBD's reliance on ray tracing, it cannot represent diffraction from structure in the focal plane well without redecomposing the field.[42] For the case of a Lyot-type coronagraph, all rays are vignetted at the focal plane mask (FPM) and the field decomposition is lost. To circumvent this, we compute the field before the FPM with GBD and propagate it through the remaining coronagraph with traditional diffraction integrals, where we expect the low-order aberrations to be small and the paraxial assumption to be valid. This hybrid method (shown in Fig. 3) enables the user to alter the propagation physics for the electric field based on where it is the most appropriate; GBD will simulate the fast beams in the fore-optics and paraxial diffraction will propagate the field through the coronagraph imaging optics. In practice, GBD would be used to propagate to the entrance pupil of the coronagraph for simulating systems with wavefront control, where it would then hand off the field to a paraxial diffraction model.

The end result of such a model allows for direct integration of the ray trace model with the physical optics model, without imposing the paraxial approximation on the observatory. Like Fresnel and Fraunhofer diffraction, GBD is an approximation to diffraction physics. The decomposition of the field into Gaussian beams does not have an analytical solution. Therefore, undesirable artifacts can be introduced into the field if the decomposition is not well-understood and the sampling is insufficient.[53] To better understand the impact of a GBD PSF on high-contrast

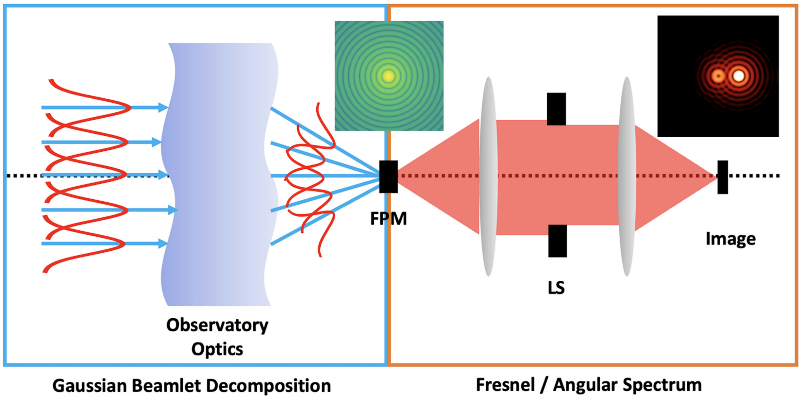

Fig. 3 Diagram demonstrating the hybrid propagation physics model. The observatory optics are best described by a ray-based propagation model, so GBD is used to compute the field at the image plane before the coronagraph FPM. The field array is then passed to a paraxial diffraction model, which propagates it past the FPM and through the remainder of the coronagraph. This propagation scheme permits the user to model the influence of the observatory optics non-paraxially, without losing accuracy after propagating past the FPM.

imaging simulations, we develop a hybrid propagation model to compare GBD to an equivalent Fraunhofer diffraction model.

To our knowledge, the transfer matrix method of GBD has not seen widespread implementation. Given its obvious benefits, we believe that this is because the transfer matrix method is not well-understood by the scientific community. We aim to remedy this by presenting a vectorizeable algorithm for the transfer matrix method and open-sourcing our simulation platform for future investigators to use. This article is the first work, to our knowledge, to provide the explicit mathematics of GBD and provide our code as an object-oriented module for more widespread use.

Our GBD module was built in the Poke[55,56] Python package currently available on Github. Poke was originally developed to be a polarization ray tracing module to study the influence of polarization aberrations on astronomical coronagraphs.[4] For this study, we expanded its capabilities to include GBD. Poke operates using ray tracer API's to trace a raybundle through every surface in the optical system. The relevant ray data are stored in a Rayfront object and can be loaded into a Python environment and interacted with independent of the ray tracer that generated the data (Fig. 4). The Rayfronts can also be compiled into binary file types using the msgpack[57] package and distributed to any interested investigator, effectively open-sourcing the physical optics calculations done on ray data.

In this study, we conduct ray traces in Zemax, which are then saved as a Poke Rayfront object. Poke performs the GBD simulations using the saved ray data to generate the field at the focal plane of the telescope. These data are exported to a coronagraph model built using HCIPy.

In Sec. 2, we outline the mathematics of Gaussian beam propagation and differential ray tracing used to perform GBD simulations. In Sec. 3, we present the mathematical algorithm for

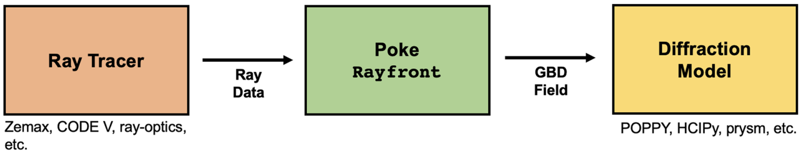

Fig. 4 Illustration of Poke's use as an interface to open-source ray-based physical optics. Poke only requires an interface between a ray tracing engine (orange, left) to generate and save a Rayfront object, which includes all ray data necessary for the GBD calculation. Currently, Poke supports sequential systems in CODE V and Zemax, but we are working on adding support for open-source packages that have ray tracers, like ray-optics.[54] The GBD field that Poke computes can then be sent to any open-source diffraction modeling package (yellow, right) to complete the hybrid propagation model.

the transfer-matrix method of GBD that we developed for this investigation. In Sec. 4, we compare the results of observatory PSFs produced by GBD with one produced using traditional diffraction methods and analyze the artifacts that remain in the field. In Sec. 5, we assess the suitability of GBD for high-contrast imaging models and establish a roadmap for our module's continued development.

2 Preliminary Mathematical Methods

In this section, we review the necessary mathematical tools that we need to formulate our GBD algorithm. This includes the propagation equations for a single Gaussian beam, how to use differential ray tracing to compute the parameters necessary for Gaussian beam propagation, and methods of decomposing the entrance pupil field in the optical system to improve the simulation's sensitivity to high-spatial frequencies.

2.1 Propagation of a Single Gaussian Beam

The equation for a single Gaussian beam is parameterized entirely by the complex beam parameter $Q(z)$:[3]

$$U(r,z) = \frac{U_o}{Q(z)} \exp\left[ik\frac{r^2}{2Q(z)}\right], \tag{1}$$

where $U(r,z)$ is the scalar Gaussian field, U_o is the amplitude, k is the wavenumber, and r is the radial coordinate in the plane perpendicular to propagation. The inverse of the complex parameter $Q(z)$ describes the beam's $1/e$ field radius [the "waist" $w(z)$] and wavefront radius of curvature $R(z)$:

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

$Q(z)^{-1}$ is a convenient expression of the Gaussian beam because it fully encapsulates the information required to describe the transverse electric field of the beam as it propagates. The real part of $Q(z)^{-1}$ is related to the radius of curvature of the wavefront:

$$R(z) = z\left(1 + \left(\frac{Z_o}{z}\right)^2\right), \tag{3}$$

where Z_o is the Rayleigh range, and z is the longitudinal propagation distance. The imaginary part of $Q(z)^{-1}$ is related to the beam waist radius:

$$w(z) = w_o\sqrt{1 + \left(\frac{z}{Z_o}\right)^2}. \tag{4}$$

In the paraxial regime, $Q(z)^{-1}$ can be propagated using the ABCD ray transfer matrices of geometrical optics:[48]

$$Q(z)_2^{-1} = \frac{C + DQ_1^{-1}}{A + BQ_1^{-1}}. \tag{5}$$

To account for system misalignments, $Q(z)^{-1}$ is a 2×2 matrix $\mathbf{Q}(z)^{-1}$ that encodes the complex curvature in two orthogonal directions and how they couple into each other, allowing for the beamlet to be generally astigmatic:[53,58]

$$\mathbf{Q}(z)^{-1} = \begin{pmatrix} Q(z)_{xx}^{-1} & Q(z)_{xy}^{-1} \\ Q(z)_{yx}^{-1} & Q(z)_{yy}^{-1} \end{pmatrix}. \tag{6}$$

This treatment allows for greater versatility in the beamlet propagation but requires that the elements of the ray transfer matrices are also 2×2 matrices. The propagation formula shown in Eq. (5) has a similar matrix extension:

$$\mathbf{Q}(z)_2^{-1} = (\mathbf{C} + \mathbf{D}\mathbf{Q}_1^{-1})(\mathbf{A} + \mathbf{B}\mathbf{Q}_1^{-1})^{-1}. \tag{7}$$

We can then express the propagated Gaussian beam with the following equation:

$$U(\mathbf{r}) = \frac{U_o}{\sqrt{\det|\mathbf{A} + \mathbf{B}\mathbf{Q}_1^{-1}|}} \exp\left[\frac{-ik}{2}\mathbf{r}^{\mathrm{T}}\mathbf{Q}_2^{-1}\mathbf{r}\right], \tag{8}$$

where $\mathbf{r}$ is the radial coordinate in the plane transverse to the propagation direction centered on the Gaussian beam. The formulation for the propagation of the complex curvature matrix allows for modeling Gaussian beams as they propagate along generally skew ray paths in non-axially symmetric optical systems. This is an important utility for diffraction modeling of wavefront aberrations introduced by system misalignment or thermal deformations, which generally break optical system symmetry. Note that the solution in Eq. (8) is only valid for propagation between planes that are orthogonal to the propagation direction of the Gaussian beam.[50] We next need to determine a method for computing the ray transfer matrix for an arbitrary ray path through an optical system in order to propogate the Gaussian beam.

2.2 Computing the Differential Ray Transfer Matrix

The ABCD ray transfer matrix is a useful and concise method for analyzing properties of ray paths along optical systems. In the regime of geometrical optics, a generally skew ray can be traced through a system using 4 × 4 ABCD ray transfer matrices.[59] These matrices model simple optical elements (e.g., thin lenses) with ease by operating on an input column vector that represents a light ray. The simplest ray transfer matrix that describes a paraxial and orthogonal optical system is a 2 × 2 operator that maps an input (i) spatial and angular coordinate to the appropriate output (o):

$$\begin{pmatrix} y_o \\ m_o \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} y_i \\ m_i \end{pmatrix}, \tag{9}$$

where y is the spatial coordinate transverse to the propagation direction, and m is the slope in that dimension. The elements of the ABCD matrix and ray vectors are real-valued scalars. To account for skew ray paths, the position and angle in the dimension orthogonal to y and the direction of propagation must be tracked, adding two dimensions to the matrix calculus. A 4 × 4 ABCD matrix describes a non-orthogonal system with tilts and decenters that map generally skew input rays to generally skew output rays:

$$\begin{pmatrix} x_o \\ y_o \\ l_o \\ m_o \end{pmatrix} = \begin{pmatrix} A_{xx} & A_{xy} & B_{xx} & B_{xy} \\ A_{yx} & A_{yy} & B_{yx} & B_{yy} \\ C_{xx} & C_{xy} & D_{xx} & D_{xy} \\ C_{yx} & C_{yy} & D_{yx} & D_{yy} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ l_i \\ m_i \end{pmatrix}. \tag{10}$$

For simplicity, it is convenient to represent the radial position in the plane transverse to propagation (x, y) and the corresponding direction in the dimension (l, m) as a position and angle vector, respectively $(\mathbf{r}, \boldsymbol{\theta})$. The ABCD matrix can similarly be condensed into 2 × 2 sub-matrices that operate on each spatial dimension, yielding a familiar notation:

$$\begin{pmatrix} \mathbf{r_o} \\ \boldsymbol{\theta}_o \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{r_i} \\ \boldsymbol{\theta}_i \end{pmatrix}. \tag{11}$$

This description is powerful because it communicates the elegance and simplicity of ray transfer matrices. All dimensions transverse to propagation are accounted for, but the calculus to propagate a ray is still the same. The ray transfer matrices for simple and paraxial optical elements (e.g., thin lens) are well known,[59] and were used in concert with GBD in a prior investigation with paraxial systems.[53] However, our aim is to use GBD to model non-paraxial optical system diffraction. Therefore, we need a method of computing the ray transfer matrix for an arbitrary skew ray path. A simple dimensional analysis of the matrix relation in Eq. (11) is a good place to start understanding how to construct an arbitrary ABCD matrix. Because the position element $\mathbf{r}$ of the ray vector must be in units of distance, and the angular element $\boldsymbol{\theta}$ must be dimensionless, the units of the ABCD matrix elements are constrained. $\mathbf{A}$ and $\mathbf{D}$ must be dimensionless and transform the ray position and angle through the optical system, indicating that they represent magnification. $\mathbf{B}$ and $\mathbf{C}$ must have units of distance and inverse distance, respectively. **B** operates on an angle and is therefore a metric of propagation distance through an optical system given some ray angle. **C** operates on a position and is therefore an indicator of the amount of refraction a ray experiences given a position in the entrance pupil.

Stone and Forbes'[60] work in differential ray tracing for inhomogeneous media was instructive in terms of deriving a method to construct the ABCD matrix. They illustrate the construction of individual optical elements through ray derivatives in a generally 4 × 4 matrix through derivatives of surface data. Using this method, the position and angular derivatives of an optical surface are taken and arranged in a matrix like in Eq. (11). The matrix product of the optical elements is then the final differential ray transfer (or ABCD) matrix. Their method is functional if the analytical expression of the optical elements are known but could be very computationally intensive if the optical system has many elements. Instead, we approximate the differential matrix by tracing additional rays and compute the finite difference of the ray coordinates and directions at the input and output of the optical system to approximate the derivative.

Our implementation of the differential ray tracing technique in Poke utilizes a user-specified ray tracer (e.g., Zemax OpticStudio) that propagates rays by computing Snell's law at each surface. For GBD, the ray coordinates of interest are on the source plane (where the field decomposition is done, e.g., the entrance pupil) and the transversal plane. The transversal plane is the plane normal to the central ray of a Gaussian beam that intersects the point in which we wish to evaluate the field. Propagation to this plane is critical, because the solution to Gaussian beam propagation [Eq. (8)] is only valid between planes orthogonal to the propagation direction defined by the central ray. An element of the ray transfer matrix can be computed by determining the ratio of the differential ray data on the transversal plane to the differential ray data on the source plane. An example of computing the element A_{yy} is given by the following equation:

$$A_{yy} = \frac{\partial y_T}{\partial y_S} = \frac{y_{+y,T} - y_{\mathrm{cen},T}}{y_{+y,S} - y_{\mathrm{cen},S}}, \tag{12}$$

where $y_{+y,T}$ and $y_{\mathrm{cen},T}$ are the ray coordinates of the differential ray and the central ray, respectively, on the transversal plane. $y_{+y,S}$ and $y_{\mathrm{cen},S}$ are the coordinates of the same rays on the source plane. In this example, the central ray (shown in black on Fig. 5) is traced along with a ray with a differential addition in input y coordinate (shown in red on Fig. 5). The difference in y coordinates of the central and δy differential ray determine the derivative.

The ray transfer matrix for a non-orthogonal optical system has 16 unknowns [see ABCD matrix in Eq. (10)], and each ray yields 4 quantities. To solve for every element of the matrix, 4 linearly independent rays must be traced. The simplest ray set is geometrically orthogonal,[40] where copies of the central ray (x, y, l, m) are modified by a differential quantity (δ) in each of the four ray coordinates, two in position (δx, δy) and two in slope (δl, δm). The differential ray set is given by the following equation:

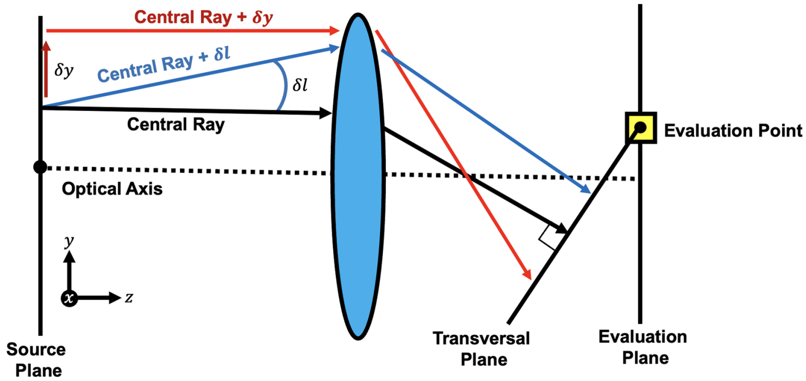

Fig. 5 Diagram illustrating differential ray tracing in the 2D case for a simple lens system. The central ray (black) is propagated along with two parabasal rays with a differential addition in the y direction (red) and another with a differential addition in the l direction (blue). To determine the ABCD matrix, the ray data are computed on the transversal plane, which is normal to the central ray and intersects the point in which we wish to evaluate the field.

$$\begin{pmatrix} x+\delta x \\ y \\ l \\ m \end{pmatrix}, \begin{pmatrix} x \\ y+\delta y \\ l \\ m \end{pmatrix}, \begin{pmatrix} x \\ y \\ l+\delta l \\ m \end{pmatrix}, \begin{pmatrix} x \\ y \\ l \\ m+\delta m \end{pmatrix}, \tag{13}$$

which are traced in addition to the central ray of interest. The full differential ray transfer matrix is given by Eq. (14). The ray transfer matrix is purely a function of the Cartesian position of the ray $((x, y)$ and slope of the ray in those directions (l, m) at the input and output of the optical system. An example of propagating a ray from the source plane to the transversal plane is shown in Eq. (14). Here the subscript S refers to the coordinate on the source plane, and the subscript T refers to the coordinate on the transversal plane. The elements of Eq. (14) are computed in a similar fashion to the example in Eq. (12) but with different ray data:

$$\begin{pmatrix} x_T & y_T & l_T & m_T \end{pmatrix} = \begin{pmatrix} \frac{\partial x_T}{\partial x_S} & \frac{\partial x_T}{\partial y_S} & \frac{\partial x_T}{\partial l_S} & \frac{\partial x_T}{\partial m_S} \\ \frac{\partial y_T}{\partial x_S} & \frac{\partial y_T}{\partial y_S} & \frac{\partial y_T}{\partial l_S} & \frac{\partial y_T}{\partial m_S} \\ \frac{\partial l_T}{\partial x_S} & \frac{\partial l_T}{\partial y_S} & \frac{\partial l_T}{\partial l_S} & \frac{\partial l_T}{\partial m_S} \\ \frac{\partial m_T}{\partial x_S} & \frac{\partial m_T}{\partial y_S} & \frac{\partial m_T}{\partial l_S} & \frac{\partial m_T}{\partial m_S} \end{pmatrix} \begin{pmatrix} x_S \\ y_S \\ l_S \\ m_S \end{pmatrix}. \tag{14}$$

With differential ray tracing, we are able to propagate a single Gaussian beam through an arbitrary optical system using the ABCD matrix. To perform GBD, we next need to understand how to decompose the field in the entrance pupil of the optical system.

2.3 Entrance Pupil Spatial Decomposition

The final variable to constrain in GBD is how to appropriately decompose the field in the entrance pupil into a finite set of Gaussian beams. This problem is illustrated in 1D earlier in Fig. 2; however, for imaging systems, the decomposition is a 2D problem. The fundamental Gaussian mode does not represent a complete set,[61] and therefore, the decomposition of the field is not unique. We must carefully consider how the beamlets are distributed in the entrance pupil for accurate diffraction calculations.

Various sampling schemes exist in the literature, with different strengths and weaknesses. The even Cartesian sampling scheme [shown in Fig. 6(a)] described by Harvey et al.[41] is the most straightforward, where the beamlets lie evenly spaced along a Cartesian grid. The ray coordinates in the entrance pupil are then computed from an overlap factor (OF), which describes the overlap of the beamlets' $1/e$ waist radii w_o:

$$\mathrm{OF} = \frac{N_g 2\omega_o}{W}, \tag{15}$$

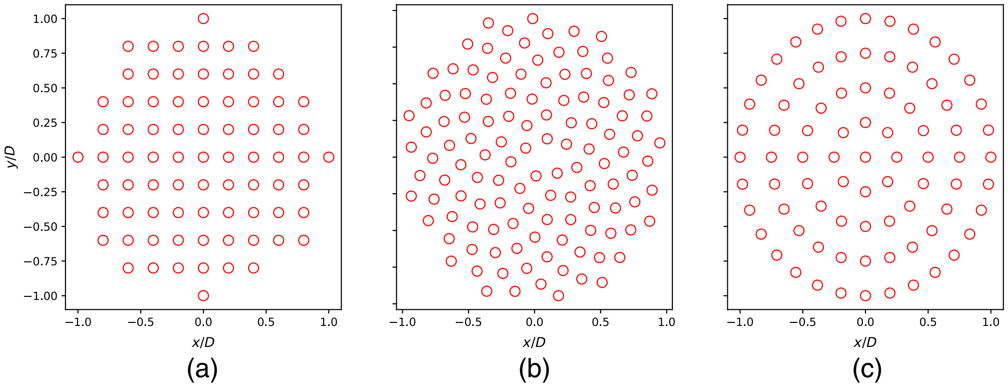

Fig. 6 Illustration of the (a) even, (b) Fibonacci, and (c) polar sample schemes for decomposing the field at the entrance pupil of the optical system with approximately the same number of beamlets in each figure. Quantifying the ramifications of these sampling schemes is of paramount importance to accurate diffraction simulation.

where N_g is the number of Gaussian beamlets across an aperture, and W is the width of the aperture. This feature is easy to implement and understand, but for undersampled cases, it introduces artifacts due to the ripple from the distribution and soft edge left by the Gaussian beamlets.

The Fibonacci sampling scheme [shown in Fig. 6(b)] introduced by Worku and Gross places the beamlets along a Fibonacci spiral, which results in a more accurate decomposition for circular apertures.[45] The distribution of the beamlets is even along polar angles on the spiral. The polar distribution of the beamlets is given by a position R and angle Θ:

$$R = \frac{W}{2}\sqrt{N_g}, \tag{16}$$

$$\Theta = \frac{2\pi}{\phi^2}N_g, \tag{17}$$

where ϕ is the golden ratio and N_g is the total number of beamlets to trace. Even polar sampling [shown in Fig. 6(c)] is also a viable method to increase the accuracy of the decomposition for fewer beamlets assuming the optical system has a circular aperture[45] but was not explored in this study due to the apparent advantages of the Fibonacci sample scheme, which are shown in Sec. 4.

3 Proposed Beamlet Propagation Algorithm

The transfer matrix method of GBD is the preferred algorithm to develop because of the recent developments by Worku and Gross to include modified GBD.[46] However, because the implementation is not public we must derive the full algorithm using the concepts described in Sec. 2. The basic concept of propagating a single beamlet through an arbitrary optical system is illustrated in Fig. 7. A Gaussian beam is placed at some position $\mathbf{r}_{\text{cen},S}$ in the source plane where the initial decomposition occurs. The central ray (shown in black on Fig. 7) that tracks the position of the Gaussian beam and the differential rays (shown in light red and blue on Fig. 7) that define the propagation are traced to the evaluation plane using a ray tracing engine. The central ray position ($\mathbf{r}_{cen,E}$) and direction ($\mathbf{k}_{cen,E}$) on the evaluation plane are used to define the transversal plane, which includes the point where we wish to evaluate the Gaussian field. The differential rays and the central ray are transformed to the transversal plane in order to compute the ABCD matrix that describes the propagation of the Gaussian beam from the source plane to the transversal plane.

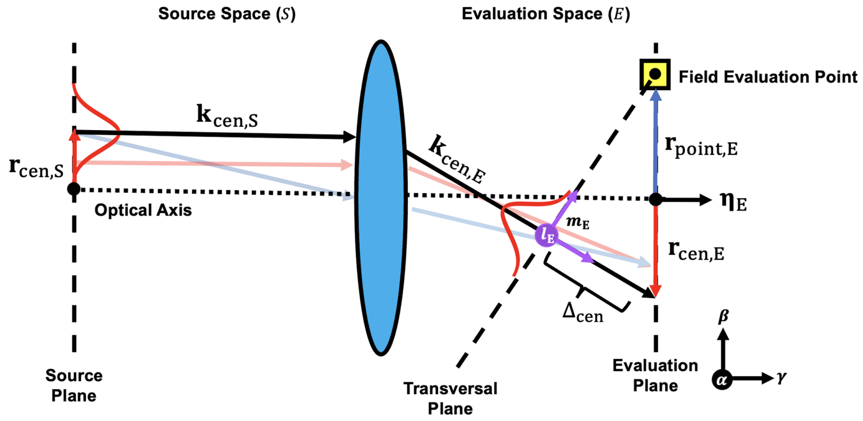

Fig. 7 Diagram illustrating the geometry of GBD and the relevant data used in the propagation calculation for a single beamlet (shown in red). GBD begins by tracing the central ray (black) and differential rays (light red and blue) from the source plane (left) to the evaluation plane (right). The rays must be propagated from the evaluation plane to the transversal plane, which is normal to the central ray and intersects a point at which the field is evaluated (yellow box). The difference of the differential ray data with the central ray data on the transversal plane gives us the ABCD matrix used to propagate a Gaussian beamlet. We evaluate the Gaussian beamlet at the intersection of the transversal plane with the evaluation plane to get the beamlet's contribution to the total field at that point. This process is repeated for each point and each beamlet. The coherent superposition of the beamlets at the field evaluation points yields the final propagated field.

We use this matrix to compute the influence of this beam on the field evaluation point. Every beamlet used in the initial field decomposition is propagated to each point on the evaluation plane this way, and the coherent superposition of the beamlets at the evaluation plane represents the propagated field.

In this section, we derive the formula to compute the propagation of a ray to the transversal plane, and how to use these data to compute the ABCD matrix and the contribution of a Gaussian beamlet to a point on the evaluation plane using the methods described in Sec. 2. We refer to several vectors in the algorithm below that are expressed using the convention $\mathbf{u}_{v,W}$. $\mathbf{u}$ is the data type of the vector, typically $\mathbf{r}$ if a position or $\mathbf{k}$ if a direction. v denotes what item the vector belongs to, "cen" means it belongs to the central ray and "point" means a point on the evaluation plane. W refers to the coordinate system of the vector, S for "source space," E for "evaluation space," or T for "transversal plane." The relevant parameters used in the propagation algorithm are shown in Fig. 7 and are referenced throughout the procedure below.

3.1 Propagating Rays to the Transversal Plane

A GBD simulation begins by running a ray trace though an optical system in the user's preferred design code. GBD needs the ray data at the plane where the decomposition occurred (the source plane) and the plane where we choose to observe the field (the evaluation plane). We use the ray data in evaluation space to propagate the rays to the transversal plane, where Eq. (8) is valid.

We first need to derive the propagation distance Δ_{ray} for the central and four differential rays. To do so, we find the intersection of the line defined by the ray we want to propagate $\mathbf{k}_{\text{ray},E}$ and the plane normal to the central ray of the Gaussian beam $\mathbf{k}_{\text{cen},E}$, which intersects the evaluation point $\mathbf{r}_{\text{point},E}$. This plane is the transversal plane and is defined by the following equation:

$$\mathbf{k}_{\text{cen},E} \cdot (\mathbf{r} - \mathbf{r}_{\text{point},E}) = 0. \tag{18}$$

The line along the ray is defined by the following equation:

$$\mathbf{r} = \mathbf{r}_{\text{ray},E} + \mathbf{k}_{\text{ray},E}\Delta_{\text{ray}}, \tag{19}$$

where $\cdot$ denotes the dot product and $\mathbf{r}$ is the space of all points that satisfy Eqs. (18) and (19). To find the distance, a ray needs to propagate along its own path in free space to intersect the transversal plane, we substitute $\mathbf{r}$ in Eq. (18) for Eq. (19) and solve for Δ_{ray}. The result is given by the following equation:

$$\Delta_{\text{ray}} = -\frac{\mathbf{k}_{\text{cen},E} \cdot (\mathbf{r}_{\text{ray},E} - \mathbf{r}_{\text{point},E})}{\mathbf{k}_{\text{cen},E} \cdot \mathbf{k}_{\text{ray},E}}. \tag{20}$$

The expression in Eq. (20) is written for the general case of propagating a ray from the evaluation plane to the transversal plane. It is applied to the central and differential rays by substituting $\mathbf{r}_{\text{ray},E}$ and $\mathbf{k}_{\text{ray},E}$ for the position and direction associated with the ray to propagate. We update the ray positons by invoking Eq. (19). The ray position on the transversal plane ($\mathbf{r}'_{\text{ray},E}$) is given in the following equation:

$$\mathbf{r}'_{\text{ray},E} = \mathbf{r}_{\text{ray},E} + \mathbf{k}_{\text{ray},E}\Delta_{\text{ray}}. \tag{21}$$

However, these coordinates are still expressed in the basis of the evaluation plane. We must next rotate the ray coordinates into the transversal plane coordinate system so that they are orthogonal to the propagation direction.

3.2 Transformation to the Transversal Plane

In the evaluation plane coordinate system, we define the orthogonal basis vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ (shown in black on Fig. 7 to the right of the evaluation plane), to be the directions along the x, y, and z axes, respectively. Similarly, the transversal plane is defined by basis vectors $\mathbf{l}_E$, $\mathbf{m}_E$, $\mathbf{k}_{\text{cen},E}$, (shown in purple on Fig. 7), which are the analogous x, y, and z directions for the transversal plane. To compute these directions, we start by taking the cross product of the central ray $\mathbf{k}_{cen,E}$ with the surface normal of the evaluation plane $\boldsymbol{\eta}_E$ (shown in black on the right of Fig. 7):

$$\mathbf{l}_E = \mathbf{k}_{\text{cen},E} \times \eta_E. \tag{22}$$

A feature of Eq. (22) is that the $\mathbf{l}_E$ is orthogonal to both the central ray and the surface normal. Similarly, to determine our final basis vector, which is mutually orthogonal to both the central ray and $\mathbf{l}_E$ basis vector, we compute their cross product:

$$\mathbf{m}_E = \mathbf{k}_{\mathrm{cen},E} \times \mathbf{l}_E. \tag{23}$$

These vectors form a complete basis that describe the coordinate system of the transversal plane but are expressed in the coordinate system of the evaluation plane. Therefore, we can construct an orthogonal transformation matrix that performs a rotation of basis from the evaluation plane to the transversal plane. This matrix is shown in the following equation:

$$\mathbf{O} = \begin{pmatrix} l_\alpha & l_\beta & l_\gamma \\ m_\alpha & m_\beta & m_\gamma \\ k_\alpha & k_\beta & k_\gamma \end{pmatrix}, \tag{24}$$

where $\mathbf{l}_E$, $\mathbf{m}_E$, and $\mathbf{k}_{\mathrm{cen},E}$ are written as l, m, k, respectively, for brevity and are shown in Eq. (24) as their components projected onto the vectors $\alpha, \boldsymbol{\beta}$, and $\boldsymbol{\gamma}$. The matrix $\mathbf{O}$ is the tool we need to express our ray coordinates in terms of the transversal plane basis. This is done by a simple multiplication of the ray position and direction vectors ($\mathbf{r'}_{\mathrm{ray},E}$, $\mathbf{k}_{\mathrm{ray},E}$) with the matrix, shown in Eqs. (25) and (26):

$$\mathbf{r}_{\mathrm{ray},T} = \mathbf{O}\mathbf{r'}_{\mathrm{ray},E}, \tag{25}$$

$$\mathbf{k}_{\mathrm{ray},T} = \mathbf{O}\mathbf{k}_{\mathrm{ray},E}. \tag{26}$$

The same must also be done for the field evaluation point of interest $\mathbf{r}_{\mathrm{point},E}$ to evaluate the Gaussian field later, shown in the following equation:

$$\mathbf{r}_{\mathrm{point},T} = \mathbf{O}\mathbf{r}_{\mathrm{point},E}. \tag{27}$$

Now that the ray data are expressed in the transversal plane coordinate system, we can use them to compute the differential ray transfer matrix used to propagate the Gaussian beamlet.

3.3 Computing the Differential Ray Transfer Matrix

To compute the differential ray transfer matrix for a Gaussian beam, we require the five ray coordinates and directions derived in the previous section ($\mathbf{r}_{\mathrm{ray},T}$ and $\mathbf{k}_{\mathrm{ray},T}$), as well as the data for the same rays on the source plane ($\mathbf{r}_{\mathrm{ray},S}$ and $\mathbf{k}_{\mathrm{ray},S}$). The differential ray transfer matrix is calculated using the methods discussed in Sec. 2. Recall that an element of the ray transfer matrix is given by the difference of the central and differential ray coordinates at the source and evaluation plane [see Eq. (12)]. The resultant matrix is given by Eq. (28). The full matrix including the ray data used to compute each of the elements of Eq. (28) can be found in Appendix A:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_T}{\partial x_S} & \frac{\partial x_T}{\partial y_S} & \frac{\partial x_T}{\partial l_S} & \frac{\partial x_T}{\partial m_S} \\ \frac{\partial y_T}{\partial x_S} & \frac{\partial y_T}{\partial y_S} & \frac{\partial y_T}{\partial l_S} & \frac{\partial y_T}{\partial m_S} \\ \frac{\partial l_T}{\partial x_S} & \frac{\partial l_T}{\partial y_S} & \frac{\partial l_T}{\partial l_S} & \frac{\partial l_T}{\partial m_S} \\ \frac{\partial m_T}{\partial x_S} & \frac{\partial m_T}{\partial y_S} & \frac{\partial m_T}{\partial l_S} & \frac{\partial m_T}{\partial m_S} \end{pmatrix}. \tag{28}$$

The total ray transfer matrix can be organized into a tensor composed of 2 × 2 submatrices shown in the left-hand side of Eq. (28). These are exactly the non-orthogonal representations of the ray transfer matrix discussed in Sec. 2 that we need to compute the propagated Gaussian field profile.

3.4 Computing the Gaussian Field

The central ray serves as the coordinate origin for our Gaussian field evaluation on the transversal plane. With the position vectors effectively transformed in Eqs. (25) and (27), we can define our centered radial coordinate $\mathbf{r}_o$ as the simple difference of these two positions to center the coordinate system in the following equation:

$$\mathbf{r}_o = \mathbf{r}_{\mathrm{point},T} - \mathbf{r}_{\mathrm{cen},T}. \tag{29}$$

Finally, we can call upon Eq. (8) to perform the Gaussian field evaluation, with some added phase factors to account for the free-space propagation of the Gaussian beamlet. The result is shown in the following equation:

$$U(\mathbf{r}_o) = \frac{U_o}{\sqrt{\det|\mathbf{A}+\mathbf{B}\mathbf{Q}_1^{-1}|}} \exp\left[\frac{-ik}{2}\mathbf{r}_o^{\mathrm{T}}\mathbf{Q}_2^{-1}\mathbf{r}_o + ik\Delta_{\mathrm{cen}} + ik\Phi_{\mathrm{opd}}\right], \tag{30}$$

where $\mathbf{Q}_2^{-1}$ is the propagated complex curvature matrix given by Eq. (7), $\mathbf{r}_o^{\mathrm{T}}$ is the transpose of the coordinate from Eq. (29), Δ_{cen} is the propagated distance of the central ray to the transversal plane, and Φ_{opd} is the optical path experienced by the central ray through the optical system, which we get from the ray tracing engine used. Note that while the phase factors Δ_{cen} and Φ_{opd} are not present in Eq. (8), they are necessary for GBD to correctly interfere all of the Gaussian beams that are propagated.

The procedure outlined above is repeated for each beamlet and location on the evaluation plane. From a computational perspective, this procedure is somewhat daunting. Repeating this process means the computation complexity scales by the number of beamlets and number of points on the evaluation plane, which can quickly become prohibitive. Fortunately, the algorithm described is entirely vector and matrix operations, so it is simple to take advantage of broadcasted array operations in Python to vectorize the computation and parallel computing to accelerate it. In our implementation, the vector operations described above are broadcasted such that the field of a single beamlet at all points is computed simultaneously. This operation exists in a loop over the number of beamlets (for our preliminary efforts in computational acceleration, see Appendix B).

Note the subtlety that the propagation procedure does not depend on the elementary field of choice until the electric field is evaluated. The assumption of this propagation method is that the elementary field is paraxial about the area encompassed by the differential rays. In principle, as long as a field's propagation formula is known analytically via the general Collins integral, it can be propagated with this method. Thus GBD is a special case of the method described in this section. Because of the history of modeling resonators with ray transfer matrices, we know the ABCD propagation laws for Laguerre–Gaussian[62] and Hermite–Gaussian[63] beams, which form complete sets. This indicates that the beamlet decomposition method would be of higher accuracy when decomposing the field into modes of higher spatial order. The formula for flattened elliptical Gaussian beams is also known,[63] which are capable of mitigating the soft-edge effect imposed by the traditional beamlet decomposition.[47] This method also works with Worku and Gross's half- and quarter-truncated Gaussian beamlets.[46] The algorithm described in this section is one of the key results of this study, because it can propagate the field of any known solution to the Collins integral to an arbitrary array of points. We build this algorithm into Poke to test its ability to mimic diffraction simulations with known results and perform a comparison against commercial software.

4 Results

To evaluate GBD as a viable physical optics propagation technique, we benchmark its performance versus traditional diffraction simulations for a given observatory coupled to a vortex coronagraph (VC). The fiducial observatory used in this study is a Ritchey–Chretien (RC) objective based on the HST using an unobscured aperture. This model is constructed in Zemax OpticStudio, using the system prescription is given in Table 1 and is illustrated in Fig. 8.

We first compare the PSFs generated by GBD to the analytical Airy function to assess the degree to which GBD can represent the focused field after a circular aperture. We then compare

Table 1 Optical system prescription for the RC telescope based on the HST used in this investigation. All distances are given in meters. RoC stands for radius of curvature, and the sign convention is chosen such that negative values are concave, and positive values are convex.

Surface	RoC (m)	Conic constant	Distance (m)	Diameter (m)
M1	−11.0400	−1.00230	−4.90607	2.40000
M2	1.35800	−1.49686	6.40620	0.28112

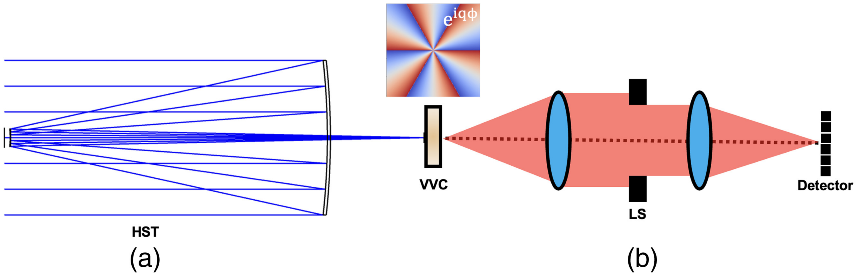

Fig. 8 Illustration of the hybrid propagation physics model used to produce the results. The system prescription in Table 1 is loaded into Zemax OpticStudio and shown in the left (labeled HST). The phase of a vortex coronagraph (VVC) is shown in the middle. The GBD PSF is computed at this plane and propagated through the coronagraph using HCIPy to arrive at the final image at the detector plane.

an aberrated GBD PSF to one produced by the Zemax Huygens PSF analysis tool to assess GBD's ability to reconstruct the field after a vignetted and aberrated wavefront. Finally, we propagate the focused field after a circular aperture produced by GBD through a Fraunhofer model of a VC and compare it to the results given using solely Fraunhofer diffraction. The PSF simulations conducted in Sec. 4.2 are monochromatic simulations at 1.65 μm on a detector with 256×256 pixels over 1×1 mm (or $25 \times 25 \frac{\lambda}{D}$). The coronagraph simulations conducted in Sec. 4.3 are conducted at the same wavelength but with 1600×1600 pixels across 8×8 mm (or $200 \times 200 \frac{\lambda}{D}$) to better sample the vortex mask and reach the desired contrast levels for both the hybrid and Fraunhofer model.

4.1 Fiducial Coronagraph

Our goal in this study is to assess the feasibility of GBD to integrate ray models of observatories into physical optics models of coronagraphs accurately. To quantify this, we propagate the images produced by GBD and the analytical Airy function through a charge-2 VC.[64,65] The optical VC is the general case of the vector VC,[66] which has shown great promise for future missions to image Earthlike exoplanets.[67] These coronagraphs are excellent at rejecting low-order spatial modes while transmitting the remainder of the light. We expect that traditional GBD will have some difficulty in accurately modeling high-spatial frequency content, and that it will manifest in the focal plane of this fiducial coronagraph if the error is limiting. If not, then we can conclude that GBD is suitable for high-contrast imaging simulation.

The complex amplitude of the charge-q VC FPM is given by

$$U(x, y) = \exp\left[iq \arctan\left(\frac{y}{x}\right)\right], \tag{31}$$

where q is the topological charge and the transmission is unity everywhere except for the center pixel, where it is 0. This is because there is a singularity in the phase ramp at this location, so it must be masked out. We chose the VC because of its ability to effectively reject on-axis starlight at a given wavelength. Should GBD introduce undesirable artifacts into the PSF, it should be visible in the coronagraph focal plane. Modeling a VC accurately is challenging computationally, because the on-axis starlight is only completely rejected if the focal plane is infinitely sampled. The singularity at the center must also be sampled highly in order to accurately sample the rapid change in phase immediately around it without discretization errors. These require very large arrays ($\geq$16k × 16k arrays) for meaningful starlight rejection, which considerably slows the simulation. To overcome this computational burden, a multistep propagation algorithm can be used to sample the central singularity higher than the rest of the field. HCIPy[27] has this algorithm implemented in the VC class, which accepts a user-specified wavefront and then outputs the wavefront after the VC and before the Lyot stop. We can define our GBD PSF as an HCIPy wavefront and propagate it through the coronagraph to complete our hybrid propagation model and analyze the image plane residuals. We can then compare the residuals of our hybrid propagation model to one where the PSF of a circular aperture was computed with Fraunhofer diffraction. This permits us to compare the scale of the error introduced by the hybrid propagation model against the numerical simulation errors present in traditional diffraction simulation.

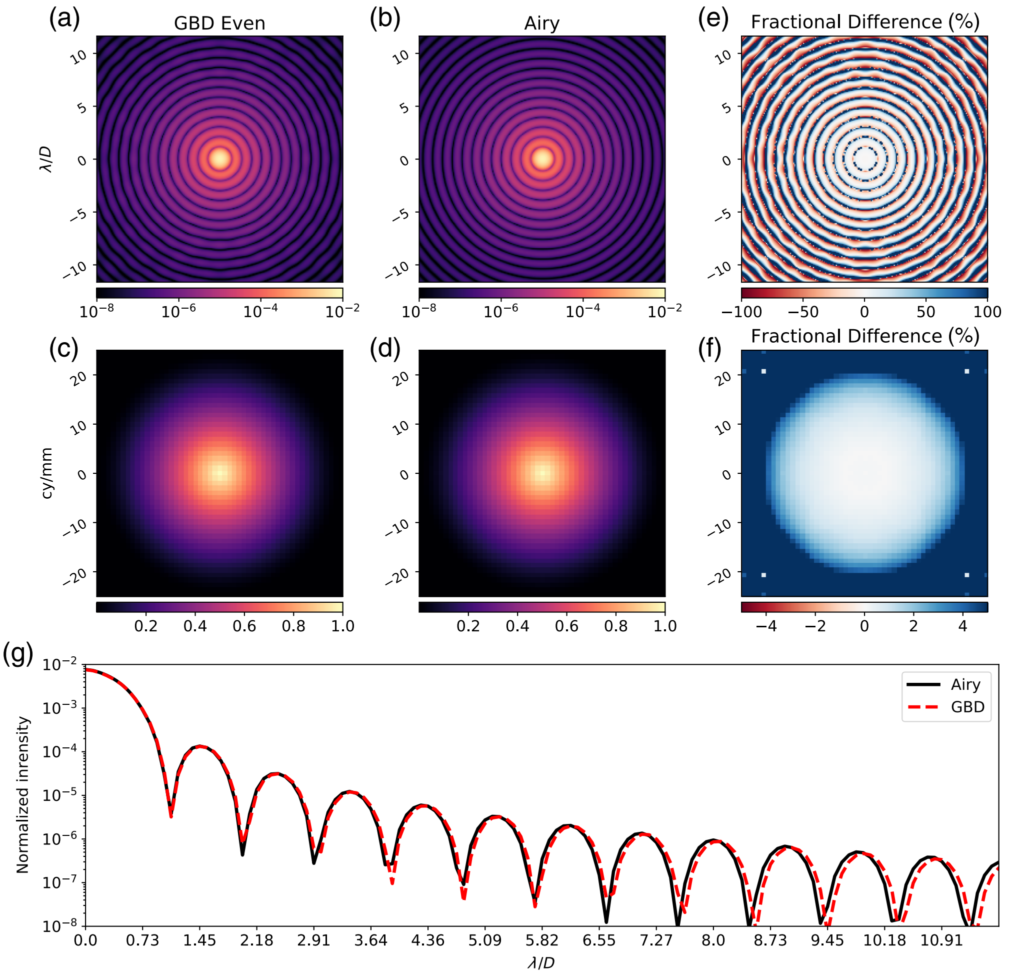

Fig. 9 Comparisons of the (a,b,e) PSF and (c,d,f) MTF with (a,c) even sampling and (b,d) the analytical Airy pattern. The PSFs are given in units of normalized intensity, such that the sum of the energy in the PSF is unity. (e,f) The fractional difference and (g) the azimuthally averaged radial profile. The MTF is plotted out to the cutoff frequency of the HST of around 25 cy/mm. The radial oscillation in the fractional difference PSF is indicative of the artifacts introduced by GBD. In the MTF, it is apparent that frequencies below 20 cy/mm are well-maintained, but the higher spatial frequencies increase in fractional difference. The effect on the PSF is revealed in the radial profile, where the PSF appears to spread out at larger angular separations. The RMS difference of the PSF data is 2.3×10^{-6}.

4.2 Observatory PSF

First, we examine GBD's ability to construct the Airy pattern. The Airy pattern represents the "ideal" diffraction-limited observatory image for a circular aperture. The analytical solution for this image is known and available in POPPY, so we have a point of comparison that is not limited by numerical simulation errors. Figures 9 and 10 show the PSF simulations generated by GBD using the even and Fibonacci sampling schemes. We also plot a comparison of the modulation transfer function (MTF) to better illustrate how the transfer of individual spatial frequencies is affected by GBD. An important parameter in these simulations was the degree in which the effective diameter of the entrance pupil was appropriately captured. In Figs. 2(c) and 2(d), we observe that GBD results in some energy spillover outside of the original aperture function, which would result in a PSF with an incorrect spatial extent. To mitigate this effect, we remove any beamlets within half a waist radius of the aperture boundary. This is a trade-off between too much energy outside of the aperture, which results in a smaller PSF and too little energy within the aperture, resulting in a larger PSF.

Figures 9 and 10 compare the same simulation of the Airy function where the only variable is how the entrance pupil was decomposed. In both the PSF and MTF dimensions, it is clear that

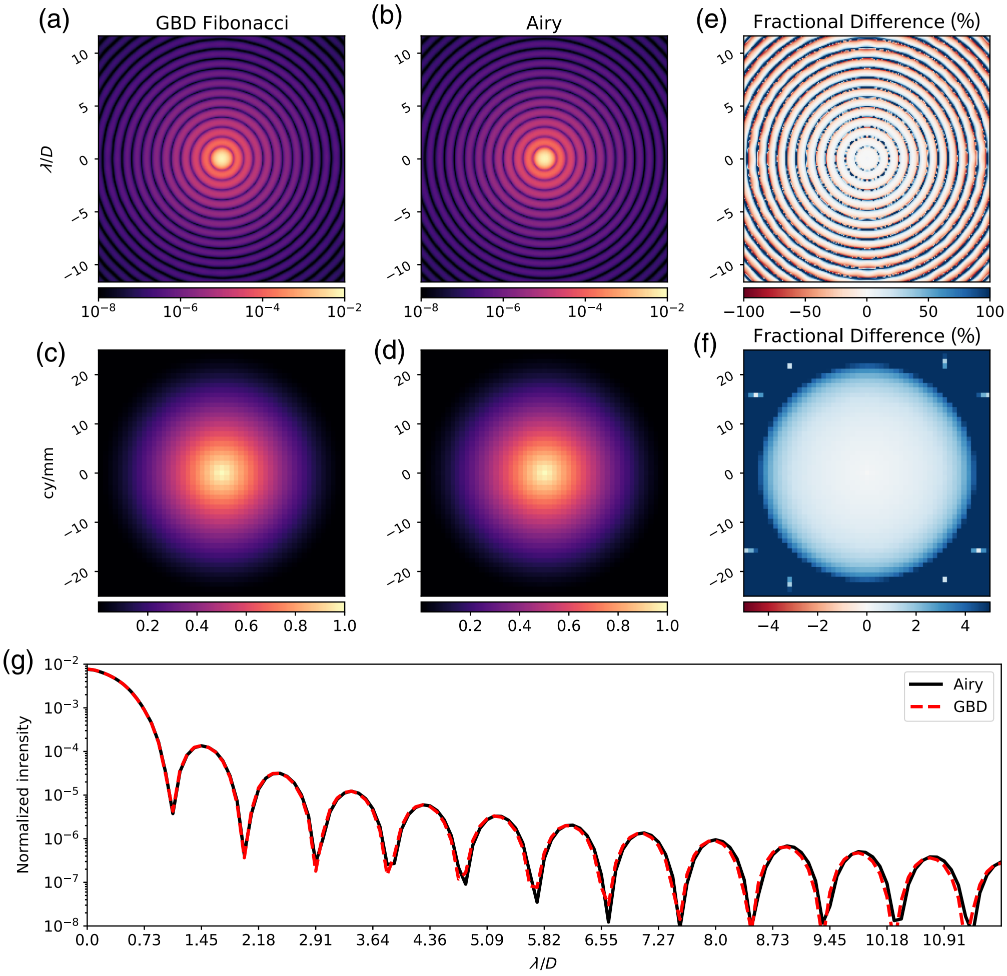

Fig. 10 Comparisons of the (a,b,e) PSF and (c,d,f) MTF with (a,c) Fibonacci sampling for the same number of beamlets as Fig. 9(a) and (b,d) the analytical Airy pattern. The PSFs are given in units of normalized intensity, such that the sum of the energy in the PSF is unity. (e,f) The fractional difference and (g) the azimuthally averaged radial profile. The MTF is plotted out to the cutoff frequency of the HST of around 25 cy/mm. The presence of ripples in the fractional difference PSF is noticeably lower than Fig. 9, and the fractional difference MTF shows that frequencies out to 23 cy/mm are well-maintained. The spreading present in Fig. 9 is less prevalent in the radial profile. The RMS difference of the PSF data is 1.6×10^{-6}, indicating that this distribution results in a more accurate simulation.

the Fibonacci sampling is the superior decomposition method for this aperture. For more undersampled cases, there may be a tradeoff in accuracy. To understand this, in Fig. 11, we examine how well the analytical Airy function is reconstructed for the two sample schemes as a function of the number of beamlets.

The Fibonacci sampling clearly wins out in terms of performance given a fixed number of beamlets, which translates directly to computation time. This also means that using the Fibonacci sample scheme can yield a simulation of the same accuracy as the even sample scheme with fewer beamlets. By judicious choice in sample scheme, the computational complexity of GBD can be lessened or the simulation accuracy can be increased.

To further test GBD's capability to model a more realistic observatory PSF, we add obscurations in the entrance pupil that corresponds to the secondary mirror and supporting spiders. We also tilt the secondary mirror by 0.05 deg to aberrate the beam. By doing so, we simultaneously test our algorithm's ability to capture aberrations from optical misalignment and diffraction from structure in the aperture. The results of this simulation are shown in Fig. 12.

The data in Fig. 12 were simulated in a comparison between our proposed algorithm and Zemax's Huygens PSF analysis tool. The Huygens PSF is computed by propagating spherical

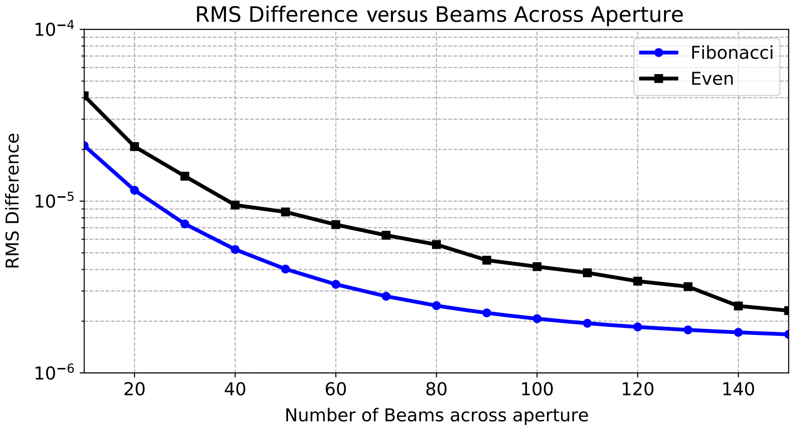

Fig. 11 RMS difference between the sum-normalized GBD PSF and analytical Airy function for the Fibonacci (blue circles) and even (black squares) sample schemes. For the circular aperture, it is clear that the Fibonacci sample scheme is more accurate for every case given a circular aperture. However, the returns are diminishing as the number of beamlets increases.

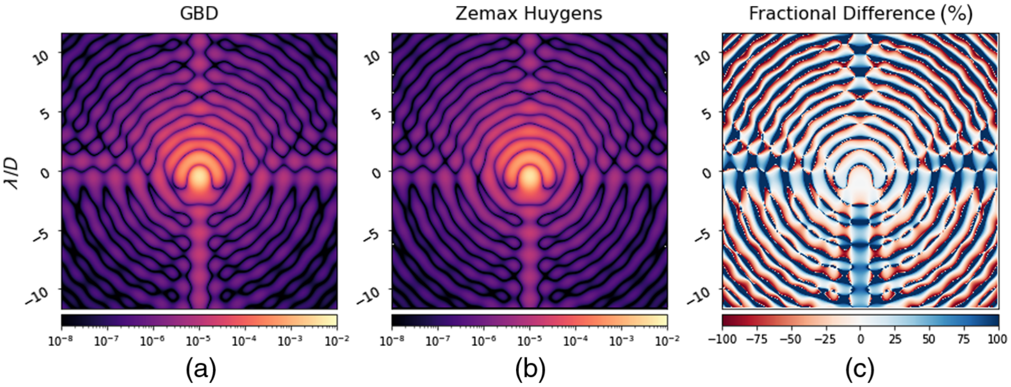

Fig. 12 Simulations of an aberrated PSF with secondary support structure in the entrance pupil of our HST model given in Table 1. (a) The GBD PSF and (b) Zemax Huygens PSF have very similar structure with (c) the dominant difference in structure being the features from the sharp-edge diffraction from the spiders. This simulation helps quantify GBD's difficulty in modeling sharp-edge diffraction effects. The RMS difference of the PSF data is 7.5×10^{-6}.

waves along ray paths in Zemax to the image plane and coherently summing them. The Zemax documentation asserts that this method makes fewer assumptions than the FFT PSF, so it was chosen as the point of comparison for the aberrated PSF. In our GBD simulation, we vignette any Gaussian beam that has any of its differential rays vignetted. Our results show extremely similar structure in the PSF from both the aberration induced by the secondary mirror misalignment and the structure from the HST spiders. The fractional difference reveals that the first couple "rings" of the PSFs are almost identical, indicating that the low-order aberrations were sufficiently simulated by GBD. There is a larger fractional difference in the dimmer structures that result from the sharp edges of the secondary obscuration and spiders, which is a known challenge for GBD simulations. This result can be further improved by careful implementation of alternative beamlet profiles, such as Worku and Gross's truncated beamlets,[46] or higher-order transverse electric field modes (e.g., Hermite and Laguerre–Gaussian). However, the result using fundamental Gaussian modes is encouraging since the RMS difference of the PSFs is within the same order of magnitude as the results shown in Figs. 9 and 10. Zemax is an industry standard optical modeling tool, but its propagation algorithms are closed-source. It is beyond the scope of this paper to validate the accuracy of Zemax's PSF simulation tools, but it is encouraging that our results in Fig. 12 agree. Now that our algorithm's ability to perform PSF simulations has been verified, we can analyze the degree to which it introduces artifacts in high-contrast imaging simulations.

Table 2 Simulation parameters for the result shown in Fig. 13.

Parameter	Value
Wavelength (λ)	1.65 μm
Number of Pixels (N_{pixels})	1600
Instantaneous FoV (Δx)	4.95 μm or $\frac{1}{8}\frac{\lambda}{D}$

4.3 Coronagraph Response

In traditional VCs, the on-axis field from an unvignetted circular aperture should be entirely rejected. We expect from the observatory PSF simulations that GBD does not trace high-spatial frequency information well due to the soft edges and amplitude ripples introduced by the beamlet decomposition. Any meaningful errors from this step should pass through the coronagraph unperturbed. To formally assess GBD's suitability for high-contrast imaging, we construct the PSF of the fiducial observatory with a circular aperture using GBD and compare it to a Fraunhofer model of the same system. Both PSFs are propagated with Fraunhofer diffraction through the VC and the coronagraph focal plane is compared to assess the presence, if any, of residual signal introduced by GBD. The parameters used in this simulation are given in Table 2, and the results shown in Fig. 13 are cropped to show the innermost $30\,\lambda/D$ to better display the structure near the inner working angle.

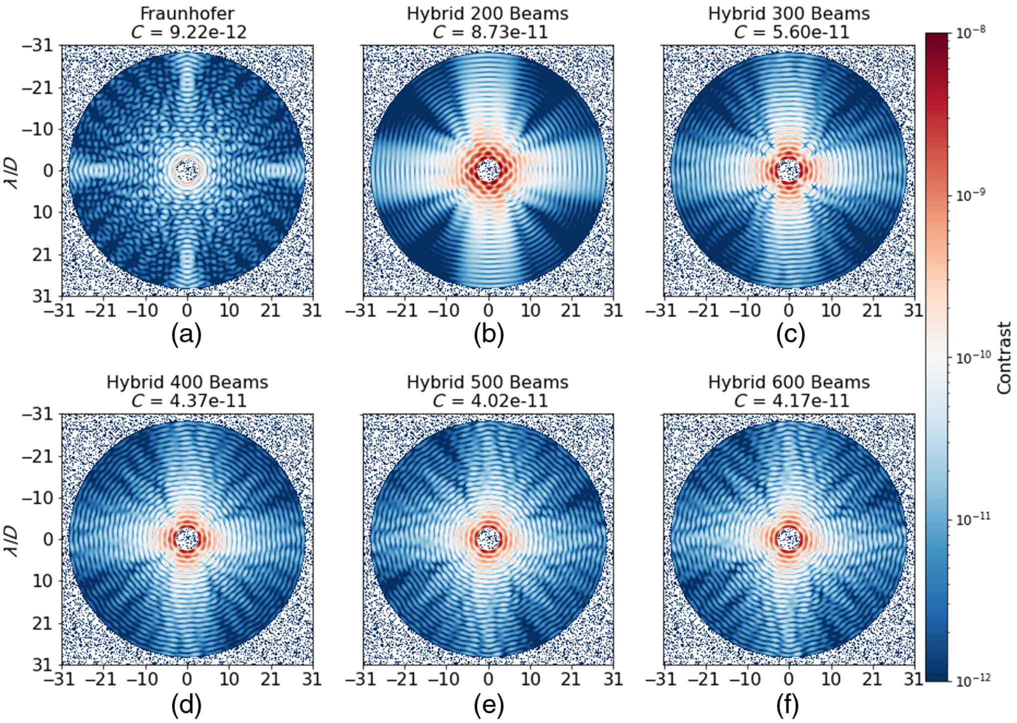

Fig. 13 Comparison of the coronagraph PSF from 3 to $30\,\lambda/D$ generated by a solely Fraunhofer diffraction model (a) and the proposed hybrid propagation scheme for a varying number of beamlets across the 2.4-m unobscured HST aperture with an OF of 2. (b) 200 beamlets, (c) 300 beamlets, (d) 400 beamlets, (e) 500 beamlets, and (f) 600 beamlets across the aperture. These data were generated by producing a PSF with the given propagation scheme and then propagating it through HCIPy's VortexCoronagraph class with a topological charge of 2. To reflect the residual starlight's astrophysical flux ratio "contrast," the PSF data are normalized to the maximum of an off-axis PSF at $\approx 16\,\lambda/D$ that propagates through the VC. The mean contrast of the masked region (C) is shown on each image. The residuals from (b)–(f) the hybrid propagation are brighter than the equivalent Fraunhofer simulation and minimize at the 500 beamlet case. This is particularly apparent near the inner working angle, where we see a maximum signal of $\approx 1 \times 10^{-8}$ for the 200 beamlet case decrease to $\approx 5 \times 10^{-9}$ in the 500 beamlet case.

Figure 13 shows the residuals that propagate through to the final image plane, which arise from inaccuracies in the propagation model. The hybrid propagation model generally minimizes in average contrast until the 600 beamlet cases [Fig. 13(f)], where there is a slight increase. The residuals from traditional Fraunhofer diffraction leave some low-energy features near the core of the PSF, which can be improved with greater sampling but the rest of the field is largely below 10^{-11} contrast. The decreasing residual energy with the number of beamlets used in the hybrid propagation scheme is encouraging, but we appear to have found the point in which increasing the number of Gaussian beams no longer increases simulation accuracy. The algorithm outlined in Sec. 3 is carried out as-written with some computational acceleration done by taking advantage of Python's ability to vectorize matrix operations and other Python packages to accelerate the exponential calculation, discussed in detail in Appendix B. We have not formally explored parallel processing packages on central processing units (CPUs) or graphical processing units (GPUs) to accelerate this computation, but expect that these could make higher-sampled simulations more feasible to minimize the artifacts that remain in GBD PSFs for high-contrast imaging simulations. Now that we have established an open-source platform for GBD, Worku's modified GBD[46] could be developed to minimize the number of beamlets required to minimize the artifacts in the coronagraphic focal plane. Exploring higher order spatial modes and the astigmatic fundamental mode of Gaussian beams could also yield greater accuracy for the same or less computational complexity. The result in Fig. 13 using traditional GBD shows that we can reduce the residuals to below 10^{-9} contrast near the inner working angle. It is also worth noting that this result suggests that with sufficient sampling, GBD can presently be used to simulate the PSFs for systems with less stringent contrast floors. To minimize the residuals from the propagation technique, other decomposition methods must be explored. We have created a platform for the algorithm's development in an open-source environment and will outline a road map for the most pressing optimizations that can be conducted by future investigations to improve GBD's accuracy.

5 Summary and Conclusions

Diffraction-limited optical systems require an accurate physical optics model to simulate the performance of the instrument. We formally describe an alternative implementation of the GBD physical optics propagation technique that is well-suited to PSF simulation. We illustrate the degree to which GBD is capable of accurately modeling the PSF of a Hubble-like astronomical observatory, and quantify the artifacts that remain in the "hybrid" simulation of a VC. We have demonstrated a new means of integrated observatory modeling, which reaches near state-of-the-art contrast levels at $<10^{-9}$ with an numerical average contrast of 4×10^{-11}. To our knowledge, this article is the first of its kind to explicitly publish the full transfer matrix GBD method and evaluate its ability in the context of astronomical telescopes. In doing so, we have made public a method of ray-based physical optics that has the potential to further integrate the optical modeling pipeline.

5.1 Future Work

The simulations presented in this article, while accurate, necessitated highly sampled simulations that were computationally intensive. The longest simulation used as many as 360,000 Gaussian beamlets across 2,560,000 pixels, which creates a phase array that is ∼100 TB in size. This simulation was done on an AMD Ryzen9 3950X processor and took ≈24 h to complete. For GBD to be a practical diffraction technique for high-contrast instrument sensitivity analysis, it would be ideal to generate a field of the same accuracy in a shorter amount of time. Preliminary efforts in accelerated computing are discussed in Appendix B, where we were able to reduce simulation runtime from taking ≈5 h to complete to ≈10 min. We intend to submit a follow-up study that outlines a more memory-efficient algorithm that we can use to more rapidly explore the degrees of freedom available in GBD (e.g., number of beamlets, OF, and etc.) for optimal diffraction simulation.

Modified GBD is already known to increase simulation accuracy for fewer beamlets[46] and as such is the next natural step in the development of our GBD module. Worku and Gross's work on truncated Gaussian beams showed high accuracy in reconstructing the field after 2D polygonal apertures and "squeezed" the half-truncated beamlets in the azimuthal direction to increase the accuracy of a field after a circular aperture. Hexagonal apertures are of particular interest to astronomy because of their use in telescopes, such as the W.M. Keck Observatory and James Webb Space Telescope. Understanding how the truncated beamlets are able to reconstruct the field after segmented apertures is another important step in developing GBD for high-contrast imaging simulations. It is also worth repeating the notion in Sec. 3 that nothing about the proposed algorithm requires the beamlets to be strictly Gaussian. Therefore, considering alternative beamlets to use in the field decomposition may increase the accuracy beyond what traditional GBD and modified GBD are capable of. GBD has previously demonstrated the ability to model plane-to-plane diffraction effects (e.g., the spot of Arago),[41] and its ability to model diffraction from surface polishing errors should be investigated for a more comprehensive modeling pipeline.

The beamlet decomposition algorithm would also benefit from more consideration into leveraging how parallelizeable it is. The contribution from each beamlet at each pixel is computed independently. With thoughtful consideration to the structure of our code and wide library of parallel processing packages available to us in Python on CPU's and GPUs,[68,69] the beamlet computation could be rapidly accelerated. Preliminary experiments on GPUs have shown a decrease in runtime by a factor of $\approx 16\times$. Several investigators have been exploring the Jax Python package for its ability to perform automatic differentiation and parallel computing in support of physical optics simulations.[70–72] Automatic differentiation could be extremely useful for future beamlet decomposition algorithms to improve the accuracy of the ABCD matrix computation. Parallelization and vectorization is the most natural path forward for our beamlet decomposition algorithm because of the independence of the beamlet operations. Jax makes this process simple with the "pmap" and "vmap" functions while also allowing for just-in-time compilation for accelerated computing.

The use of a beamlet decomposition algorithm completely integrates a diffraction model with a ray model of an optical system, resulting in a more physically complete modeling pipeline. Consequently, other ray-based analyses can be integrated directly into the diffraction model. Polarization ray tracing[5] is a natural extension to beamlet decomposition simulations because it can trace the complex amplitude of individual beamlets for generally vectorial field propagation through optical systems. This capability was demonstrated by Worku and Gross[44] in the context of high-numerical aperture microscope objectives but has the potential to be a powerful simulation tool for astronomical telescopes, including the next generation giant segmented mirror telescopes (Thirty Meter Telescope, Extremely Large Telescope, and Giant Magellan Telescope)[4] and the Astro2020-recommended IROUV space observatory, which may be sensitive to polarization aberrations.

5.2 Open-Source Science and Engineering

This research was inspired by POPPY, a Python physical optics module originally developed to simulate the James Webb Space Telescope. Our goal is to expand the capabilities of POPPY by investigating new propagation physics. We are developing the GBD module in a self-contained package with interfaces to other popular diffraction codes (POPPY and HCIPy) to provide ray information to diffraction models. The code used to prototype the GBD method can be found in the Poke repository on GitHub.[55] This repository is intended to be purely experimental as we develop the propagation physics for other high-contrast imaging packages that have substantive support.

6 Appendix A: Full Differential Ray Transfer Matrix Calculation

Equation (32) contains the explicit ray data used in the computation of each element of the differential ray transfer matrix. Each element of the matrix is given by the form $a_{\mathrm{ray},B}$, where a denotes the data (position x, y or direction cosine l, m), ray denotes which of the five rays the data are from ($+x$, $+y$, $+l$, $+m$, and cen), and B, which denotes the plane the data was taken from (S for source plane and T for transversal plane):

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{pmatrix} \frac{x_{+x,T}-x_{\mathrm{cen},T}}{x_{+x,S}-x_{\mathrm{cen},S}} & \frac{x_{+y,T}-x_{\mathrm{cen},T}}{y_{+y,S}-y_{\mathrm{cen},S}} & \frac{x_{+l,T}-x_{\mathrm{cen},T}}{l_{+l,S}-l_{\mathrm{cen},S}} & \frac{x_{+m,T}-x_{\mathrm{cen},T}}{m_{+m,S}-m_{\mathrm{cen},S}} \\ \frac{y_{+x,T}-y_{\mathrm{cen},T}}{x_{+x,S}-x_{\mathrm{cen},S}} & \frac{y_{+y,T}-y_{\mathrm{cen},T}}{y_{+y,S}-y_{\mathrm{cen},S}} & \frac{y_{+l,T}-y_{\mathrm{cen},T}}{l_{+l,S}-l_{\mathrm{cen},S}} & \frac{y_{+m,T}-y_{\mathrm{cen},T}}{m_{+m,S}-m_{\mathrm{cen},S}} \\ \frac{l_{+x,T}-l_{\mathrm{cen},T}}{x_{+x,S}-x_{\mathrm{cen},S}} & \frac{l_{+y,T}-l_{\mathrm{cen},T}}{y_{+y,S}-y_{\mathrm{cen},S}} & \frac{l_{+l,T}-l_{\mathrm{cen},T}}{l_{+l,S}-l_{\mathrm{cen},S}} & \frac{l_{+m,T}-l_{\mathrm{cen},T}}{m_{+m,S}-m_{\mathrm{cen},S}} \\ \frac{m_{+x,T}-m_{\mathrm{cen},T}}{x_{+x,S}-x_{\mathrm{cen},S}} & \frac{m_{+y,T}-m_{\mathrm{cen},T}}{y_{+y,S}-y_{\mathrm{cen},S}} & \frac{m_{+l,T}-m_{\mathrm{cen},T}}{l_{+l,S}-l_{\mathrm{cen},S}} & \frac{m_{+m,T}-m_{\mathrm{cen},T}}{m_{+m,S}-m_{\mathrm{cen},S}} \end{pmatrix}. \tag{32}$$

7 Appendix B: Accelerated Computing

The independence of the Gaussian beamlet operations is uniquely suited to the exploration of multithreaded computation to accelerate diffraction simulations. Accelerated computing is integral to diffraction modeling to enable rapid and precision simulation of small signals. The time to conduct traditional Fourier-based diffraction modeling is set by the complexity of the system. The sampling of each optical element and the number of total optical elements increase the complexity and number of fast Fourier transforms (FFTs) used, resulting in more computation time. GBD circumvents the FFT entirely by tracing rays to propagate through the optical system in a fraction of the time of the FFT. GBD's diffraction calculation at the plane of interest computes an exponential of a complex-valued array that scales with the number of beamlets and sampling of the image plane, resulting in longer computation times. Preliminary explorations into accelerated computing were conducted using the numexpr[69] and numba[68] Python packages that showed favorable computation time decreases by multithreading the operation on a CPU. Both packages operate by precompiling a given function into machine code that the program calls and breaking up the operation into chunks of arrays that a CPU core can handle efficiently. The key computational advantage of GBD is the ability to do diffraction calculations in parallel. Numba was the first package explored due to its ease of implementation. The package works by applying a decorator to a Python function that processes the large array of interest and then specifying the number of CPU cores for the process to use. The distribution of the information stored within the array is handled automatically by numba, and results in considerable runtime decreases for GBD. On a 16 core 2.4 GHz CPU, runtime for a simulation of 1876 Gaussian beamlets through a coronagraph to simulate a 256 × 256 pixel focal plane was sped up by a factor of 5, which approached POPPY's Fresnel diffraction runtime. This experiment was repeated using the numexpr package, which showed an even greater decrease in computation time, consistent with results for accelerating Fresnel diffraction in POPPY.[26] The comparison in runtime versus number of CPU cores is shown in Fig. 14.

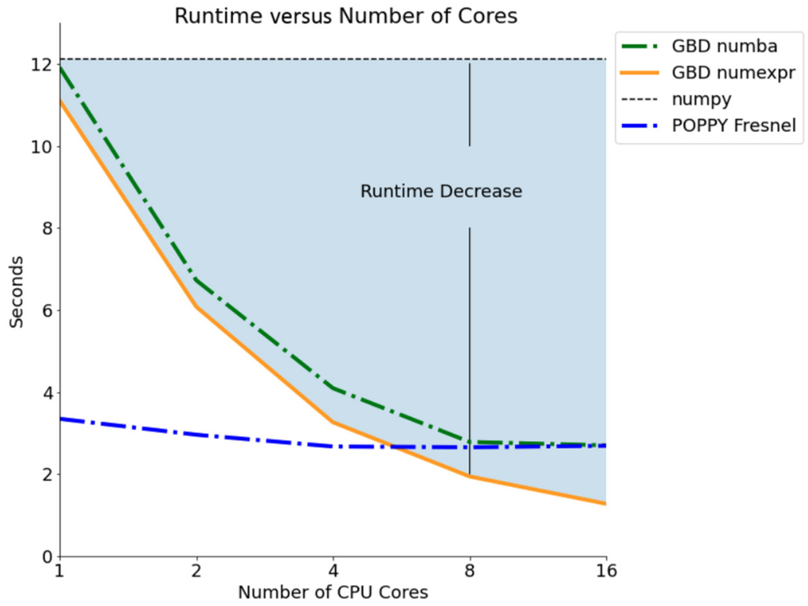

Fig. 14 Run time comparison for a 50 × 50 grid of Gaussian beamlets on a 256 × 256 detector grid versus number of CPU cores used. Numexpr considerably accelerates the exponential calculation given multiple processing cores and is an encouraging path forward for further parallelization.

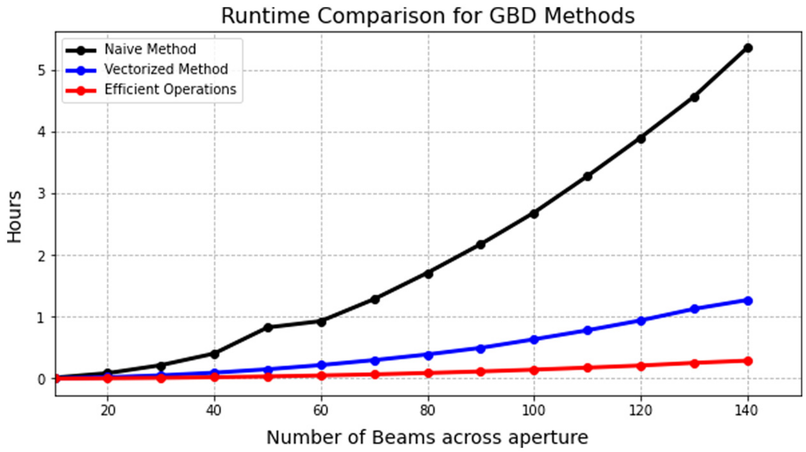

Fig. 15 Runtime comparison of the iterations of the GBD algorithm applied in Python 3.8. The version using nested for loops is plotted in black, and high-beamlet simulations were computationally prohibitive. The runtime for the vectorized code is shown in blue, and the vectorized code with more efficient linear algebra operations (determinant, inverse) is shown in red. Overall, the decrease in runtime from the black curve to the red curve is ≈40×, which enabled the highly sampled simulations in Fig. 13.

The vector operations used in the algorithm described in Sec. 3 can be broadcasted across the entire array, enabling more efficient computation of the ABCD matrix. Broadcasting refers to the practice of writing code such that a given operation is applied to each element of the array simultaneously. This is more commonly known as "vectorized" computing. Our initial demonstration of the GBD algorithm was written using nested for loops (black, Fig. 15), which was inefficient for highly sampled simulations. Vectorizing our code resulted in a ≈10× decrease in runtime (blue, Fig. 15). We then determined that the elementary matrix operations done using Numpy's linear algebra library were considerably dominating the runtime, so we wrote our own versions of the determinant and inverse functions. This change resulted in a net ≈40× decrease from the original code written using nested loops (red, Fig. 15). The runtimes for a simulation of various numbers of beams across the aperture to a 256×256 pixel detector are shown in Fig. 15.

We anticipate even greater speedups on GPUs. In particular, the creation of the phase array from the ray data is the slowest operation and needs to be parallelized for accelerated diffraction modeling. Using the Cupy python package, adding support for GPUs was made quite simple given its compatibility with the numpy API. Preliminary experiments using our beamlet decomposition algorithm show a ≈16× decrease in computation time versus the algorithm shown in red on Fig. 15. We have added GPU support though Cupy in our Poke package and plan to publish a formal study quantifying the degree to which GBD can benefit from accelerated computing in a future report.

Code and Data Availability

The code used to conduct these simulations is located on the Poke repository on GitHub in the GBD_Paper directory. The test_worku_transversal.py script was used to conduct the simulations in this article. These simulations were done during a very early release (v0.1.0) of Poke without much optimization for runtime or a user-friendly design. This version has been moved to the legacy branch of the repository as Poke and the GBD module is actively developed. Presently, the updated version of this algorithm exists in Poke v1.1.0 as the poke.gbd module.

Acknowledgments

This work benefited greatly from the insight of Dr. Norman Girma Worku, who pioneered the development of the transfer-matrix method of GBD. We would like to thank Trenton Brendel, Weslin Pulin, and Brandon Dube for helpful discussion on the fundamental physics and design of the Poke API; Kian Milani for the guidance with matplotlib; Marcos Esparza, Kevin Derby, Hyukmo Kang, and Ramya Anche for help proof-reading this article; and Sebastiaan Haffert for guidance on the use of HCIPy's VortexCoronagraph model. This research made use of several open-source Python packages, including POPPY,[24] HCIPy,[27] prsym,[29] numpy,[73] matplotlib,[74] ipython,[75] astropy[76–78] scipy,[79] numba,[68] and numexpr.[69] This research made use of high-performance computing resources supported by the University of Arizona TRIF (Technology and Research Initiative Fund), UITS, and Research, Innovation, and Impact and maintained by the UArizona Research Technologies Department. This work was supported by a NASA Space Technology Graduate Research Opportunity.

References

1. T. Andersen and A. Enmark, Integrated Modeling of Telescopes, Astrophysics and Space Science Library, Vol. **377**, Springer New York, New York (2011).
2. B. D. Dube et al., "Exascale integrated modeling of low-order wavefront sensing and control for the Roman coronagraph instrument," *J. Opt. Soc. A. A* **39**, C133 (2022).
3. J. W. Goodman, *Introduction to Fourier Optics*, Vol. **1**, 3rd ed., Roberts & Co. Publishers, Englewood, Colorado (2005).
4. R. Anche et al., "Polarization aberrations in next-generation giant segmented mirror telescopes (GSMTs). I. Effect on the coronagraphic performance," *Astron. Astrophys.* **672**, A121 (2023).
5. R. A. Chipman, W. S. T. Lam, and J. Breckinridge, "Polarization aberration in astronomical telescopes," *Proc. SPIE* **9613**, 96130H (2015).
6. D. M. Harrington and S. R. Sueoka, "Polarization modeling and predictions for DKIST. Part 1: Telescope and example instrument configurations," *Proc. SPIE* **9912**, 99126U (2017).
7. T. T. Hyde et al., "Integrated modeling activities for the James Webb Space Telescope: optical jitter analysis," *Proc. SPIE* **5487**, 588–599 (2004).
8. O. Guyon et al., "Theoretical limits on extrasolar terrestrial planet detection with coronagraphs," *Astrophys. J. Suppl. Ser.* **167**, 81–99 (2006).
9. C. C. Stark et al., "Lower limits on aperture size for an ExoEarth detecting coronagraphic mission," *Astrophys. J.* **808**, 149 (2015).
10. L. Pueyo, "Direct imaging as a detection technique for exoplanets," in *Handbook of Exoplanets*, H. J. Deeg and J. A. Belmonte, Eds., Springer, Cham (2018).
11. J. Lozi et al., "SCExAO, an instrument with a dual purpose: perform cutting-edge science and develop new technologies," *Proc. SPIE* **10703**, 1070359 (2018).
12. J. R. Males et al., "MagAO-X: project status and first laboratory results," *Proc. SPIE* **10703**, 1070309 (2018).
13. B. F. Castellá et al., "Commissioning and first light results of an L'-band vortex coronagraph with the Keck II adaptive optics NIRC2 science instrument," *Proc. SPIE* **9909**, 990922 (2016).
14. B. A. Macintosh et al., "The Gemini planet imager: first light and commissioning," *Proc. SPIE* **9148**, 91480J (2014).
15. J. L. Beuzit et al., "Sphere: the exoplanet imager for the very large telescope," *Astron. Astrophys.* **631**, A155 (2019).
16. R. I. Thompson, "Near infrared camera and multiobject spectrometer (NICMOS): the near infrared space mission on HST," *Proc. SPIE* **2209**, 319–330 (1994).
17. S. D. Horner and M. J. Rieke, "The near-infrared camera (NIRCam) for the James Webb Space Telescope (JWST)," *Proc. SPIE* **5487**, 628–634 (2004).
18. J. H. Girard et al., "JWST/NIRCam coronagraphy: commissioning and first on-sky results," *Proc. SPIE* **12180**, 121803Q (2022).
19. Decadal Survey on Astronomy and Astrophysics 2020 (Astro2020), Space Studies Board, Board on Physics and Astronomy et al., *Pathways to Discovery in Astronomy and Astrophysics for the 2020s*, p. 26141, National Academies Press, Washington, D.C. (2021).
20. J. M. Howard et al., "Integrated modeling activities for NASA's Roman Space Telescope," *Proc. SPIE* **12187**, 121870V (2022).
21. J. M. Howard, "Optical integrated modeling activities for the James Webb Space Telescope (JWST)," *Proc. SPIE* **8336**, 83360E (2011).
22. J. Krist, "Tiny Tim: an HST PSF simulator," in *Astronomical Data Analysis Software and Systems II*, R. J. Hanisch, R. J. V. Brissenden, and J. Barnes, Eds., Astronomical Society of the Pacific Conference Series, Vol. **52**, p. 536, Astronomical Society of the Pacific (1993).
23. J. E. Krist, "PROPER: an optical propagation library for IDL," *Proc. SPIE* **6675**, 66750P (2007).
24. M. D. Perrin et al., "Simulating point spread functions for the James Webb Space Telescope with WebbPSF," *Proc. SPIE* **8442**, 84423D (2012).
25. M. Perrin et al., "POPPY: Physical Optics Propagation in PYthon," Astrophysics Source Code Library, record ascl:1602.018 (2016).
26. E. S. Douglas and M. D. Perrin, "Accelerated modeling of near and far-field diffraction for coronagraphic optical systems," *Proc. SPIE* **10698**, 106982U (2018).

27. E. H. Por et al., "High Contrast Imaging for Python (HCIPy): an open-source adaptive optics and coronagraph simulator," *Proc. SPIE* **10703**, 1070342 (2018).
28. M. J. Townson et al., "AOtools: a python package for adaptive optics modelling and analysis," *Opt. Express* **27**, 31316–31329 (2019).
29. B. Dube, "prysm: a python optics module," *J. Open Source Software* **4**(37), 1352 (2019).
30. A. Allen, "Scicodes: astronomy research software and beyond," astro-ph.IM arXiV eprint 2111.14278 (2011).
31. J. E. Krist, L. Pueyo, and S. B. Shaklan, "Practical numerical propagation of arbitrary wavefronts through PIAA optics," *Proc. SPIE* **7731**, 77314N (2010).
32. R. J. Vanderbei, "Diffraction analysis of two-dimensional pupil mapping for high-contrast imaging," *Astrophys. J.* **636**, 528–543 (2006).
33. O. Guyon et al., "Phase induced amplitude apodization (PIAA) coronagraphy: recent results and future prospects," *Proc. SPIE* **8442**, 84424V (2012).
34. J. Krist et al., "WFIRST coronagraph flight performance modeling," *Proc. SPIE* **10698**, 106982K (2018).
35. J. E. Krist, B. Nemati, and B. P. Mennesson, "Numerical modeling of the proposed WFIRST-AFTA coronagraphs and their predicted performances," *J. Astron. Telesc. Instrum. Syst.* **2**(1), 011003 (2015).
36. B. Nemati, "EMCCD detect," https://github.com/wfirst-cgi/emccd_detect (2023).
37. M. Arko et al., "Pyxel 1.0: an open source Python framework for detector and end-to-end instrument simulation," *Proc. SPIE* **12187**, 1218705 (2022).
38. J. J. Wang et al., "pyKLIP: PSF subtraction for exoplanets and disks," Astrophysics Source Code Library, record ascl:1506.001 (2015).
39. B. Ren, "nmf_imaging, doi: 10.5281/zenodo.3738623," (2020).
40. A. W. Greynolds, "Vector formulation of the ray-equivalent method for general Gaussian beam propagation," *Proc. SPIE* **0679**, 129–133 (1986).
41. J. E. Harvey, R. G. Irvin, and R. N. Pfisterer, "Modeling physical optics phenomena by complex ray tracing," *Opt. Eng.* **54**(3), 035105 (2015).
42. "Modeling coherence," FRED Application Note, Photon Engineering, 2010, https://photonengr.com/wp-content/uploads/2011/04/appNoteCoherence.pdf.
43. "CODE V beam synthesis propagation," SYNOPSYS Datasheet, SYNOPSYS, 2018, https://www.synopsys.com/content/dam/synopsys/optical-solutions/documents/datasheets/codev-beam-synthesis-seo.pdf.
44. N. G. Worku and H. Gross, "Vectorial field propagation through high NA objectives using polarized Gaussian beam decomposition," *Proc. SPIE* **10347**, 103470W (2017).
45. N. G. Worku, R. Hambach, and H. Gross, "Decomposition of a field with smooth wavefront into a set of Gaussian beams with non-zero curvatures," *J. Opt. Soc. Am. A* **35**, 1091–1102 (2018).
46. N. G. Worku and H. Gross, "Propagation of truncated Gaussian beams and their application in modeling sharp-edge diffraction," *J. Opt. Soc. Am. A* **36**, 859–868 (2019).
47. A. W. Greynolds, "Ten years after (no, not the band): advancements in optical engineering computations over the decade(s)," *Proc. SPIE* **11483**, 114830C (2020).
48. A. E. Siegman, *Lasers*, University Science Books, Mill Valley, California (1986) [Includes bibliographical references and index].
49. J. Arnaud, "Representation of Gaussian beams by complex rays," *Appl. Opt.* **24**, 538 (1985).
50. S. A. Collins, "Lens-system diffraction integral written in terms of matrix optics," *J. Opt. Soc. Am.* **60**, 1168–1177 (1970).
51. N. G. Worku and H. Gross, "Gaussian pulsed beam decomposition for propagation of ultrashort pulses through optical systems," *J. Opt. Soc. Am. A* **37**, 98–107 (2020).
52. A. W. Greynolds, "Fat rays revisited: a synthesis of physical and geometrical optics with Gaußlets," *Proc. SPIE* **9293**, 92931U (2014).
53. J. N. Ashcraft and E. S. Douglas, "An open-source Gaussian beamlet decomposition tool for modeling astronomical telescopes," *Proc. SPIE* **11450**, 114501D (2020).
54. M. J. Hayford, "ray-optics," 2023, https://github.com/mjhoptics/ray-optics/
55. J. Ashcraft, "poke v0.1.0," 2022, https://zenodo.org/records/7117214.
56. J. N. Ashcraft et al, "Poke: an open-source, ray-based physical optics platform," *Proc. SPIE* **12664**, 1266404 (2023).
57. N. Inada et al, "msgpack-python," 2023, https://github.com/msgpack/msgpack-python.
58. Y. Cai and Q. Lin, "Decentered elliptical Gaussian beam," *Appl. Opt.* **41**(21), 4336–4340 (2002).
59. W. Brouwer, "Matrix methods in optical instrument design," Volume 2 of Lecture notes and supplements in physics, W.A. Benjamin (1964).
60. B. D. Stone and G. W. Forbes, "Differential ray tracing in inhomogeneous media," *J. Opt. Soc. Am. A* **14**, 2824–2836 (1997).
61. R. J. Koshel, "Novel methods of intracavity beam shaping," *Proc. SPIE* **4443**, 47–57 (2001).

62. Z. Mei, J. Guo, and D. Zhao, "Decentred elliptical Laguerre–Gaussian beam," *J. Opt. A: Pure Appl. Opt.* **8**, 77–84 (2005).
63. Y. Cai and Q. Lin, "The elliptical Hermite–Gaussian beam and its propagation through paraxial systems," *Opt. Commun.* **207**(1), 139–147 (2002).
64. D. Mawet et al., "Annular groove phase mask coronagraph," *Astrophys. J.* **633**, 1191 (2005).
65. J. H. Lee et al., "Experimental verification of an optical vortex coronagraph," *Phys. Rev. Lett.* **97**, 053901 (2006).
66. D. Mawet et al., "The vector vortex coronagraph: laboratory results and first light at Palomar observatory," *Astrophys. J.* **709**, 53 (2010).
67. E. Serabyn et al., "Vector vortex coronagraphy for exoplanet detection with spatially variant diffractive waveplates," *J. Opt. Soc. Am. B* **36**, D13 (2019).
68. S. K. Lam, A. Pitrou, and S. Seibert, "Numba: a LLVM-based python JIT compiler," in *Proc. Second Workshop LLVM Compiler Infrastruct. in HPC*, pp. 1–6 (2015).
69. R. McLeod, "pydata/numexpr: NumExpr v2.6.9," *Zenodo* (2018).
70. L. Desdoigts, B. Pope, and P. Tuthill, "Optical design, analysis, and calibration using dLux," *Proc. SPIE* **12180**, 1218032 (2022).
71. A. Wong et al., "Phase retrieval and design with automatic differentiation: tutorial," *J. Opt. Soc. Am. B: Opt. Phys.* **38**, 2465 (2021).
72. B. J. S. Pope et al., "Kernel phase and coronagraphy with automatic differentiation," *Astrophys. J.* **907**, 40 (2021).
73. C. R. Harris et al., "Array programming with NumPy," *Nature* **585**, 357–362 (2020).
74. J. D. Hunter, "Matplotlib: a 2D graphics environment," *Comput. Sci. Eng.* **9**(3), 90–95 (2007).
75. F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Comput. Sci. Eng.* **9**, 21–29 (2007).
76. Astropy Collaborationet al., "Astropy: a community Python package for astronomy," *Astron. Astrophys.* **558**, A33 (2013).
77. Astropy Collaborationet al., "The astropy project: building an open-science project and Status of the v2.0 core package," *Astron. J.* **156**, 123 (2018).
78. Astropy Collaboration et al., "The astropy project: sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package," *Astrophys. J.* **935**, 167 (2022).
79. P. Virtanen et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods* **17**, 261–272 (2020).

Jaren N. Ashcraft is a PhD candidate at the University of Arizona's Wyant College of Optical Sciences working with the UA Space Astrophysics Laboratory and Large Optics Fabrication and Testing Group. He received his BS degree in optical engineering from the University of Rochester in 2019 and his MS degree in optical sciences from the University of Arizona in 2022. He is a recipient of a NASA Space Technology Graduate Research Opportunities Award.

Ewan S. Douglas is an assistant professor of astronomy at the University of Arizona and an assistant astronomer at Steward Observatory. He completed a postdoctoral appointment at Massachusetts Institute of Technology Department of Aeronautics and Astronautics. He received his master's and doctoral degrees in astronomy from Boston University and his bachelor's degree in physics from Tufts University. He specializes in space telescopes and instrumentation for high-contrast imaging of debris disks and extrasolar planets.

Daewook Kim is a faculty of optical sciences and astronomy at the University of Arizona. His research area covers precision optical engineering, optics fabrication, and freeform metrology including interferometry and deflectometry. He is the chair of the Optical Manufacturing and Testing (SPIE) and Optical Fabrication and Testing (Optica) conferences. He has served as an associate editor for the *Optics Express*. He is a senior member of Optica and SPIE Fellow.

A. J. Eldorado Riggs is an optical engineer at the Jet Propulsion Laboratory. His research focuses on the high-contrast imaging of exoplanets, in particular mask optimization and wavefront sensing and control for the Coronagraph Instrument on the Nancy Grace Roman Space Telescope. He received his BS degree in physics and mechanical engineering from Yale University in 2011 and his PhD in mechanical and aerospace engineering from Princeton University in 2016.