

# Simulating speckle fields in deep turbulence via wave optics: Angular spectrum method versus sinc-basis propagation

Matthias T. Banet<sup>a</sup>, Kevin Luna<sup>a</sup>, and James R. Fienup<sup>b</sup>

<sup>a</sup>Directed Energy Directorate, Air Force Research Laboratory, Kirtland Air Force Base, New Mexico 87117, USA

<sup>b</sup>The Institute of Optics, University of Rochester, Rochester, New York 14627, USA

## ABSTRACT

Wave optics simulations of speckle fields can be particularly prone to aliasing when using the angular spectrum propagator. We examined several methods of mitigating aliasing, which include larger guard bands, output plane windowing, absorbing boundaries, and an alternative propagator known as the sinc-basis propagator. We compared the angular spectrum propagator to the sinc-basis propagator and found that, while absorbing boundaries greatly assisted the angular spectrum propagator, the sinc-basis propagator always achieved lower root-mean-squared errors for a given array size due to the non-periodicity of the sinc basis. We examined the computation time as a function of the number of pixels and the root-mean-squared error associated with each of the propagators. Direct comparisons on same array size configurations primarily indicated that the relative wall clock time between the two methods depended highly on the core count of the machine. For all machines tested at the same pixel number, the sinc-basis propagator was generally faster up to a machine dependent pixel threshold, after which the angular spectrum propagator was faster. For machines with more parallelization, this threshold was higher and the speed-up of the sinc-basis propagator relative to the angular spectrum method was larger. It was found that the sinc-basis propagator usually has comparable to shorter computation times than the angular spectrum method to achieve the same threshold error in simulations on the computers tested.

**Keywords:** wave-optics simulations, rough-surface scattering, speckle, deep turbulence, aliasing, angular spectrum method, sinc basis

## 1. INTRODUCTION

Wave optics simulations that involve light reflecting off of surfaces that are rough compared to the wavelength of light—hereafter referred to as rough surface scattering—require accurate representations of the scattered complex-valued optical field along the propagation volume. We focus on long-range imaging of coherent speckle fields in this paper, for which we desire the field that scatters off of a distant object into a collecting aperture after propagating through atmospheric turbulence. In physical reality, coherent light (e.g., quasi-monochromatic light or laser light) will scatter into  $2\pi$  steradians upon reflecting off of a rough object. For long-range imaging, however, we only care about the scattered wave components that actually reach the aperture and its surrounding area, which typically have an angular subtense of  $< 1$  mrad. That said, wave optics simulations of long-range coherent imaging typically require some method to deal with the high spatial-frequency components of the field that reach the boundary of the propagation volume of interest and wrap around causing aliasing, which must be avoided to ensure high physical accuracy.

Rough surface scattering of coherent light results in the phenomenon known as speckle,<sup>1</sup> and propagation of speckle fields is pertinent to: wavefront sensing and adaptive optics,<sup>2,3</sup> image reconstruction,<sup>4,5</sup> and 2D and 3D coherent imaging.<sup>6-9</sup> Accurate modeling of these systems is essential to gauge the performance of real-world systems. For wave-optics simulations that include atmospheric turbulence, the paraxial angular spectrum (AS)

---

Further author information: (Send correspondence to M.T.B.)

M.T.B.: E-mail: AFRL.RDL.OrgBox@us.af.mil

propagation method has been used most commonly. This method is a transfer-function representation of the Fresnel diffraction integral (which is itself a convolution) given by

$$U_2(x_2, y_2) = \mathcal{F}_{f_x, f_y}^{-1} \left\{ \exp[-i\pi\lambda z_0(f_x^2 + f_y^2)] \tilde{U}_1(f_x, f_y) \right\} \quad , \quad (1)$$

where  $\tilde{U}_1(f_x, f_y)$  is the 2D Fourier transform of the input field  $U_1(x_1, y_1)$ ,  $\lambda$  is the wavelength,  $z_0$  is the propagation distance,  $\mathcal{F}_{f_x, f_y}^{-1}$  denotes an inverse Fourier transform, and  $U_2(x_2, y_2)$  is the output field. The coordinates  $(x_1, y_1)$  and  $(x_2, y_2)$  correspond to the input and output plane transverse coordinates, respectively, and the coordinates  $(f_x, f_y)$  are the spatial frequency coordinates. Although reflection off of rough surfaces results in scattering at large angles in physical reality, the paraxial approximation still holds here as the angular subtenses of the collecting aperture when viewed from the object (and the object when viewed from the collecting aperture) are small. This also means that the paraxial approximation is valid even if the Fresnel numbers associated with the propagations considered are large (e.g.,  $> 50$ ). Furthermore, atmospheric turbulence ordinarily bends light only over small angles, which justifies the use of the paraxial approximation.

To model a turbulent atmosphere, we employ the split-step beam propagation method,<sup>10</sup> for which we simulate turbulence within the propagation volume as a series of discrete phase screens that each represent the integrated phase aberrations within some slab in the volume. In the split-step beam propagation method, one propagates from screen-to-screen through vacuum while applying a phase to the field at each of the screen locations, which we refer to as intermediate planes, via a multiplicative phasor. Often the AS method is used for the propagations between screens.

The AS method expresses the input field in a discrete Fourier basis, which inherently exists in a 2D wrap-around space in which aliasing can occur. For speckle fields, the innate scattering angles coupled with the inclusion of distributed-volume, or deep, turbulence means that a large guard band (or zero padding) is typically required around the object and collecting aperture in order to avoid aliasing. For some scenarios, this can greatly increase the number of required simulation grid points and, hence, increase simulation runtimes.

Several methods exist to mitigate or prevent aliasing when using the AS method, including guard bands that scale with turbulence strength,<sup>10,11</sup> output plane windowing,<sup>12</sup> and absorbing boundaries.<sup>10</sup> One alternative to the AS method is the sinc-basis propagator.<sup>13,14</sup> This method expresses the optical field in a basis of weighted sinc functions, where each sinc function is a decreasing function in the transverse directions, which is highly useful in avoiding aliasing of speckle fields altogether. In contrast, the Fourier basis functions of complex exponentials do not decay in the transverse directions. In this paper we discuss a number of ways to mitigate aliasing during the propagation calculations including the performance of the AS and sinc-basis propagation methods.

This paper lays out the simulation of speckle fields in deep turbulence via wave optics simulations in a tutorial-esque fashion. Section 2 discusses the simulation frame work and setup that we use for the remainder of the paper, Sec. 3 covers some of the methods used to mitigate or avoid aliasing altogether, including the sinc method, Sec. 4 covers results comparing the AS method to the sinc-basis method, and Sec. 5 concludes the paper. Appendix A includes some extra figures of the profile view of the propagation volume for several of the cases examined in Sec. 4.

## 2. SIMULATION SETUP

### 2.1 Vacuum propagation

In this section we describe the methodology behind all of the wave-optics simulations shown here, which is heavily inspired by Schmidt.<sup>10</sup> We begin by defining regions of interest in the input and output planes with 1D widths of  $R_1$  and  $R_2$ , respectively. These could be regions defined, for example, by an extended object and an aperture, respectively. We also define the physical array side-length of the input and output planes with 1D widths of  $L_1$  and  $L_2$ , respectively. In general, we require that  $L_1 \geq R_1$  and  $L_2 \geq R_2$ . We also define uniform pixel spacings in the input and output planes,  $\delta_1$  and  $\delta_2$ , respectively, and choose all planes (including the input and output planes) within the propagation volume to be  $N_p \times N_p$  square arrays (although this is not strictly required in general).

In wave optics simulations, there is a maximum angular deviation of light which can emanate from a single point. For rough-surface scattering, which we are primarily concerned with in this paper, light scatters at large angles in physical reality; however, in wave optics simulations, the maximum computed scattering angle is given by  $\theta_{max} = \lambda/\delta_1$ .<sup>10</sup> Figure 1 shows the profile view of the propagation volume. In order to ensure physical accuracy in the rough-surface scattering model, we require that each point in the input plane region of interest completely fill the region of interest in the output plane with scattered light. This is the case in Fig. 1, where the scattered light from the edges of the region of interest in the input plane overlap and just barely fill the output plane region of interest. The area outside of the region of interest in the output plane is physically inaccurate in this example. Moving forward, we assume that  $R_1 = R_2 = R$ ,  $L_1 = L_2 = L$ ,  $\delta_1 = \delta_2 = \delta$  for simplicity.

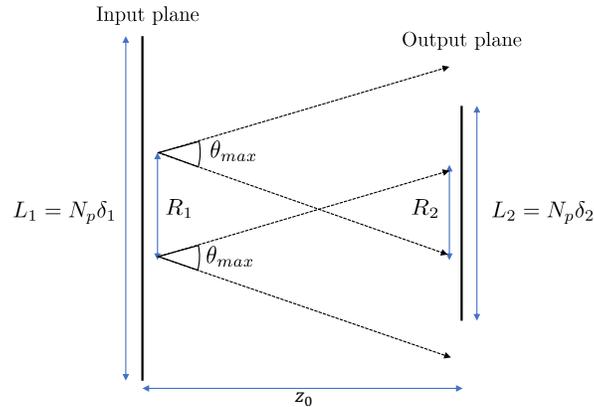


Figure 1. Diagram showing the formalism used for the propagation of speckle fields in this paper.

Figure 1 shows how knowledge of  $\theta_{max}$  can be used to ensure physical accuracy for the speckle field within the region of interest in the output plane, namely, that each point in the input plane region of interest fills or overfills the output plane region of interest with scattered light. Figure 1 does not speak the possibility of aliasing in the propagation volume, however. When the input and output planes have equivalent sampling (i.e.,  $R_1 = R_2$  and  $L_1 = L_2$ ), the propagation volume is defined by a square prism with the input and output planes at the ends of the prism. For the AS method, any light that reaches the boundary of this prism will wrap around to the opposite side of the propagation volume and alias, which is non-physical. This is shown in Fig. 2, which displays profile views (i.e., a projection of the 3D irradiance into the  $x$  direction) of an example propagation volume with  $L = 0.3$  m and  $z_0 = 300$  m for two different input fields. In Fig. 2(a), light scatters from a single on-axis point and completely fills the output plane such that  $\theta_{max} = L/z_0$  with the small-angle approximation.

This condition where a single on-axis scatterer will just barely fill the output plane is met when  $N_p = L^2/(\lambda z)$ , which is referred to as critical sampling of the AS transfer function.<sup>15</sup> In Fig. 2(b), the source consists of one on-axis scatterer (red) and two off-axis scatterers (blue and green). Note that the coloring of the sources here is just for illustration purposes. In this example, we see that the blue and green scatterers are located at the edges of the region of interest in the input plane, and that each scatterer just barely fills the region of interest in the output plane, which satisfies the physical requirements set previously in the description of Fig. 1. Note, however, that the light from the blue and green scatterers reaches the edges of the propagation volume and aliases, resulting in aliased light outside of the region of interest in the output plane. This is acceptable, as we will crop out the central region of interest in the output plane before any further processing. This condition where light is aliased outside of the region of interest in the output plane but physically valid inside the region of interest can be met when  $L_1 = 2R_1$  and  $L_2 = 2R_2$  (i.e., each region of interest is zero-padded by a factor of 2), and we employ it for modeling rough-surface scattering from extended objects. The framework described above (propagation through vacuum) adheres to the AS sampling requirements laid out in Chapter 7 of Ref. 10.

## 2.2 Propagation through turbulence

In the simulations in this study we employed the split-step beam propagation method and used five Kolmogorov phase screens, where we placed the screens at the centers of five equally-spaced slabs of the propagation volume.

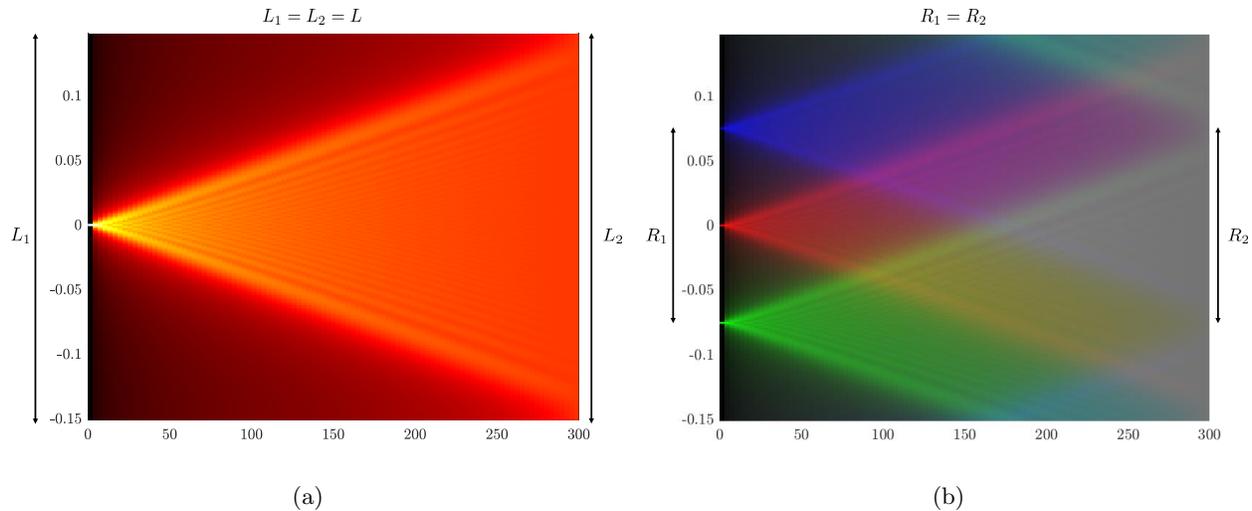


Figure 2. Profile views of the propagation volume showing (a) an on-axis point scatterer and (b) three point scatterers propagating from the input plane to the output plane. The diagrams here satisfy  $N_p = L^2/(\lambda z_0)$ .

We generated the phase screens via the method laid out by Dainty and Lane<sup>16</sup> with added subharmonics. In simulating uniform, distributed-volume turbulence, we found that decreasing the sample spacing for the intermediate planes in the split-step propagation process yielded more physically accurate results. One could accomplish this by varying the sample spacing as a function of propagation distance via the scalable angular spectrum (SAS) method.<sup>10,11</sup> However, we accomplish this here by decreasing the sample spacing for all planes by a factor of 2 and thereby increasing  $N_p$  by a factor of 2 while holding all other variables constant. Decreasing the sample spacing,  $\delta$ , by a factor of 2 increases the maximum scattering angle,  $\theta_{max}$ , by a factor of 2, which could potentially give rise to severe aliasing by the same arguments made with respect to Fig. 2. This change also violates two of the sampling conditions in Chapter 7 of Ref. 10 (conditions 2 and 4 on page 127). However, aliasing can be prevented in this case, as is described below.

To prevent severe aliasing, we define input fields at the lower grid resolution [i.e., when  $N_p = L^2/(\lambda z)$ ] and then upsample the fields prior to propagation to the finer grid resolution [i.e., when  $N_p = 2L^2/(\lambda z)$ ] via the chirp-Z transform.<sup>17</sup> Upsampling the input fields this way preserves their space-bandwidth product, and prevents any extra aliasing due to the finer grid resolution. In effect, modeling rough-surface scatterers this way preserves the qualities of Fig. 2(a): when the input and output plane regions of interest are zero-padded by a factor of 2, aliasing begins midway through the propagation path, but the output plane region of interest remains free of any aliasing for vacuum propagation. We used the process outlined here to simulate all propagations of speckle fields through both vacuum and turbulence.

### 3. ALIASING MITIGATION METHODS

With the description of simulating speckle fields in deep turbulence complete, we now describe some aliasing mitigation methods. While aliasing can obviously occur for speckle fields in vacuum—depending on the amount of zero-padding in the input plane—the techniques laid out here were primarily developed to avoid turbulence-induced aliasing. In deep turbulence, phase screens located along the propagation path can deviate rays and cause aliasing in the output plane region of interest, which might not have occurred in the vacuum case.

#### 3.1 Guard bands based on turbulence strength

One option for avoiding aliasing is to expand the regions of interest in the input and output planes while keeping the source field consistent. This method is inspired by Schmidt<sup>10</sup> as well. Imagine a case where one wishes to propagate speckle field reflecting off of a rough, square plate of width 0.3 m (the input plane) to a circular aperture of diameter 0.3 m (the output plane).

In the vacuum case, we would define  $R_1$  and  $R_2$  to be the widths of the rough, square plate and the aperture, respectively, and pad the input and output fields by a factor of 2 such that  $L = L_1 = L_2 = 0.6$  m. This methodology is based on the geometric arguments laid out with respect to Fig. 2, and it will ensure that aliasing is largely avoided in the output (aperture plane) region of interest. However, these arguments do not account for the field spreading due to turbulence along the propagation path after reflection off of the rough surface.

If the region of interest in the output plane is padded by a factor of 2 (as described above in Sec. 2.1), spreading due to turbulence can cause aliasing to appear at the edges of the region of interest in the output plane. To prevent this, one can define a width,  $R'_2$ , of the effective region of interest in the output plane as

$$R'_2 = R_2 + c_1 \left( \frac{\lambda z_0}{r_{0,1 \rightarrow 2}} \right) , \quad (2)$$

where  $r_{0,1 \rightarrow 2}$  is the Fried parameter<sup>18</sup> calculated by integrating the weighted index-of-refraction structure parameter,  $C_n^2$ , profile from the input plane to the output plane, and  $c_1$  is a user-defined constant. The term in the parentheses in Eq. (2) accounts for spreading due to turbulence. Schmidt<sup>10</sup> defines a similar equation in his analysis (Chapter 9), which recommends  $c_1 \geq 2$ . We use  $c_1 = 2$  for the simulations in this paper that include deep turbulence.

Equation (2) accounts for light spreading upon propagation from the input to output plane, but such spreading could also occur if back-propagating from the output to input plane (e.g., if reconstructing a coherent image of the input plane object after cropping the field in the output plane with an aperture). To account for this, we define a similar width,  $R'_1$ , of the effective region of interest in the input plane via

$$R'_1 = R_1 + c_2 \left( \frac{\lambda z_0}{r_{0,2 \rightarrow 1}} \right) , \quad (3)$$

where  $r_{0,2 \rightarrow 1}$  is the Fried parameter<sup>18</sup> calculated by integrating the weighted  $C_n^2$  profile from the output plane to the input plane, and  $c_2$  is a user-defined constant. Note that, upon examining Eqs. (2) and (3), even if  $R_1 = R_2$ ,  $R'_1$  and  $R'_2$  can differ depending on the  $C_n^2$  profile. Here, for simplicity, we simulate uniform  $C_n^2$  profiles and enforce  $c_1 = c_2$  such that  $R'_1 = R'_2$ . Note, using effective regions of interest that are larger than the original will increase the number of grid points,  $N_p$ , in the simulation as well as decrease the grid sample spacing. For example, consider the case where  $R_1 = R_2 = 0.3$  m,  $\lambda = 1.5$   $\mu$ m,  $z_0 = 2$  km,  $r_{0,1 \rightarrow 2} = r_{0,2 \rightarrow 1} = 2$  cm, and  $c_1 = c_2 = 2$ . In this case  $R'_2 = R'_1 = 0.6$  m, and the increased widths of the effective regions of interest cause  $N_p$  to quadruple when compared to the case where  $R'_2 = R'_1 = 0.3$  m. This is because (in accordance with Sec. 2.2) we enforce that  $L = 2R'_2 = 2R'_1 = 1.2$  m and  $N_p = 2L^2/(\lambda z_0)$ . Recall that the factor of 2 in the equation for  $N_p$  is included to ensure proper sampling of the intermediate planes along the propagation volume.<sup>11</sup> As a result, the grid sampling halves when compared to the case where  $R'_2 = R'_1 = 0.3$  m.

### 3.2 Output plane windowing

Output plane windowing (also referred to as receiver plane windowing/filtering<sup>12</sup> or the super aperture method<sup>19</sup>) effectively reduces the bandwidth of the source field in the input plane such that the scattering angle from each pixel is less than  $\theta_{max}$ . Recall that when simulating deep turbulence (and in all of the subsequent simulations), we upsample the source fields from a coarser grid, such that  $\theta_{max} = \lambda/(2\delta)$ . Output plane windowing limits the scattering angle of the source field even further by generating a filtered version of the original source field. The purpose of this filtering is to reduce turbulence-induced aliasing. We generate this filtered source field according to the following steps:

1. Propagate the original source field through vacuum to the output plane.
2. Apply a window to the output field that is ideally unity (or near unity) over the output plane region of interest and gradually tapers off near the edge of the propagation volume.
3. Propagate back to the input plane.

The goal of step 2 above is to limit the bandwidth of the input plane somewhat in an effort to reduce aliasing. One must take care to ensure that the window in the output plane has an effective support that is larger than the output plane region of interest, but smaller than the total output plane array size. Figure 3 shows some profile views of light scattering off of a rough square plate. For context, in these figures, the propagation axis (the  $z$ -axis) is orientated horizontally, and one of the transverse axes (the  $y$  axis) is oriented vertically. These profile views of the light display a projection of the irradiance of the light over the  $x$ -axis (into the page) within the propagation volume. Here,  $R_1 = R_2 = 0.3$  m,  $\lambda = 1.5$   $\mu$ m,  $z_0 = 2$  km,  $r_{0,1 \rightarrow 2} = r_{0,2 \rightarrow 1} = 2$  cm, and  $c_1 = c_2 = 0$  (i.e.,  $R'_1 = R_1$  and  $R'_2 = R_2$ ). All of the profiles in Fig. 3 were generated with 5 phase screens placed along the propagation path. Note that projecting the irradiance along the  $x$ -axis effectively averages some of the speckle noise in the propagation volume, resulting in the smooth irradiance distributions in Fig. 3. Had we displayed a slice of the propagation volume instead, the speckle contrast would be much greater.

Figure 3(a) shows the profile view of the propagation volume with no attempt to mitigate aliasing. In vacuum, the output plane region of interest (which is padded by a factor of 2), would be largely free of aliasing. However, in this case  $R_2/r_{0,1 \rightarrow 2} = 15$ , so the presence of turbulence induces aliasing into the output plane region of interest. One can observe the aliasing past  $z = 1000$  m as rays of light wrap around from the bottom of the propagation volume and appear at the top, and vice-versa. Figure 3(b) shows the profile view of the propagation volume after output plane windowing is applied (resulting in a source field with a reduced bandwidth). In this case, a Tukey window was applied in the output plane to limit the bandwidth of the source field. The uniform region of the window took up 60% of the output plane, and the tapered region on either side of the uniform region took up 10% of the output plane. The aliasing is obviously lessened in this case due to the reduced spread of the rays from the rough object. However, one can observe that the rays with the largest scattering angle magnitudes are still aliasing. To avoid this, one could make the output plane window smaller; however, it is good practice to make the output plane window larger than the output plane region of interest to allow for rays that would have missed the region of interest in the vacuum case to potentially enter the region of interest in the turbulence case.

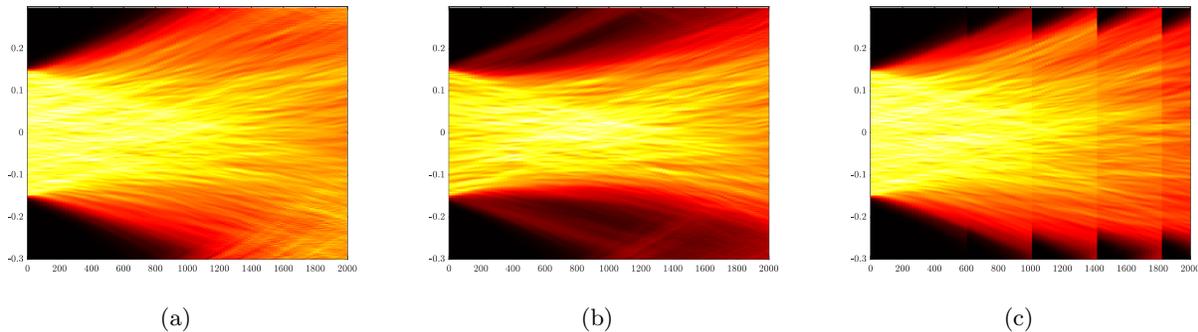


Figure 3. Profile views of the propagation volume of light reflecting off of an optically-rough, square plate through deep turbulence (5 phase screens). Here we see (a) the base propagation with no aliasing mitigation employed, (b) output plane windowing employed, and (c) absorbing boundaries employed. The propagations here set  $c_1 = c_2 = 0$  and satisfy  $N_p = 2L^2/(\lambda z_0)$  in order to properly sample the intermediate planes in the propagation volume.

In practice, we recommend using guard bands—as described in Sec. 3.1—that scale with turbulence in conjunction with output plane windowing if one desires to use it to avoid aliasing with the AS method. In general, results can vary depending on the size and profile of the window applied in the output plane. As a side note, output plane windowing is also helping in avoiding sampling errors in propagated fields that have propagated through deep turbulence (an issue that is separate from aliasing). For example, Sec. 2.2 recommended increasing  $N_p$  to  $2L^2/(\lambda z_0)$  to avoid such errors when propagating through turbulence, but we have found that, in practice, one can utilize fewer pixels across the propagation arrays [i.e.,  $N_p < 2L^2/(\lambda z_0)$ ] if output plane windowing is used. This is essentially due to the fact that output plane windowing reduces the bandwidth of the source field, resulting in fewer required samples across the intermediate planes in the propagation volume. One could low-pass filter the fields at each intermediate plane to achieve a similar effect, and we have also demonstrated this in simulation.

### 3.3 Absorbing boundaries

Absorbing boundaries are an aliasing mitigation technique described by Schmidt<sup>10</sup> as well. This is a straightforward method where one applies a tapered absorbing window to the field at each intermediate plane in a split-step propagation. This has the effect of reducing light that has the potential of aliasing. Although this method was developed to prevent turbulence-induced aliasing, it is a useful technique for avoiding aliasing of speckle fields that propagate through vacuum as well. Figure 3(c) shows a profile view of the propagation volume with absorbing boundaries at each of the 5 intermediate planes. For the absorbing boundary profiles, we used Tukey windows where the uniform region of the window took up 75% of the intermediate plane, and the tapered region on either side of the uniform region took up 10% of the intermediate plane. As was the case with output plane windowing, results may vary depending on the number of intermediate planes where absorbing boundaries are applied and the profile of each absorbing window. Usually absorbing boundaries are only used where phase screens are located along the propagation path, but one could think of pathological cases where one might desire absorbing boundaries at locations where there is no phase screen (e.g., when simulating speckle fields in vacuum using the AS method).

One consequence of using absorbing boundaries is the potential for light to diffract off of the absorbing boundaries themselves into the region of interest in the output plane. If the taper from the uniform region of the absorbing window to the edge of the boundary is smooth enough, this diffracted light will be negligible. However, there are cases where the effects of diffraction are noticeable. In one example, we examined the sidelobes in images formed with a circular aperture and tapered square Tukey absorbing boundaries placed along the propagation path. The jinc sidelobes due to diffraction off of the circular aperture were quite noticeable, but secondary sinc-like sidelobes of lesser magnitude were evident as well, due to the diffraction off of the absorbing boundaries, which is non-physical. Despite that, absorbing boundaries are a valid method for the reduction of aliasing with proven success.

### 3.4 Sinc-basis propagation

One option to completely prevent aliasing is to use a propagator that employs a non-periodic basis, as opposed to the periodic Fourier basis set used in the AS method. We explore the sinc-basis propagator described by Cubillos and Jimenez,<sup>13,14</sup> and we describe the basic theory behind the use of the propagator for the Fresnel diffraction integral.

To start, we assume that the field in the input plane,  $U_1(x_1, y_1)$ , is band-limited; i.e., its spectrum completely fits into a square space of width  $2W_0$  (so in 1D the support of the spectrum fits within the interval  $[-W_0, W_0]$ ). According to the Whittaker-Shannon interpolation theorem,<sup>20</sup> one can express  $U_1(x_1, y_1)$  (as long as it is band limited) in terms of an infinite sum of sinc functions, i.e.,

$$U_1(x_1, y_1) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} U_{1,mn} \text{sinc}\left(\frac{x_1}{\delta} - m\right) \text{sinc}\left(\frac{y_1}{\delta} - n\right) \quad , \quad (4)$$

where  $U_{1,mn}$  is the sampled version of  $U_1(x_1, y_1)$  and  $m$  and  $n$  are integers that index the sample locations. Defining  $\delta = 1/(2W)$ , then Eq. 4 requires that  $W > W_0$  (i.e., the input function must be at least Nyquist sampled).

The Fresnel diffraction integral, in convolution form, is given by

$$U_2(x_2, y_2) = (h * U_1)(x_2, y_2) = \frac{e^{ikz_0}}{i\lambda z_0} \iint_{-\infty}^{\infty} \exp\left\{\frac{i\pi}{\lambda z_0} \left[(x_2 - x_1)^2 + (y_2 - y_1)^2\right]\right\} U_1(x_1, y_1) dx_1 dy_1 \quad , \quad (5)$$

where  $h(x_1, y_1) = e^{ikz_0}/(i\lambda z_0) \exp\{[i\pi/(\lambda z_0)](x_1^2 + y_1^2)\}$  is the Fresnel propagation kernel and  $*$  denotes a 2D convolution. If we define the solution of the Fresnel diffraction integral for a single, on-axis sinc function input, i.e.,

$$\Phi(x_2, y_2) = \left[ h(x_1, y_1) * \text{sinc}\left(\frac{x_1}{\delta}\right) \text{sinc}\left(\frac{y_1}{\delta}\right) \right] (x_2, y_2) \quad , \quad (6)$$

then after plugging Eq. (4) into Eq. (5)—and because the Fresnel diffraction integral is linear, shift-invariant—we can express  $U_2(x_2, y_2)$  as an infinite summation of weighted, shifted versions of  $\Phi(x_2, y_2)$ :

$$U_2(x_2, y_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} U_{1,mn} \Phi(x_2 - m\delta, y_2 - n\delta) \quad . \quad (7)$$

All that is left to determine is the solution to  $\Phi(x_2, y_2)$ . We accomplish this by expressing  $\Phi(x_2, y_2)$  in the transfer function representation of the Fresnel diffraction integral, i.e.,

$$\Phi(x_2, y_2) = \delta^2 \mathcal{F}_{f_x, f_y}^{-1} \left\{ \tilde{h}(f_x, f_y) \text{rect} \left( \frac{f_x}{2W} \right) \text{rect} \left( \frac{f_y}{2W} \right) \right\} \quad , \quad (8)$$

where  $\tilde{h}(f_x, f_y) = e^{ikz_0} \exp[-i\pi\lambda z_0(f_x^2 + f_y^2)]$  is the free-space optical transfer function. Expanding Eq. 8 yields

$$\Phi(x_2, y_2) = e^{ikz_0} \delta^2 \int_{-W}^W \int_{-W}^W \exp[-i\pi\lambda z_0(f_x^2 + f_y^2)] \exp[-i2\pi(f_x x_2 + f_y y_2)] df_x df_y \quad (9)$$

after leveraging the rect functions to change the bounds of the integrals. Following along with Cubillos and Jimenez,<sup>13,14</sup> we can express Eq. (9) in terms of the Fresnel sine and cosine integrals, which are given by

$$S(x) = \int_0^x \sin(\mu^2) d\mu; \quad C(x) = \int_0^x \cos(\mu^2) d\mu \quad . \quad (10)$$

We can now write

$$\Phi(x_2, y_2) = e^{ikz_0} \phi(x_2) \phi(y_2) \quad , \quad (11)$$

where

$$\phi(x_2) = \frac{\delta}{\pi} \sqrt{\frac{k}{2z}} \exp \left( i \frac{k}{2z} x_2^2 \right) \{ C(\mu_+) - C(\mu_-) - i [S(\mu_+) - S(\mu_-)] \} \quad , \quad (12)$$

and

$$\mu_{\pm} = \pm \pi \sqrt{\frac{2z}{k}} W - \sqrt{\frac{k}{2z}} x_2 \quad (13)$$

and  $\phi(y_2)$  is defined similarly but with all arguments of  $x_2$  swapped with  $y_2$ . Thus, for the Fresnel diffraction integral, one can compute an analytic solution for  $\Phi(x_2, y_2)$  for a given propagation geometry.

In practice, all that is needed to evaluate Eq. (7) is to truncate the infinite summation such that it encompasses the support of the sampled input field,  $U_{1,mn}$ . Assuming the infinite summation is truncated such that it spans from  $-N_p/2$  to  $N_p/2$  in both dimensions (assuming  $N_p$  is even),  $U_2(x_2, y_2)$  becomes a discrete convolution, namely

$$U_2(x_2, y_2) = \sum_{m=-N_p/2}^{N_p} \sum_{n=-N_p/2}^{N_p} U_{1,mn} \Phi(x_2 - m\delta, y_2 - n\delta) \quad . \quad (14)$$

One can solve Eq. (14) via circular convolution after padding  $U_{1,mn}$  and  $\Phi(x_2, y_2)$  by a factor of 2 in each dimension. Cubillos and Jimenez<sup>13,14</sup> also express  $U_2(x_2, y_2)$  in terms of the product of three  $N_p \times N_p$  matrices [Eq. (66) in Ref. 13 and Eq. (29) in Ref. 14], which is the method we employ for most of the remaining simulations in this paper, save for a brief comparison on the computation times between the two methods in Sec. 4.

## 4. RESULTS

We now examine qualitative and quantitative results comparing the AS method and the sinc-basis propagation method. We examine two base scenarios: (1) propagation through deep turbulence, and (2) propagation through deep turbulence with absorbing boundaries at each intermediate plane. For each scenario we used  $z_0 = 2$  km,  $\lambda = 1.5$   $\mu\text{m}$ ,  $R_1 = R_2 = R = 0.3$  m,  $c_1 = c_2 = 2$  (i.e.,  $R'_1 = 2R_1 = 0.6$  m and  $R'_2 = 2R_2 = 0.6$  m). The use of larger guard bands results in a zero-padding factor of 4 around the input source field. Additionally, deep

turbulence with a uniform  $C_n^2$  profile was simulated via the split-step beam propagation method with 5 phase screens. For each case, the input source field was an optically-rough,  $0.3\text{ m} \times 0.3\text{ m}$  square plate that was uniformly illuminated.

For each of the base scenarios described above, we generated “truth” fields for the two propagators studied here: the AS method and the sinc-basis method. For each propagator, the truth field was the complex field collected in the central  $R_2 \times R_2$  region in the output plane. To compare the two propagators, we reduced the amount of zero padding in the input plane via cropping while keeping the sample spacing the same for the input fields and compared the resulting field in the central  $R_2 \times R_2$  region in the output plane to the previously collected truth field for that propagator. In this way, each propagator was judged relative to its own best case scenario.

We found output plane windowing (see Sec. 3.2) to be effective both at reducing artifacts in the output plane due to sampling errors of the intermediate planes in the split-step beam propagation method<sup>11</sup> and reducing aliasing. Absorbing boundaries (see Sec. 3.3) were more effective at reducing aliasing specifically, especially for larger numbers (greater than 3) of phase screens distributed along the propagation path. For this reason, we included absorbing boundaries in scenario (2) here, but did not include output plane windowing. Both scenarios (1) and (2) make use of increased guard bands based on turbulence strength, described in Sec. 3.1.

We parameterized our study with,  $Q$ , which is defined as the ratio of the propagation volume transverse width,  $L$ , to the width of the input plane region of interest,  $R_1$ . This means that  $Q$  effectively describes the amount of zero padding present around the input plane source field. For both scenarios (1) and (2), we generated truth fields with  $Q = 4$ .

Our error metric was the root-mean-square (RMS) error given by

$$\text{RMS}_{error} = \sqrt{\frac{\sum_{x_2, y_2 \in [-R_2/2, R_2/2]^2} |U_{truth}(x_2, y_2) - U_2(x_2, y_2)|^2}{N^2}}, \quad (15)$$

where  $U_{truth}(x_2, y_2)$  is the truth field for a given propagator and  $N$  was the linear number of pixels across the region of interest in the output plane.

## 4.1 Qualitative results

Each of the subfigures in this section shows the amplitude (left, with colormap gray) and phase (right, with colormap parula) of the cropped  $R_2 \times R_2$  center of the output plane (i.e., the output plane region of interest). This region of interest is where we would potentially truncate the field with an aperture, for example. Note that the fields here have also been collimated by multiplying the output by a thin lens function of the form  $\exp[-i\pi/(\lambda z_0)(x_2^2 + y_2^2)]$  in order to better interpret the phase of each plot.

### 4.1.1 Scenario (1)

For scenario (1) (propagation through deep turbulence with  $Q = 4$  for the truth fields), Fig. 4 shows the complex fields (both amplitude and phase) for (a) the truth case where  $Q = 4$  and (b) the case where  $Q = 2$  for the AS method. Figure 4(a) shows the truth case for a field of speckle and scintillation. When we decreased  $Q$  to 2, as is the case for Fig. 4(b), aliasing corrupted the entire region of interest in the output plane. This case exemplifies how deep turbulence can severely exacerbate aliasing and how even a substantial guard band of  $Q = 2$  does not suffice for the AS method. Figure 5 shows results formed with the sinc-basis propagator. The truth field, shown in Fig. 5(a), looks nearly identical to the results for  $Q = 2$ , shown in Fig. 5(b), which indicates a lack of aliasing.

For scenario (2) (propagation through deep turbulence with absorbing boundaries and with  $Q = 4$  for the truth fields), Fig. 6 shows the complex fields (both amplitude and phase) for (a) the truth case where  $Q = 4$  and (b) the case where  $Q = 2$  for the AS method. For the absorbing boundaries, we used Tukey windows where the uniform region of the window took up 75% of the intermediate plane, and the tapered region on either side of the uniform region took up 10% of the intermediate plane. Figure 6(a) shows a similar speckle and scintillation field to Fig. 4(a) because they were formed from the same realizations of speckle and turbulence. When we decreased

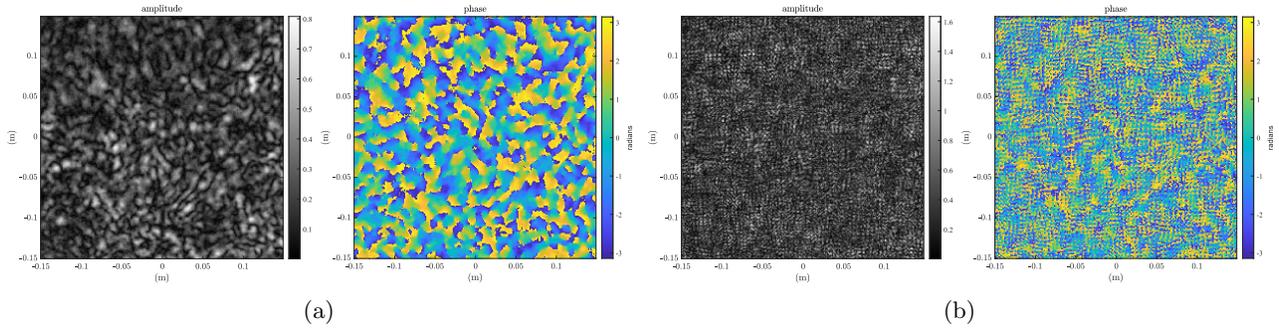


Figure 4. Complex fields in the region of interest of the output plane for scenario (1) and the AS method. Here, (a) displays the truth field where  $Q = 4$  and (b) displays the case when  $Q = 2$ . In each subfigure, the amplitude is on the left and the phase is on the right.

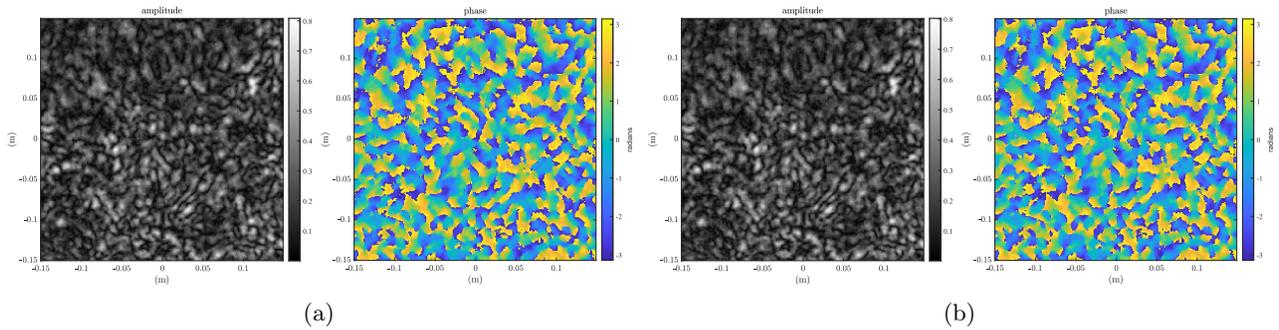


Figure 5. Complex fields in the region of interest of the output plane for scenario (1) and the sinc-basis method. Here, (a) displays the truth field where  $Q = 4$  and (b) displays the case when  $Q = 2$ . In each subfigure, the amplitude is on the left and the phase is on the right.

$Q$  to 2, as is the case for Fig. 6(b), aliasing induced errors in the region of interest in the output plane, but the field is still recognizably similar to the field in Fig. 6(a). Note that the only difference between Figs. 6(b) and 4(b) is the inclusion of absorbing boundaries at the intermediate planes for Fig. 6(b), so it is quite clear that absorbing boundaries can help mitigate aliasing a great deal for speckle fields formed with the AS method. As mentioned in Sec. 3.3, the benefits of absorbing boundaries can vary greatly depending on the distribution and profiles of the absorbing windows themselves.

Figure 7 shows results that are similar to Fig. 5. The truth field, shown in Fig. 7(a), looks nearly identical to the results for  $Q = 2$ , shown in Fig. 7(b), which again indicates a lack of aliasing. There are extremely subtle qualitative differences between Figs. 7(b) and 5(b). For example, there are faint horizontal fringes (that look non-physical) in the amplitude of Fig. 5(b) at approximately (0.04 m, 0.03 m) (the case with no absorbing boundaries). These fringes are not visible at the same location in Fig. 7(b) (the case with absorbing boundaries), which indicates that absorbing boundaries may have some positive effects for the sinc-basis propagator as well.

We theorize that absorbing boundaries assist the sinc-basis propagator when the field at the intermediate planes overfills the propagation grid. Although no aliasing will occur for the sinc-basis propagator, the field at each intermediate plane becomes the new effective source for the subsequent propagation. When this field overfills the propagation grid, it imposes sharp discontinuities at the edges of the propagation grid, which could induce artifacts in the final output plane. Absorbing boundaries ensure that the effective source fields at each intermediate plane have some “wobble room” at the edges of the propagation grid, which, in turn, prevents sharp discontinuities in these fields. Appendix A shows the profile views of the propagation volumes shown here, where one can plainly see cases where the fields propagated via the sinc-basis method overfill the propagation grid, especially for larger  $z$ .

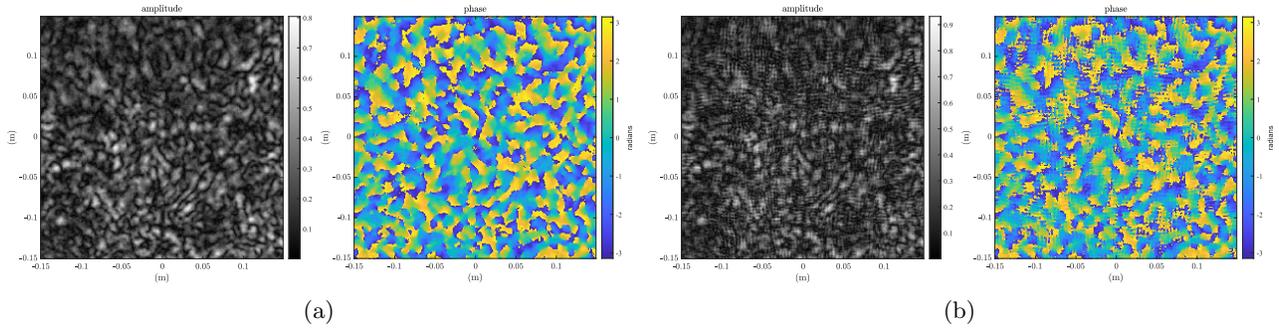


Figure 6. Complex fields in the region of interest of the output plane for scenario (2) and the AS method. Here, (a) displays the truth field where  $Q = 4$  and (b) displays the case when  $Q = 2$ . In each subfigure, the amplitude is on the left and the phase is on the right.

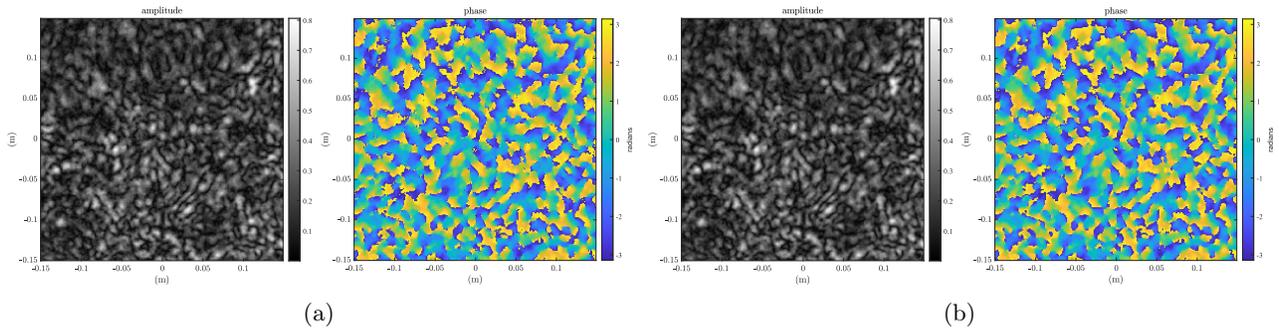


Figure 7. Complex fields in the region of interest of the output plane for scenario (2) and the sinc-basis method. Here, (a) displays the truth field where  $Q = 4$  and (b) displays the case when  $Q = 2$ . In each subfigure, the amplitude is on the left and the phase is on the right.

## 4.2 Quantitative results

Now we report the quantitative results obtained in this study. We begin with Fig. 8, which shows RMS error versus  $Q$  for the three scenarios explored here. The error bars in these plots correspond to  $\pm 1$  standard deviation calculated over 10 independent realizations.

Figure 8(a) shows the results for scenario (1), which corresponded to propagation through deep turbulence with  $Q = 4$  for the truth fields and no absorbing boundaries. Here, the AS method error balloons as  $Q$  decreases due to the introduction of turbulence, which causes more severe aliasing. The sinc-basis method does experience an increase in error as  $Q$  decreases, but it is much more robust to decreasing  $Q$ . The RMS deviation between the truth fields for the AS and sinc-basis methods was 0.0115, on average, for scenario (1).

Figure 8(b) shows the results for scenario (2), which corresponded to propagation through deep turbulence with  $Q = 4$  for the truth fields and with absorbing boundaries located at the intermediate planes. Here we can see the large benefits of including absorbing boundaries at the intermediate planes for the AS method. The error for the AS method still increases as  $Q$  decreases, but does not reach the same heights as Fig. 8(a), where absorbing boundaries were not included. This may prompt one to suggest placing absorbing boundaries along the path for the vacuum case as well. While this would certainly reduce the error for the AS method, it would come at the cost of additional computation time, since the AS method requires two fast Fourier transforms (FFTs) per propagation step. Figure 8(b) does not indicate a substantial change in the RMS error for the sinc-basis method with the inclusion of absorbing boundaries. However, we have noticed small changes qualitatively, described in Sec. 4.1, that suggest that the sinc-basis method could benefit from absorbing boundaries when propagating through multiple phase screens, which warrants further exploration. The RMS deviation between the truth fields for the AS and sinc-basis methods was 0.0020, on average, for scenario (2). The decrease from scenario (1) is likely due to implementation of absorbing boundaries, which reduces the potential for aliasing in the AS method.

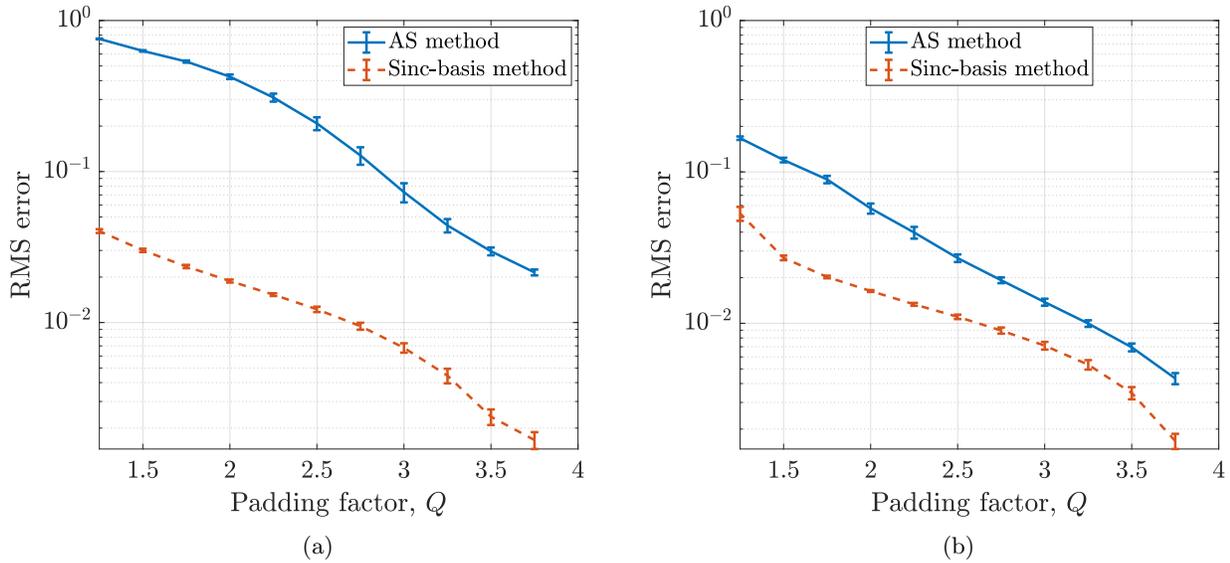


Figure 8. RMS error results for both the AS and sinc-basis methods for (a) scenario (1), and (b) scenario (2). The results are an average of 10 independent realizations of speckle and turbulence and the errorbars correspond to  $\pm 1$  standard deviation. Note that for all array sizes explored here, the sinc-basis method was able to obtain smaller RMS error values than the AS method.

Next, we profiled the run-times of a single propagation step on 3 different computational platforms within the MATLAB environment. To show the fastest possible run-times and to take full advantage of the GPU available to us, we used single precision for all of the computations we timed. In particular, we examined FFT and matrix-based implementations of the AS and sinc-basis methods. For an  $N_p \times N_p$  array, the FFT-based AS method consists of 2 2D FFTs and one element-wise multiplication. The matrix implementation of the AS method replaces each 2D FFT with a triple matrix product of  $N_p \times N_p$  DFT matrices,<sup>21,22</sup> requiring a total of two triple matrix products and one element-wise multiplication. The matrix multiplication form of the sinc-basis method requires a single matrix triple product of  $N_p \times N_p$  matrices where each matrix is a Toeplitz matrix. The FFT-based sinc-basis method corresponds to replacing each matrix multiplication operation with  $N_p$  1D FFTs of size  $2N_p$  through embedding each Toeplitz matrix into a circulant matrix.<sup>23</sup> The three systems we profiled wall clock times on were an AMD EPYC 7402P CPU with 24 cores at 2.8 GHz, a dual Intel Xeon Gold 6152 workstation with a total of 44 cores at 2.1GHz, and a NVIDIA GeForce RTX 3070 Ti GPU that has 6144 CUDA cores at 1.575 GHz.

Figures 9(a)–9(c) show plots of the computational time of each single step propagation as a function of  $N_p$ , and Figs. 9(d)–9(f) plot the computational time required to achieve a given error tolerance. Focusing on Figs. 9(a)–9(c), we observe completely expected results. In the following discussion we omit the DFT implementation of the AS method, as it is the slowest method of all on every platform because it suffers from requiring twice as many consecutive operations relative to the other methods. To understand why these results are expected, it is important to recall two facts. First, matrix multiplication is an embarrassingly parallel algorithm (it can be partitioned without communication between processes), and second, the parallel FFT requires communication between processes. Therefore, for sufficiently small sizes of the matrix relative to the core count of the machine, matrix multiplication can outperform the FFT. Thus, as the core count rises, matrix multiplication operations can profit substantially more than FFT algorithms provided the sizes of the arrays of interest remain fixed.

Next, returning to the results, this is why the sinc-basis matrix multiplication method goes from being the fastest (or equally fastest) method for sizes of  $N_p \leq 512$  on the 24 core machine in Fig. 9(a) to always being the fastest method in Fig. 9(b) on the 44 core machine. A similar trend is observed on our GPU in Fig. 9(c). Sizes of  $N_p < 600$  are substantially accelerated for the sinc-basis matrix multiplication method over the AS FFT method. The flat behavior of the AS FFT method in Fig. 9(c) indicates that the method is maximally

parallelized over the number of cores, and is mainly limited by the core clock speed (due to communication of the algorithm). However, as seen in the CPU cases [i.e., Figs. 9(a)–9(b)], an increase in the number of cores on the GPU will lead to further substantial gains for the sinc-basis matrix multiplication method and for larger  $N_p$ . Since GPUs primarily grow in core count over core frequency, the sinc-basis matrix multiplication method will continue to improve as the core count of new GPU models continue to climb. For CPUs with smaller core counts, the AS FFT and the sinc-basis method implemented with the FFT will generally be comparable to each other [see Figs. 9(a)–9(b) for the largest sizes]. However, the time for operations of a certain array size only tells a portion of story.

Recall that all forms of the sinc method required significantly smaller  $N_p$  to reach a given error threshold [as shown in Fig. 8]. When we define a threshold for tolerable error in the region of interest of the output plane, then the requisite number of grid points while using the sinc-basis propagator is naturally reduced. For example, if RMS errors  $\simeq 0.05$  are tolerable, then we can reduce  $Q$  to 1.2 for the sinc-basis propagator and get an effective speed up over the AS method that is not accounted for in Figs. 9(a)–9(c). Note that these results are specific to these speckle simulations, and speeds ups for the sinc-basis method can be even larger for other types of simulations.<sup>13</sup>

In Figs. 9(e)–9(f) we plot the error for the fastest implementation of each method as a function of the time of each single propagation step. Generally, we see that the sinc-basis method provides an acceleration of at least two to eight times to reach a given error tolerance. The only exception to this trend is larger array sizes on our GPU. The large error bars in this regime would not disappear despite more samples being averaged, indicating that full timing of the split-step beam propagation is needed to accurately gauge performance due to noise from background process scheduling. Furthermore, we remark that as the core count is increased on the GPU, we would expect the red line in Fig. 9(f) to shift to the left of the blue line as discussed earlier. Finally we remark that relative performance of different methods depends on core counts, core clock speeds, and desired error thresholds. Therefore, we recommend that every user should profile their desired methods on their machine with the minimum required number of pixels and minimum precision needed to obtain optimal speeds for their simulations. For example, here we found the sinc method generally sped up execution of the code, but the correct type of implementation for a given size needed to be identified.

## 5. CONCLUSIONS

In conclusion, we have demonstrated methods for propagating speckle fields in deep turbulence using the angular spectrum (AS) method, as well as several aliasing mitigation methods including: guard bands that scale with turbulence, output plane windowing, and absorbing boundaries. Moreover, we implemented an alternative propagator known as the sinc-basis propagator that is free from any aliasing. We demonstrated that both propagators can be implemented with the split-step beam propagation method to model distributed-volume turbulence within a propagation volume.

We qualitatively demonstrated the harmful effects of aliasing for the AS method, which are present even for propagations through vacuum, as well as the fact that the sinc-basis method exhibits no aliasing. We showed that large errors due to aliasing appear for the AS method when the padding factor,  $Q$ , around the input plane source field and propagated field decreases. Absorbing boundaries greatly aid the AS method in reducing aliasing, and we theorize that they provide some small benefits to the sinc-basis method as well. One could employ absorbing boundaries in vacuum propagations with the AS method, but at the cost of increasing the number of propagations per simulation.

For the AS method, one requires  $Q \geq 2$  for vacuum propagations, and even larger  $Q$  for deep turbulence conditions depending on turbulence strength. For example, for the turbulence conditions simulated here, we recommend  $Q \geq 4$  without the use of absorbing boundaries and  $Q \geq 3$  with the use of absorbing boundaries. For the sinc-basis method, one requires  $Q \geq 1$  for vacuum propagations and slightly larger  $Q$  for deep turbulence conditions depending on turbulence strength. For example, for the turbulence conditions simulated here, we recommend  $Q \geq 1.25$  with or without the use of absorbing boundaries.

We timed the AS and sinc-basis propagators implemented with FFTs and matrix multiplications on two CPUs and one GPU. First, we examined computation times as a function of array size. Up to a given critical

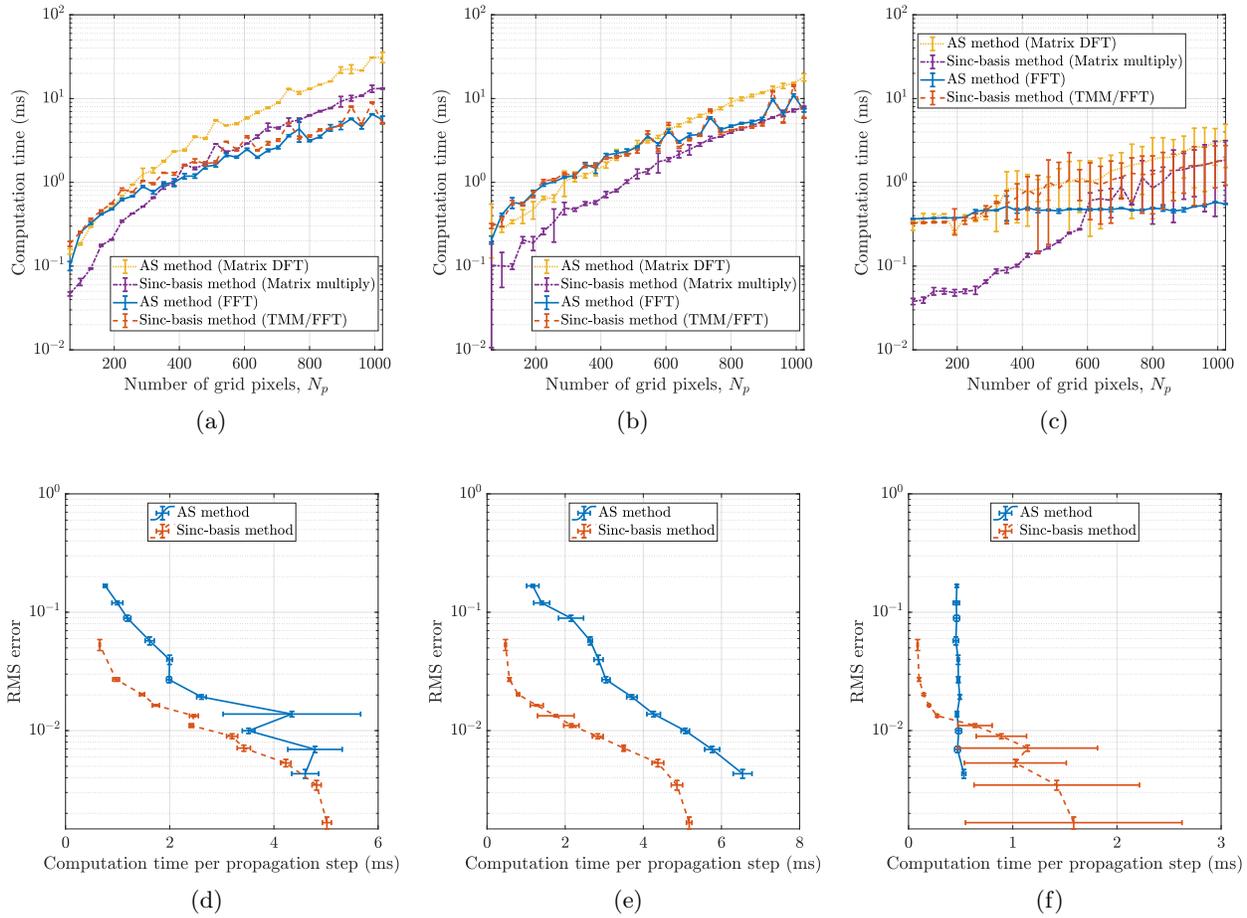


Figure 9. Left column: AMD EPYC 7402P CPU (24 cores at 2.8 GHz). Middle column:  $2 \times$  Intel Xeon Gold 6152 ( $2 \times 22$  cores at 2.1 GHz). Right column: NVIDIA GeForce RTX 3070 Ti (6144 CUDA cores at 1.575 GHz). All cases are with absorbing boundaries. Top row: time versus number of linear pixels for (a) the 24 core CPU, (b) the  $2 \times 22$  core CPU, and (c) the GPU. Bottom row: RMS error versus fastest computation time per propagation step (which varied with  $N_p$ ) of each method for (d) the 24 core CPU, (e) the  $2 \times 22$  core CPU, and (f) the GPU. Each time error bar is the results of 100 realizations (400 for GPU).

pixel count,  $N_p^c$ , the matrix implementation of the sinc-basis method was the fastest method, followed by the FFT-based AS method, and the FFT-based AS method was fastest for arrays larger than  $N_p^c$ . The speed-up and  $N_p^c$  that demarcated where sinc-basis matrix multiplication was fastest depended on the core count of the machine. The higher the core count, the larger the demarcating  $N_p^c$  and the speed-up for the sinc-basis matrix multiplication method. By taking into account the relaxed zero-padding constraints of the sinc-basis method, it was found that, when comparing the AS method and sinc-basis method for RMS errors of  $\simeq 0.01$  and above, the appropriate sinc-basis method implementation had a lower computation time than the AS method on all the machines tested.

## APPENDIX A. PROFILE VIEWS OF THE PROPAGATION VOLUME

This Appendix shows the profile views of the propagation volume for several of the cases simulated in Sec. 4.

### A.1 Scenario (1)

Figure 10 shows the profile views of the propagation volume for scenario (1), generated by the AS method. Here, (a) shows the truth field, which was generated with  $Q = 4$ , and (b) and (c) show the cases where  $Q = 3$  and 2,

respectively. Here, aliasing is due to both the propagation of speckle fields and deep turbulence.

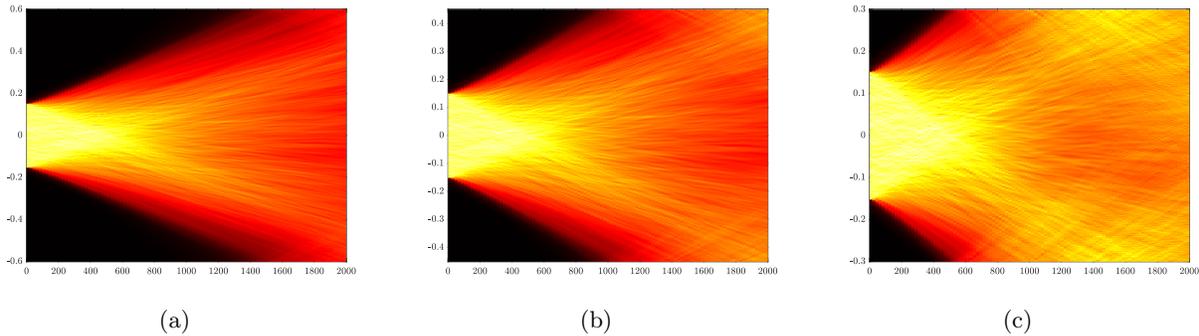


Figure 10. Profile views of the light for scenario (1) and AS propagation where (a) is the truth case and  $Q = 4$ , (b) corresponds to  $Q = 3$ , and (c) corresponds  $Q = 2$ .

Figure 11 shows the profile views of the propagation volume for scenario (1), generated by the sinc-basis method. Here, (a) shows the truth field, which was generated with  $Q = 4$ , and (b) and (c) show the cases where  $Q = 3$  and 2, respectively. Aliasing is not present at all, even though the propagated field obviously fills the propagation grid for  $z > 3z_0/4$ .

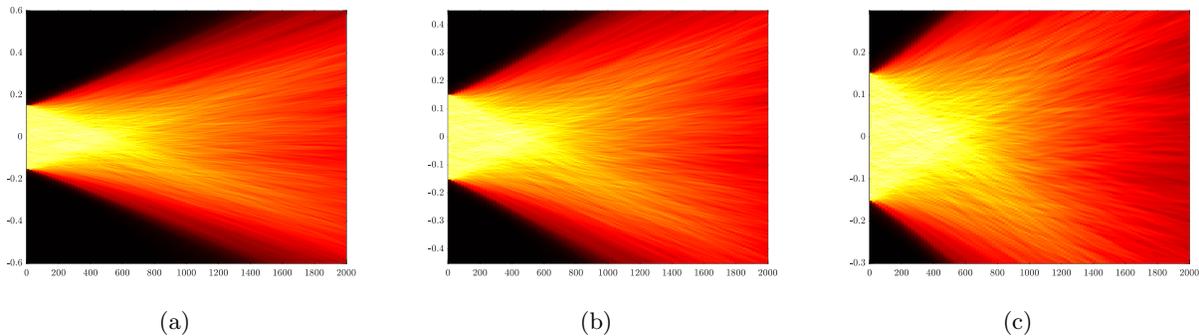


Figure 11. Profile views of the light for scenario (1) and sinc-basis propagation where (a) is the truth case and  $Q = 4$ , (b)  $Q = 3$ , and (c)  $Q = 2$ .

Figure 12 shows the profile views of the propagation volume for scenario (2), generated by the AS method. Here, (a) shows the truth field, which was generated with  $Q = 4$ , and (b) and (c) show the cases where  $Q = 3$  and 2, respectively. Aliasing is due to both the propagation of speckle fields and deep turbulence. The inclusion of absorbing boundaries greatly reduces aliasing along the propagation path.

Figure 13 shows the profile views of the propagation volume for scenario (2), generated by the sinc-basis method. Here, (a) shows the truth field, which was generated with  $Q = 4$ , and (b) and (c) show the cases where  $Q = 3$  and 2, respectively. Aliasing is not present at all, even though the propagated field obviously fills the propagation grid for  $z > 3z_0/4$ . We expect that the inclusion of absorbing boundaries somewhat aids in the sinc-basis propagation, especially after the propagated field fills the propagation grid. Still, no aliasing is present.

## ACKNOWLEDGMENTS

The authors of this paper would like to acknowledge Sam Thurman for many fruitful discussions about propagators and sampling requirements for the split-step beam propagation method. The views expressed are those of the author and do not necessarily reflect the official policy or position of the Department of the Air Force,

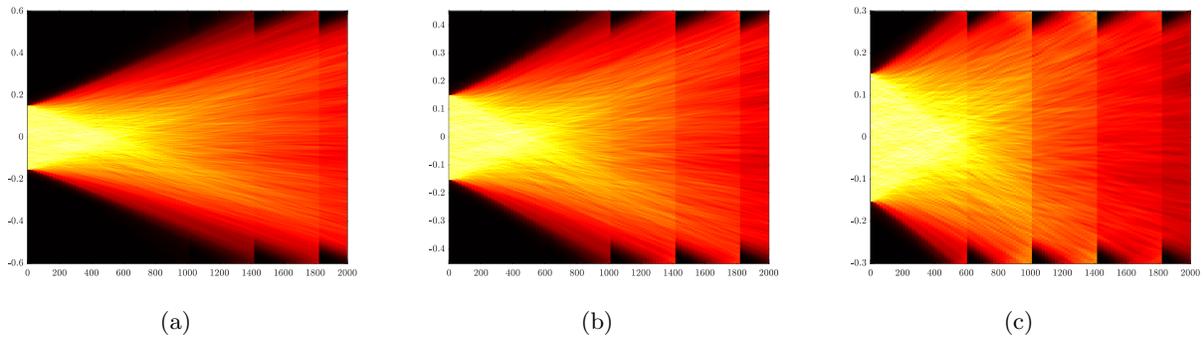


Figure 12. Profile views of the light for scenario (2) and AS propagation where (a) is the truth case and  $Q = 4$ , (b)  $Q = 3$ , and (c)  $Q = 2$ .

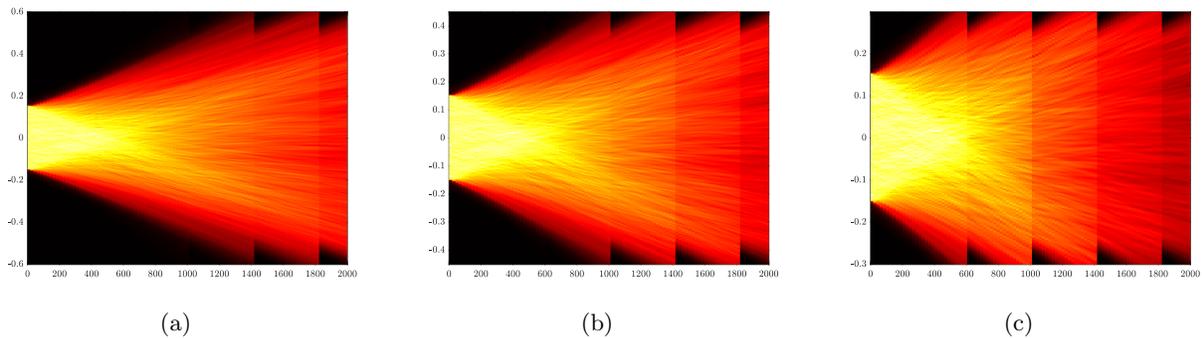


Figure 13. Profile views of the light for scenario (2) and sinc-basis propagation where (a) is the truth case and  $Q = 4$ , (b)  $Q = 3$ , and (c)  $Q = 2$ .

the Department of Defense, or the U.S. Government. The authors declare that there are no conflicts of interest related to this article. Approved for public release; distribution is unlimited. Public Affairs release approval #AFRL-2024-4616.

## REFERENCES

- [1] Goodman, J. W., [*Speckle phenomena in optics: theory and applications*], SPIE Press, 2nd ed. (2020).
- [2] Weyrauch, T. and Vorontsov, M. A., “Atmospheric compensation with a speckle beacon in strong scintillation conditions: directed energy and laser communication applications,” *Appl. Opt.* **44**(30), 6388–6401 (2005).
- [3] Banet, M. T. and Spencer, M. F., “Compensated-beacon adaptive optics using least-squares phase reconstruction,” *Opt. Express* **28**(24), 36902–36914 (2020).
- [4] Paxman, R. G. and Marron, J. C., “Aberration correction of speckled imagery with an image-sharpness criterion,” *Proc. SPIE* **976**, 37–47 (1988).
- [5] Thurman, S. T. and Fienup, J. R., “Correction of anisoplanatic phase errors in digital holography,” *J. Opt. Soc. Am. A* **25**(4), 995–999 (2008).
- [6] Marron, J. C. and Schroeder, K. S., “Three-dimensional lensless imaging using laser frequency diversity,” *Appl. Opt.* **31**(2), 255–262 (1992).
- [7] Shirley, L. G. and Hallerman, G. R., “Nonconventional 3D imaging using wavelength-dependent speckle,” *Lincoln Laboratory Journal* **9**(2), 153–186 (1996).
- [8] Burrell, D. J., Spencer, M. F., Van Zandt, N. R., and Driggers, R. G., “Wave-optics simulation of dynamic speckle: II. in an image plane,” *Appl. Opt.* **60**(25), G77–G90 (2021).
- [9] Banet, M. T. and Fienup, J. R., “Speckle decorrelation effects on motion-compensated, multi-wavelength 3D digital holography: theory and simulations,” *Opt. Eng.* **62**(7), 073103 (2023).

- [10] Schmidt, J. D., [*Numerical simulation of optical wave propagation: With examples in MATLAB*], SPIE Press (2010).
- [11] Thurman, S. T. and DeSantis, Z. D., “Sampling requirements for coherent imaging,” *Pending publication in Proc. SPIE* (2024).
- [12] Brennan is the sole author of WavePlex with correspondence to the following address, T. J. Prime Plexus, 650 N Rose Drive, #439, Placentia, CA, USA 92870.
- [13] Cubillos, M. and Jimenez, E., “Diffraction integral computation using sinc approximation,” *Applied Numerical Mathematics* **178**, 69–83 (2022).
- [14] Cubillos, M. and Jimenez, E., “Numerical simulation of optical propagation using sinc approximation,” *J. Opt. Am. Soc. A* **39**(8), 1403–1413 (2022).
- [15] Voelz, D. G., [*Computational Fourier optics: a MATLAB tutorial*], SPIE Press (2011).
- [16] Lane, R., Glindemann, A., and Dainty, J., “Simulation of a Kolmogorov phase screen,” *Waves in random media* **2**(3), 209 (1992).
- [17] Jurling, A. S., Bergkoetter, M. D., and Fienup, J. R., “Techniques for arbitrary sampling in two-dimensional Fourier transforms,” *J. Opt. Am. Soc. A* **35**(11), 1784–1796 (2018).
- [18] Andrews, L. C. and Phillips, R. L., [*Laser beam propagation through random media*], SPIE Press (2005).
- [19] Coy, S. C. and Praus, R. W., “WaveTrain, A Commercial Quality Tool for Wave Optics Simulation.” Also see WaveTrain documentation and related publications are at <http://www.mza.com>.
- [20] Whittaker, E. T., “Xviii.—on the functions which are represented by the expansions of the interpolation-theory,” *Proceedings of the Royal Society of Edinburgh* **35**, 181–194 (1915).
- [21] Barrett, H. H. and Myers, K. J., [*Foundations of image science*], John Wiley & Sons (2013).
- [22] Soummer, R., Pueyo, L., Sivaramakrishnan, A., and Vanderbei, R. J., “Fast computation of lyot-style coronagraph propagation,” *Opt. Express* **15**(24), 15935–15951 (2007).
- [23] Gray, R. M., “Toeplitz and circulant matrices: A review,” *Foundations and Trends in Communications and Information Theory* **2**(3), 155–239 (2006).