Journal of the Optical Society of America A
OPTICS, IMAGE SCIENCE, AND VISION

Techniques for arbitrary sampling in two-dimensional Fourier transforms

Alden S. Jurling,[1,2] Matthew D. Bergkoetter,[1,2] and James R. Fienup[1,*]

[1]The Institute of Optics, University of Rochester, Rochester, New York 14627, USA
[2]NASA Goddard Space Flight Center, Greenbelt, Maryland 20771, USA
*Corresponding author: fienup@optics.rochester.edu

Received 15 June 2018; revised 6 September 2018; accepted 9 September 2018; posted 10 September 2018 (Doc. ID 326354); published 3 October 2018

In this paper, we discuss two effective methods for computing optical propagations using two-dimensional (2D) discrete Fourier transforms: the matrix triple product (MTP) and the chirp *z*-transform (CZT) and analyze their performance both in theory and via benchmarks compared to the performance of a traditional padded fast Fourier transform (FFT). We show that, in many regimes of interest for phase-retrieval algorithms, the MTP or CZT is comparable to or better than the FFT in terms of run time while offering more flexible control over the sampling. We propose that for many applications, the CZT makes a robust general purpose alternative to the padded 2D FFT.

OCIS codes: (070.7345) Wave propagation; (000.4430) Numerical approximation and analysis; (070.0070) Fourier optics and signal processing; (100.5070) Phase retrieval; (010.7350) Wave-front sensing.

https://doi.org/10.1364/JOSAA.35.001784

1. INTRODUCTION

In image reconstruction and phase retrieval, the two-dimensional (2D) Fourier transform is very commonly used to model an optical propagation from the pupil of an imaging system to an image plane. When modeled numerically using sampled grids in a computer, propagations are typically implemented using a fast Fourier transform (FFT) algorithm [1]. The use of the FFT is advantageous because it exhibits favorable asymptotic scaling compared to direct integration methods and because highly optimized implementations [2,3] are available. However, the use of the FFT also imposes a fixed relationship between the sample spacing in its input and output domains based on the integer transform length. The array length and sampling relationship can be controlled coarsely through zero-padding, but only in integer steps. This fixed relationship is not desirable for some applications, in particular broadband phase retrieval and phase retrieval in the presence of significant chromatic aberrations. As a specific example of the broadband case, consider wavefront sensing for alignment of an astronomical telescope, where it is advantageous to maximize the signal-to-noise ratio in the image of a star by using the widest spectral filter available [4,5]. With regard to chromatic aberrations, recent work in phase retrieval for measurement of residual dispersion in chirped-pulse amplification (CPA) lasers has shown fine control of the simulated wavelength can resolve ambiguities in the sign of the dispersion [6,7].

Furthermore, the FFT assumes every pixel of its input domain is potentially nonzero and every pixel of its output domain is of interest. In practice, many phase retrieval and image reconstruction algorithms work in a regime where fields are constructed such that their intensities will be Nyquist-sampled. In this case, the input pupil fields are zero-padded such that half of the input array in each dimension is equal to zero. Furthermore, when computing point spread functions (PSFs) for phase retrieval, it often happens that only the inner portion of the simulated array is of interest, either because of the aliasing introduced in the other regions by the discretization of the pupil or because there is simply little PSF energy in those other regions. In those cases, alternative transforms that allow for the smaller nonzero extent of the pupil and limited region of interest in the image may be more computationally efficient.

In this work, we explore three methods for computing the transforms that allow arbitrary sampling and limited regions of interest in the pupil and image. Each of these makes different trade-offs among speed, flexibility, and implementation difficulty. Naïve direct integration of the discrete Fourier transform (DFT) is discussed in Section 2.C, the matrix triple product (MTP) DFT is derived in Section 2.D, and the chirp *z*-transform (CZT) is explained in Section 2.E. In Section 3, we demonstrate how the need to accurately simulate complex physical optics models changes the relative computational costs in several specific examples. We show that the MTP and CZT

are advantageous over the FFT in many situations, particularly when flexibility in choosing sample spacing is important.

2. THEORY

A. General

Let the variables x and y denote the continuous coordinates of the pupil (Fourier transform input) plane and f_x and f_y denote their Fourier spatial frequency conjugate variables (for a summary of symbols used throughout this paper, see Table 2). Consider a complex optical field $g(x, y)$ with finite extent D_x and D_y in the x and y directions, respectively. Its continuous Fourier transform is given by

$$G(f_x, f_y) = \iint_{-\infty}^{\infty} g(x, y) \exp[-2\pi i(f_x x + f_y y)]\mathrm{d}x\mathrm{d}y. \quad (1)$$

This continuous integral, while useful for theoretical analysis and for some special cases, is not suitable for computer modeling of most optical fields. For practical models implemented in the computer, we need to have discrete analogs to $G(f_x, f_y)$ and $g(x, y)$ and to compute finite sums. The discrete analog we will use can be written as

$$G[r, s] = \sum_{m,\, n \in \mathcal{M}, \mathcal{N}} g[m, n] \exp[-2\pi i(mr\Delta x\Delta f_x + ns\Delta y\Delta f_y)], \quad (2)$$

where the sets $\mathcal{M}$ and $\mathcal{N}$ represent the range of values of m and n defining the nonzero area of g. If $g(x, y)$ is sampled with intervals Δx and Δy, then $g[m, n]$ is nonzero over a rectangle of width M by N pixels defined by

$$M = \frac{D_x}{\Delta x}, \qquad N = \frac{D_y}{\Delta y}. \quad (3)$$

Common choices for the range of indices are

$$\mathcal{M} = \{0, \ldots, M - 1\} \quad (4)$$

and

$$\mathcal{M} = \{-\lfloor M/2 \rfloor, \ldots, \lceil M/2 \rceil - 1\}, \quad (5)$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote floor and ceiling, respectively, and likewise for $\mathcal{N}$. Similarly, if we are interested in the output plane $G(f_x, f_y)$ only over a region of dimensions C_x by C_y, then the corresponding region of interest in pixels in $G[r, s]$ will be of size R by S, defined by

$$R = \frac{C_x}{\Delta f_x}, \qquad S = \frac{C_y}{\Delta f_y}. \quad (6)$$

We can also write the DFT [Eq. (2)] as

$$G[r, s] = \sum_{m,\, n \in \mathcal{M}, \mathcal{N}} g[m, n] \exp\left[-2\pi i\left(\frac{mr}{K} + \frac{ns}{L}\right)\right], \quad (7)$$

where

$$K = \frac{1}{\Delta x\Delta f_x}, \qquad L = \frac{1}{\Delta y\Delta f_y}. \quad (8)$$

As we will later see in detail, K and L correspond to implicit periods of both $g[m, n]$ and $G[r, s]$, and must be larger than the regions of interest defined by M, N, R, and S in order to prevent missing data and introducing artifacts. Figure 1 shows a visual interpretation of these values. We note that, at this stage, there is no need for the periods K and L to be integer-valued. While integer periodicity is often a useful assumption, it is not strictly necessary for the correctness of the DFT.

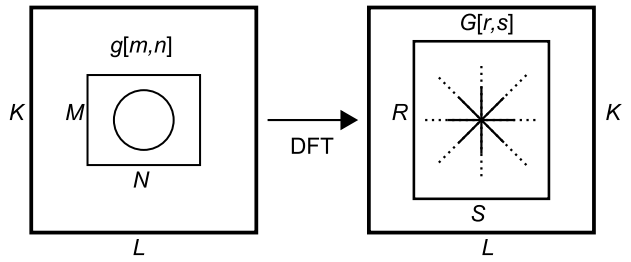

Fig. 1. Visual depiction of the sizes of the input array g ($M \times N$), output array G ($R \times S$), and DFT period ($K \times L$).

In the next section, we will analyze the discrete representation [Eq. (7)] by reconstructing it through a series of approximations to $g(x, y)$. First we make it sampled, then periodic, then both sampled and periodic. Although a periodic and sampled pupil is nonphysical, it is the model that our computer simulations implicitly use when we attempt to represent real systems. Understanding the sampling and aliasing trade-offs implicit in its construction is important. When dealing with the continuous quantities in what follows, a subscript S will denote a sampled version of a quantity, while a subscript P will denote a periodic version of the quantity.

As a tool to explore periodic and sampled representations, we will use the impulse train or "comb" function [8]:

$$\mathrm{comb}(\xi, \eta) \equiv \sum_{p,\, q=-\infty}^{\infty} \delta(\xi - p)\delta(\eta - q), \quad (9)$$

whose Fourier transform pair [8] is

$$\mathrm{comb}(\xi, \eta) \overset{\mathrm{FT}}{\longleftrightarrow} \mathrm{comb}(f_\xi, f_\eta). \quad (10)$$

Using the comb, we can represent a sampled pupil as

$$\begin{aligned} g_{\mathrm{S}}(x, y) &= \mathrm{comb}\left(\frac{x}{\Delta x}, \frac{y}{\Delta y}\right) g(x, y) \\ &= \sum_{m,\, n \in \mathcal{M}, \mathcal{N}} g(m\Delta x, n\Delta y)\delta(x - m\Delta x)\delta(y - n\Delta y)\Delta x\Delta y, \end{aligned} \quad (11)$$

the Fourier transform of which is

$$\begin{aligned} G_{\mathrm{P}}(f_x, f_y) &= G(f_x, f_y) * \mathrm{comb}(f_x\Delta x, f_y\Delta y)\Delta x\Delta y \\ &= \sum_{p,\, q=-\infty}^{\infty} G\left(f_x - \frac{p}{\Delta x}, f_y - \frac{q}{\Delta y}\right), \end{aligned} \quad (12)$$

where $*$ denotes 2D convolution. This convolution with the comb creates a periodic replication of $G(f_x, f_y)$ in the Fourier domain. At this point, f_x and f_y are still continuous quantities, but by sampling $g(x, y)$ we have forced $G(f_x, f_y)$ to be periodic (indicated by the P subscript):

$$G_{\mathrm{P}}\left(f_x + \frac{p'}{\Delta x}, f_y + \frac{q'}{\Delta y}\right) = G_{\mathrm{P}}(f_x, f_y), \quad (13)$$

where p' and q' are integers. This periodicity is the origin of the wraparound aliasing artifacts in images computed via FFT, where, for example, a streak exiting one side of the Fourier transform of an image will appear to enter on the opposite side and continue. It should be understood that this aliasing is not a result of the fixed FFT sampling relationships, but rather a fundamental consequence of the uniform sampling of the pupil. Once the discrete sample locations are fixed, both the maximum unique field of view of $G_{\mathrm{P}}(f_x, f_y)$ and the form of the aliasing due to wraparound therein is determined. Although the algorithms we will explore in the following sections give us the freedom to sample $G_{\mathrm{P}}(u, v)$ flexibly and oversample it arbitrarily, they cannot eliminate or reduce the intrinsic aliasing in $G_{\mathrm{P}}(f_x, f_y)$. The intrinsic aliasing can be reduced by selecting a smaller Δx, or by modifying the nominal g to reduce high-frequency content.

The second step toward the discrete transform we will consider is making the input field periodic, which will give us a sampled output without sampling $g(x, y)$:

$$g_{\mathrm{P}}(x, y) = g(x, y) * \mathrm{comb}\left(\frac{x}{T_x}, \frac{y}{T_y}\right)\frac{1}{T_x T_y}, \tag{14}$$

where we require $T_x \geq D_x$ and $T_y \geq D_y$ to avoid overlap between the periods. The factor of $(T_x T_y)^{-1}$ is included to preserve the units of g_{P}. Defined this way, $g_{\mathrm{P}}(x, y)$ satisfies the condition

$$g_{\mathrm{P}}(x, y) = g_{\mathrm{P}}(x + T_x\tilde{p}, y + T_y\tilde{q}), \tag{15}$$

where $\tilde{p}$ and $\tilde{q}$ are arbitrary integers. The Fourier transform of $g_{\mathrm{P}}(x, y)$ is

$$G_{\mathrm{S}}(f_x, f_y) = G(f_x, f_y)\mathrm{comb}(f_x T_x, f_y T_y), \tag{16}$$

which is a sampled model but has no wraparound artifacts. Unfortunately, because it requires computing a continuous Fourier transform via integration, G_{S} is not applicable to simulating the general case with arbitrary pupils and aberrations, which typically must be handled digitally using discrete models.

A pupil model that is fully realizable in the computer must be discrete in both the input and output. To achieve this, we construct a pupil-domain model that is both sampled and periodic. We do this by making the sampled model from Eq. (11) periodic with periods T_x and T_y, as we did in Eq. (14):

$$\begin{aligned} g_{\mathrm{SP}}(x,y) &= g_{\mathrm{S}}(x,y) * \mathrm{comb}\left(\frac{x}{T_x}, \frac{y}{T_y}\right)\frac{1}{T_x T_y} \\ &= \left[\mathrm{comb}\left(\frac{x}{\Delta x}, \frac{y}{\Delta y}\right) g(x,y)\right] * \mathrm{comb}\left(\frac{x}{T_x}, \frac{y}{T_y}\right)\frac{1}{T_x T_y}, \end{aligned} \tag{17}$$

which has Fourier transform

$$\begin{aligned} G_{\mathrm{PS}}(f_x, f_y) &= G_{\mathrm{P}}(f_x, f_y)\mathrm{comb}(f_x T_x, f_y T_y) \\ &= [G(f_x, f_y) * \mathrm{comb}(f_x \Delta x, f_y \Delta y)\Delta x \Delta y] \\ &\quad \times \mathrm{comb}(f_x T_x, f_y T_y), \end{aligned} \tag{18}$$

which is both periodic and sampled; it has the same period $1/\Delta x$ and $1/\Delta y$ as Eq. (12). We can also write this as

$$\begin{aligned} &G_{\mathrm{PS}}(f_x, f_y) \\ &= \sum_{r,s=-\infty}^{\infty} G_{\mathrm{P}}(f_x, f_y)\delta(f_x - r\Delta f_x)\delta(f_y - s\Delta f_y)\Delta f_x \Delta f_y, \end{aligned} \tag{19}$$

where $\Delta f_x = 1/T_x$ and $\Delta f_y = 1/T_y$. If we introduce the discrete quantity $G[r, s]$ as the amplitude of the delta functions under the sum we get

$$G[r, s] = G_{\mathrm{P}}(r\Delta f_x, s\Delta f_y) \tag{20}$$

and

$$\begin{aligned} &G_{\mathrm{PS}}(f_x, f_y) \\ &= \sum_{r,s=-\infty}^{\infty} G[r, s]\delta(f_x - r\Delta f_x)\delta(f_y - s\Delta f_y)\Delta f_x \Delta f_y, \end{aligned} \tag{21}$$

where $G[r, s]$ is the discrete quantity such that its values are the complex amplitudes of the δ functions in $G_{\mathrm{PS}}(f_x, f_y)$. This makes $G[r, s]$ a sampled version of $G_{\mathrm{P}}(f_x, f_y)$. We can observe that since $G_{\mathrm{PS}}(f_x, f_y)$ has periods $1/\Delta x$ and $1/\Delta y$ in continuous coordinates, $G[r, s]$ has periods $1/(\Delta f_x \Delta x)$ and $1/(\Delta f_y \Delta y)$ in pixels, which are the definitions of K and L previously given in Eq. (8). We can likewise define the discrete version of $g_{\mathrm{SP}}(x, y)$, $g[m, n]$, such that

$$g_{\mathrm{SP}}(x, y) = \sum_{m,n=-\infty}^{\infty} g[m, n]\delta(x - m\Delta x)\delta(y - n\Delta y)\Delta x \Delta y. \tag{22}$$

We are again defining the discrete quantity $g[m, n]$ as the amplitudes of the δ function samples in $g_{\mathrm{SP}}(x, y)$. By construction, $g_{\mathrm{SP}}(x, y)$ has periods $T_x = 1/\Delta f_x$ and $T_y = 1/\Delta f_y$ in continuous space and, correspondingly, $g[m, n]$ has periods $1/(\Delta f_x \Delta x)$ and $1/(\Delta f_y \Delta y)$ in pixels; note that this is the same as the periods of $G[r, s]$.

To show how $G[r, s]$ may be computed, we explicitly evaluate the Fourier transform of $g_{\mathrm{SP}}(x, y)$ in the form given by Eq. (22), and use the sifting property of the delta functions to find

$$G_{\mathrm{PS}}(f_x, f_y) = \sum_{m,n=-\infty}^{\infty} g[m, n]\exp[-2\pi i(f_x m\Delta x + f_y n\Delta y)]. \tag{23}$$

Using the knowledge that, by definition, $g[m, n]$ is periodic, we can rewrite the infinite summation in Eq. (23) in terms of summations over the nonzero area $(\mathcal{M}, \mathcal{N})$ of a single period, such that

$$\begin{aligned} &G_{\mathrm{PS}}(f_x, f_y) \\ &= \sum_{m',n'=-\infty}^{\infty}\left\{\sum_{m,n\in\mathcal{M},\mathcal{N}} g[m, n]\right. \\ &\quad \left.\times \exp[-2\pi i(f_x(m + Km')\Delta x + f_y(n + Ln')\Delta y)]\right\}, \end{aligned} \tag{24}$$

where m' and n' are introduced to index each repetition of the unique portion of $g[m, n]$. Replacing the periods K and L with their definition in terms of sample spacing in Eq. (8) and

moving the parts of the Fourier kernel independent of (m, n) out of the inner summation yields the product of two independent sums:

$$G_{\mathrm{PS}}(f_x, f_y) = \sum_{m', n'=-\infty}^{\infty} \exp\left[-2\pi i\left(m'\frac{f_x}{\Delta f_x} + n'\frac{f_y}{\Delta f_y}\right)\right] \times \sum_{m,n\in\mathcal{M},\mathcal{N}} g[m,n]\exp[-2\pi i(f_x m\Delta x + f_y n\Delta y)]. \tag{25}$$

The infinite sum on the first line of Eq. (25) is only nonzero at points where f_x and f_y are integer multiples of Δf_x and Δf_y, respectively, while at the nonzero points, it goes towards infinity. Therefore, Eq. (25) may be written in terms of delta functions to yield

$$G_{\mathrm{PS}}(f_x, f_y) = \sum_{r,s=-\infty}^{\infty} \delta(f_x - r\Delta f_x)\delta(f_y - s\Delta f_y) \times \sum_{m,n\in\mathcal{M},\mathcal{N}} g[m,n]\exp[-2\pi i(mr\Delta x\Delta f_x + ns\Delta y\Delta f_y)]. \tag{26}$$

We recognize this expression is the same weighted set of delta functions in Eq. (21), with the weights $G[r, s]$ now given by

$$G[r,s] = \sum_{m,n\in\mathcal{M},\mathcal{N}} g[m,n]\exp[-2\pi i(mr\Delta x\Delta f_x + ns\Delta y\Delta f_y)], \tag{27}$$

which is equivalent to the definition of the DFT stated previously in Eq. (7) in terms of K and L. If K and L are integers, this gives the standard 2D DFT/FFT; but in general, they need not be integers.

We define two sampling ratios Q^{P} and Q^{I} in terms of the sizes of the nonzero regions of interest in the pupil and image planes, respectively, with relation to the fundamental periods K and L:

$$Q_{\mathrm{x}}^{\mathrm{P}} = \frac{K}{M} \qquad Q_{\mathrm{y}}^{\mathrm{P}} = \frac{L}{N}$$
$$Q_{\mathrm{x}}^{\mathrm{I}} = \frac{K}{R} \qquad Q_{\mathrm{y}}^{\mathrm{I}} = \frac{L}{S}. \tag{28}$$

In the common case of square arrays and equal sample spacing in both directions, we have $K = L$, $M = N$, and $R = S$, and then $Q_{\mathrm{x}}^{\mathrm{P}} = Q_{\mathrm{y}}^{\mathrm{P}} = Q^{\mathrm{P}}$ and $Q_{\mathrm{x}}^{\mathrm{I}} = Q_{\mathrm{y}}^{\mathrm{I}} = Q^{\mathrm{I}}$. Notice that since the nonzero region and region of interest can usefully span at most one full period, we have $Q^{\mathrm{P}} \geq 1$, $Q^{\mathrm{I}} \geq 1$. Q^{P} relates to the finite extent of $g[m, n]$ and thus to the band limit of $G[r, s]$: With $Q^{\mathrm{P}} = 1$, $G[r, s]$ Nyquist samples $G(f_x, f_y)$ and with larger values of Q^{P}, it oversamples $G(f_x, f_y)$; $Q^{\mathrm{P}} = 2$ corresponds to Nyquist sampling for $|G(f_x, f_y)|^2$. We will see later that this definition of Q^{P} can be connected with the sampling factor from [9]. Q^{I} similarly relates the width of the region of interest in $G(f_x, f_y)$ to the wraparound period introduced by finite sampling of $g(x, y)$; it can be thought of as a model fidelity factor related to the acceptable amount of wraparound aliasing in $G[r, s]$; increasing Q^{I} moves the repeated copies of G further from the region of interest and reduces their influence.

Since Q^{I} and Q^{P} are both connected with the overall period of the discrete transform, we have the following relations:

$$MQ_{\mathrm{x}}^{\mathrm{P}} = RQ_{\mathrm{x}}^{\mathrm{I}} = K, \tag{29a}$$

$$NQ_{\mathrm{y}}^{\mathrm{P}} = SQ_{\mathrm{y}}^{\mathrm{I}} = L. \tag{29b}$$

Notice that K and L are the padded array size in a conventional FFT-based propagator. This relationship allows us to make a consistent choice of array sizes in terms of our desired sampling in the two domains. Only three of M, $Q_{\mathrm{x}}^{\mathrm{I}}$, R, and $Q_{\mathrm{x}}^{\mathrm{P}}$ can be chosen independently.

Typically in an application, some of the variables are stipulated by the physics of the problem, and some are free to be chosen by the modeler to meet the needs of the model. In the phase retrieval applications we will discuss, R and $Q_{\mathrm{x}}^{\mathrm{P}}$ are generally fixed, and $Q_{\mathrm{x}}^{\mathrm{I}}$ and M can be chosen (and likewise in y). In this case we express the requirement on M as

$$M = Q_{\mathrm{x}}^{\mathrm{I}}\frac{R}{Q_{\mathrm{x}}^{\mathrm{P}}}, \tag{30}$$

and similarly in y. Notice that the ratio on the right side is the number of Nyquist samples across the region of interest. The equality can be relaxed to a greater-than-or-equal requirement for computational purposes if we break the strict correspondence between M and the diameter of the nonzero region of $g(x, y)$ and allow it to be the width of potentially larger array that the nonzero part of $g(x, y)$ fits inside.

Regardless of the underlying physics, a particular set of M, R, $Q_{\mathrm{x}}^{\mathrm{I}}$, and $Q_{\mathrm{x}}^{\mathrm{P}}$ (and analogous y dimension quantities) uniquely defines a particular numerical DFT calculation problem. We will next consider several methods for calculating these DFTs, which produce the same numerical results but make different trade-offs in computational cost.

B. FFT

If we require K and L to be integers in Eq. (8), then Eq. (7) becomes the definition of a conventional DFT/FFT. In this case, both $G[r, s]$ and $g[m, n]$ are taken to be 2D arrays of size $K \times L$. The 2D FFT has asymptotic computational complexity:

$$t_{\mathrm{FFT}} \propto KL\log_2(KL). \tag{31}$$

Although we can make an FFT-based algorithm aware of the finite extent and region of interest by padding the input with zeros from size $M \times N$ up to size $K \times L$ and cropping the output down to size $R \times S$, the knowledge that many of the data points are unnecessary is not leveraged to improve the speed of the algorithm. We make the following observations about this formulation:

1. It has favorable asymptotic complexity for large arrays.
2. The image sample spacing is restricted by Eq. (8) with integer K and L.
3. It is efficient for transforms where the nonzero extent of g and the region of interest of G are similar to the array size $K \times L$, and conversely, is inefficient when either of those is much smaller than $K \times L$. In other words, the FFT is most efficient when $Q^{\mathrm{P}} = Q^{\mathrm{I}} = 1$.
4. If M and R are fixed, the cost of the FFT is driven by Q^{P} and Q^{I}.

C. Naïve Direct Integration

We can of course compute Eq. (2) directly by simply performing the sum over m and n. If we let the number of nonzero points in $g[r,s]$ be RS and the number of points of interest in $G[m,n]$ be MN, then the computational cost of the direct sum is

$$t_{\mathrm{INT}} \propto MNRS. \tag{32}$$

If the transform will be repeated many times, we can precompute a matrix of the complex kernel factors (the matrix will have size $MN \times RS$) and implement the transform as a matrix product with the vector (of length RS) representation of $g[r,s]$; that saves the potentially expensive complex exponential calculation and allows the use of fast linear algebra functions to compute the sum. Implementing the sum as a matrix product does not ultimately improve the asymptotic computational complexity. We can make the following observations about the direct integration method:

1. It has unfavorable asymptotic complexity, making it unacceptably slow for many practical numerical applications that require large arrays.
2. It is most efficient for transforms where either the nonzero extent of g or the region of interest in G is small.
3. It is extremely flexible; the location of the input and output samples of g and G can be arbitrary. Not only are arbitrary Δx, Δy, Δf_x, and Δf_y possible, but arbitrary samplings where the points do not fall on regular grids [e.g., are not described by Eq. (2)] are possible.

D. Separable DFT/MTP

For the cases typically of interest in phase retrieval, we do not require the full generality of the naïve direct integration method above. If we limit ourselves to the regularly sampled grids of the DFT from Eq. (2), we can improve on the naïve algorithm. By factoring the separable DFT kernel into two one-dimensional (1D) kernels and moving one sum inside the other to get

$$G[r,s] = \sum_m \exp(-2\pi i \Delta x \Delta f_x mr) \times \left[\sum_n g[m,n] \exp(-2\pi i \Delta y \Delta f_y ns)\right]_{ms}, \tag{33}$$

the term in brackets represents a whole 2D matrix, which is indexed by the subscript ms. The sum in brackets costs MNS to compute, while the outer sum costs RMS to compute, so the overall cost is cubic:

$$t_{\mathrm{MTP1}} \propto RMS + MNS = MS(N+R). \tag{34}$$

This is equivalent to performing a 1D DFT on each row, followed by 1D DFTs for every column. It can be implemented more efficiently, however, if we recognize that the two sums have the form of matrix products. If we define the following quantities:

$$\Omega_x[r,m] = \exp(-2\pi i \Delta x \Delta f_x mr), \tag{35}$$

$$\Omega_y[n,s] = \exp(-2\pi i \Delta y \Delta f_y ns), \tag{36}$$

we can write Eq. (33) as the MTP [10,11]:

$$G = \Omega_x g \Omega_y, \tag{37}$$

where Ω_x, g, and Ω_y are all interpreted as matrices. If the desired sample spacings can be held constant, Ω_x and Ω_y can be computed once and reused, alleviating the need to repeat the expensive trigonometric operations required to construct them. For the general case where $R \neq S \neq M \neq N$, the computational cost of this triple product depends on the order in which we compute the matrix products. If we group the triple product on the right, we get the same as Eq. (33) above:

$$G = \Omega_x (g \Omega_y), \tag{38}$$

and we again get Eq. (34), whereas if we group on the left, we get

$$G = (\Omega_x g)\Omega_y \tag{39}$$

and

$$t_{\mathrm{MTP2}} \propto RMN + RNS = NR(M+S). \tag{40}$$

In the case where $R \approx S \approx M \approx N \approx K \approx L$, the cost of the sum in Eq. (33) is $O(N^3)$, while naïve direct integration by Eq. (2) is $O(N^4)$, and the FFT is $O(N^2 \log N)$. In other words, the separable DFT achieves a linear speedup compared to direct integration and is slower than the FFT by less than a linear factor. While naïve direct integration is impractical for many problems because of its severe computational cost, the separable DFT remains practical for a much larger range of array sizes.

We may also write a simpler form of Eq. (40) in terms of the sampling ratios Q^{I} and Q^{P} for cases where the arrays are square, but $K \neq M$. Using Eq. (30) to replace M, we obtain

$$t_{\mathrm{MTP}} \propto \frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}} R^3 \left(1 + \frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}}\right). \tag{41}$$

While the complexity analysis offer no distinction between the explicit sums of Eq. (33) and the MTP of Eq. (38), in practice the MTP implementation is favored for two reasons. First, the matrix multiplication routines in high-performance linear algebra packages are highly optimized for performance and can achieve substantially better constant factor performance than a typical handwritten implementation of the sum. Second, state-of-the-art algorithms for multiplying two square matrices can achieve asymptotic performance better than the N^3 naïve matrix product used here, so in practice, depending on the numerical libraries employed, the matrix multiply DFT may actually scale somewhat better than this analysis indicates, as discussed in [12]. Because of these advantages, we do not consider the explicitly separated sum in our benchmarks.

Summarized,

1. The separable DFT (or MTP) has complexity midway between that of direct integration and the FFT.
2. Like direct integration, the separable DFT (or MTP) is most efficient when the nonzero extent of g or the region of interest of G is small.
3. The sample spacing and number of samples of g and G may be arbitrary, but both must be sampled on regular Cartesian grids.
4. If M and R are fixed, the cost of the separable DFT is independent of Q^{P} and Q^{I}.

The origins of the MTP and separable DFT techniques are difficult to trace. The MTP (for square matrices) was employed for computational savings in [10] and [11]. However, the essential insight is that a 2D DFT is separable into 1D DFTs along rows and columns. Indeed, transforming rows and columns in one dimension at a time is a common method for computing 2D FFTs, as mentioned in [13] and [14], while the matrix multiplication version is mentioned in [15], but none of these references appears to be the origin of the MTP.

E. CZT

In this section, as in Section 2.D, we maintain the restriction that the coordinate axes x, y, f_x, and f_y be regularly sampled, while allowing the sample spacings Δx, Δy, Δf_x, and Δf_y to be arbitrary. In this regime, we have an alternative to the MTP in the CZT algorithm [16], also known as Bluestein's FFT [17]. The algorithm works by treating the DFT as a convolution, which may be computed in the Fourier domain via FFTs. This gives it the same asymptotic complexity as an FFT, and a performance advantage over MTP for large arrays.

Following the formulation in [18], we define new parameters:

$$\alpha_x \equiv \Delta f_x \Delta x = \frac{1}{K}, \qquad \alpha_y \equiv \Delta f_y \Delta y = \frac{1}{L}, \tag{42}$$

which depend on the sample spacing values, and substitute them into Eq. (2) to obtain

$$G[r,s] = \sum_{m,n \in \mathcal{M},\mathcal{N}} g[m,n] \exp[-i2\pi(mr\alpha_x + ns\alpha_y)]. \tag{43}$$

A key insight is that the factors $2mr$ and $2ns$ may be rewritten as [17]

$$\begin{aligned} 2mr &= m^2 + r^2 - (r-m)^2, \\ 2ns &= n^2 + s^2 - (s-n)^2, \end{aligned} \tag{44}$$

which yield, when inserted into Eq. (43),

$$\begin{aligned} G[r,s] = \exp[-i\pi(r^2\alpha_x + s^2\alpha_y)] \\ \times \sum_{m,n \in \mathcal{M},\mathcal{N}} \{\exp[-i\pi(m^2\alpha_x + n^2\alpha_y)] \\ \times g[m,n] \exp\{i\pi[(r-m)^2\alpha_x + (s-n)^2\alpha_y]\}\}, \end{aligned} \tag{45}$$

$$G[r,s] = a[r,s] \sum_{m,n \in \mathcal{M},\mathcal{N}} b[m,n] g[m,n] h[r-m, s-n], \tag{46}$$

where we have defined a, b, and h as

$$a[r,s] \equiv \exp[-i\pi(r^2\alpha_x + s^2\alpha_y)], \tag{47a}$$

$$b[m,n] \equiv \exp[-i\pi(m^2\alpha_x + n^2\alpha_y)], \tag{47b}$$

$$h[m,n] \equiv \exp[i\pi(m^2\alpha_x + n^2\alpha_y)]. \tag{47c}$$

The summation in Eq. (46) is in the form of a discrete convolution between $h[r,s]$ and the product $b[r,s]g[r,s]$. According to the convolution theorem of Fourier transforms, this can be computed as a product in the Fourier domain; that is, one multiplies the Fourier transforms of the two factors, and then the inverse transforms their product. However, this discrete Fourier-domain convolution is circular, whereas the summation in Eq. (46) represents a noncircular convolution. We can make the circular convolution compute the correct sum by appropriately zero-padding $g[m,n]$, $b[m,n]$, and $h[m,n]$, while duplicating some values in $h[m,n]$, as explained in detail in [16] for 1D transforms. Here we show the analogous results for 2D DFTs. We denote the dimensions of the CZT's internal padded arrays as $K' \times L'$ and the padded quantities with a hat, and use & and || to represent "and" and "or," respectively:

$$\hat{g}[m,n] \equiv \begin{cases} g[m,n], & (0 \le m < M)\ \&\ (0 \le n < N) \\ 0, & (M \le m < K') \| (N \le n < L') \end{cases}, \tag{48a}$$

$$\hat{b}[m,n] \equiv \begin{cases} b[m,n], & (0 \le m < M)\ \&\ (0 \le n < N) \\ 0, & (M \le m < K') \| (N \le n < L') \end{cases}, \tag{48b}$$

$$\hat{h}_x[m] \equiv \begin{cases} \exp[i\pi m^2\alpha_x], & 0 \le m \le R-1 \\ \text{arbitrary}, & R-1 < m < K'-M+1 \\ \exp[i\pi(m-K')^2\alpha_x], & K'-M+1 \le m < K' \end{cases}, \tag{48c}$$

$$\hat{h}_y[n] \equiv \begin{cases} \exp[i\pi n^2\alpha_y], & 0 \le n \le S-1 \\ \text{arbitrary}, & S-1 < n < L'-N+1 \\ \exp[i\pi(n-L')^2\alpha_y], & L'-N+1 \le n < L' \end{cases}, \tag{48d}$$

$$\hat{h}[m,n] = \hat{h}_x[m]\hat{h}_y[n]. \tag{48e}$$

The minimum amount of padding required to prevent wraparound artifacts from the circular convolution is that which ensures that the internal DFT period is larger than the sum of the input and output array sizes, i.e.,

$$K' \ge M + R - 1, \tag{49a}$$

$$L' \ge N + S - 1. \tag{49b}$$

However, additional padding may be added in order to bring the array sizes up to some value that is particularly advantageous for an FFT, e.g., a highly composite number. It is important to note that these manipulations are only applied to the forward and inverse Fourier transform pair that are used internally by the CZT and do not affect the sample spacings of the input and output data arrays g and G. Instead, the sample spacings are reflected in the values α_x and α_y. This is an important difference between the CZT and conventional FFT and is key to its flexibility when modeling optical propagation.

With the above quantities defined, the algorithm may be expressed succinctly as

$$\boldsymbol{G} = \boldsymbol{a} \circ \mathrm{crop}[(\hat{\boldsymbol{b}} \circ \hat{\boldsymbol{g}}) * \hat{\boldsymbol{h}}], \tag{50}$$

or, equivalently in the Fourier domain, as

$$\boldsymbol{G} = \boldsymbol{a} \circ \mathrm{crop}[\mathcal{F}^{-1}\{\mathcal{F}\{\hat{\boldsymbol{b}} \circ \hat{\boldsymbol{g}}\} \circ \hat{\boldsymbol{H}}\}], \tag{51}$$

where $\hat{H}$ is the Fourier transform of $\hat{h}$, the bold symbols represent matrices, and the $\circ$ symbol denotes the Hadamard product (i.e., element-wise multiplication). The arrays containing $(\hat{\boldsymbol{b}} \circ \hat{\boldsymbol{g}}) * \hat{\boldsymbol{b}}$ and $\mathcal{F}^{-1}\{\mathcal{F}\{\hat{\boldsymbol{b}} \circ \hat{\boldsymbol{g}}\} \circ \hat{\boldsymbol{H}}\}$ are of size $K' \times L'$, but only the first $R \times S$ are multiplied by $\boldsymbol{a}$. The "crop" statement represents a function that removes the unused portions of the $K' \times L'$ arrays.

It should also be noted that quantities $\boldsymbol{a}$, $\hat{\boldsymbol{b}}$, and $\hat{\boldsymbol{H}}$ are all independent of the particular input data $\boldsymbol{g}$ being transformed, and instead only depend on the choice of sample densities and array sizes in the input and output domains. For iterative algorithms such as phase retrieval, which run the same transform many times with different data, these need only be computed once and can be reused on subsequent iterations.

The asymptotic complexity of the CZT algorithm is dominated by its internal pair of FFTs, which is

$$t_{\mathrm{CZT}} \propto K'L' \log_2(K'L'). \tag{52}$$

If the minimal amount of padding [Eq. (49)] is applied to the CZT's internal arrays, then Eq. (52) becomes

$$t_{\mathrm{CZT}} \propto (M+R)(N+S) \log_2[(M+R)(N+S)]. \tag{53}$$

We may also express Eq. (53) in terms of Q^{P} and Q^{I} by substituting from Eq. (29) to obtain, for the case of $R = S$, $Q_x^{\mathrm{P}} = Q_y^{\mathrm{P}} = Q^{\mathrm{P}}$, and $Q_x^{\mathrm{I}} = Q_y^{\mathrm{I}} = Q^{\mathrm{I}}$:

$$t_{\mathrm{CZT}} \propto \left[R\left(1+\frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}}\right)\right]^2 \log_2 R\left(1+\frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}}\right). \tag{54}$$

In summary,

1. The CZT has an asymptotic computational complexity of the form $N^2 \log N$ [Eq. (52)], similar to that of an FFT.
2. The CZT is, in theory, a constant factor slower than a single FFT, since it uses two FFTs of at least twice the data array size, and it contains extra multiplicative factors.
3. In contrast with the standard FFT, the CZT algorithm does not impose any restrictions of its own on the sample spacing or number of samples in g or G. (We note that the usual Nyquist sampling requirements still apply, as with any DFT.)
4. In practice, the arrays used internally by the CZT can be zero-padded to a size that allows for more efficient FFTs without affecting the sample spacings of the input and output arrays.

3. APPLICATIONS

A. Polychromatic Phase Retrieval

In this section, we wish to explore the application of the propagators, discussed previously, to phase-retrieval algorithms. We will assume that the continuous amplitude transmittance of our system $A(x, y; \lambda)$ is discretely sampled, to give $A[m, n; \lambda]$, and likewise the wavefront aberrations $W(x, y; \lambda)$ (in units of optical path length) are sampled to give $W[m, n; \lambda]$. The λ indicates that these two quantities may vary with wavelength. In order to account for the broadband nature of the systems, we will modify the propagators to control sampling while holding the discretization of W and A fixed. This is a choice: it is also possible to interpolate A and W or to use a polynomial basis for W and an analytic model for A that allows us to produce different sampled approximations of A and W and hold the propagation model fixed. However, if we ultimately hope to obtain a point-by-point amplitude or pupil reconstruction, it is desirable to keep a fixed-pupil model so that a single point-by-point model can be used for all wavelengths without interpolation, as in [4].

The PSF at the image plane of a polychromatic system in the Fraunhofer approximation in continuous coordinates is

$$I(u,v) = \int_{-\infty}^{\infty} w(\lambda) \left| \iint_{-\infty}^{\infty} A(x,y;\lambda) \exp\left[\frac{i2\pi}{\lambda} W(x,y;\lambda)\right] \times \exp\left[\frac{-i2\pi}{\lambda f}(ux+vy)\right] \mathrm{d}x\mathrm{d}y \right|^2 \mathrm{d}\lambda, \tag{55}$$

where $w(\lambda)$ is the spectrum of the source, f is the focal length, and $u/\lambda f$ is the equivalent of f_x in Eq. (1). This formulation includes the potential of both chromatic wavefront aberrations $W(x, y; \lambda)$ and wavelength-dependent variations in transmittance, $A(x, y; \lambda)$. The discrete form is

$$I[r,s] = \sum_k w_k \left| \sum_{m,n} A[m,n;\lambda_k] \exp\left\{\frac{i2\pi}{\lambda_k} W[m,n;\lambda_k]\right\} \times \exp\left[\frac{-i2\pi}{\lambda_k f}(mr\Delta u\Delta x + ns\Delta v\Delta y)\right] \right|^2, \tag{56}$$

where Δx and Δy are sample spacings in the pupil, Δu and Δv are sample spacings in the image plane, and k indexes the wavelengths. We recognize the sum over m and n as a DFT. We can write the exponential term in the form of the DFT from Eq. (7) with the substitutions

$$\Delta f_x \to \frac{\Delta u}{\lambda_k f}, \qquad \Delta f_y \to \frac{\Delta v}{\lambda_k f}. \tag{57}$$

The width of the clear aperture of A (given by D_x and D_y) determines the minimum size of the $M \times N$ input array that contains all nonzero values in the pupil, according to Eq. (3). We can use these quantities and combine Eqs. (8) and (28) to arrive at another expression for Q^{P}, based on the physical parameters of the system [9]:

$$Q_{x,k}^{\mathrm{P}} = \frac{\lambda_k f}{D_x \Delta u}, \qquad Q_{y,k}^{\mathrm{P}} = \frac{\lambda_k f}{D_y \Delta v}. \tag{58}$$

Similarly, the period of the DFT in each dimension in Eq. (8) may now be given as

$$K_k = \frac{\lambda_k f}{\Delta x \Delta u}, \qquad L_k = \frac{\lambda_k f}{\Delta y \Delta v}. \tag{59}$$

Typically, Q^{P} is determined by the physical parameters of the system being modeled, while Δx and Δy are chosen by the modeler subject to sampling fidelity requirements of the pupil phase and amplitude. Once these are chosen, M and N are determined through Eq. (3), and thus the DFT periods are fixed via Eq. (59). When we have pixelated measured data in both the pupil and image planes, matching the physical sample spacings in both domains without interpolation is precluded by use of an FFT. In this scenario, an arbitrarily sampled method is needed.

We will now derive requirements relating M and N to the image size. Let us assume that the vast majority of the energy of the PSF of our system falls within a rectangle of size R by S pixels, and that to avoid the aliased energy in the image we wish

to have a factor of Q^{I} extra space around the image, as introduced in Eq. (28). Choosing $Q^{\mathrm{I}} = 2$ is a good balance between computational cost and accuracy, but other choices are possible. Since K and L are also the spatial periods of the computed fields in the image plane, we can express the extra space requirement as

$$K \geq Q^{\mathrm{I}} R, \qquad L \geq Q^{\mathrm{I}} S \tag{60}$$

for any particular wavelength. Through Eq. (29), the following conditions on M and N hold for all wavelengths:

$$M \geq \frac{Q^{\mathrm{I}} R}{Q_{\mathrm{x}}^{\mathrm{P}}}, \qquad N \geq \frac{Q^{\mathrm{I}} S}{Q_{\mathrm{y}}^{\mathrm{P}}}, \tag{61}$$

where these should be satisfied for whatever wavelengths carry the most stringent sampling requirements. Handling polychromatic propagation with the MTP and CZT methods is straightforward. For the MTP, one may specify a distinct Ω_k in Eq. (35) for each kth wavelength, and the CZT distinct arrays for Eqs. (48b)–(48d) may be computed with a different α_k for each wavelength. These changes do not affect the array sizes the algorithms operate on, and so the choice of wavelength components does not affect the per-wavelength computational cost. For the FFT, the pad sizes depend on wavelength, and so the cost can be affected.

B. Broadband Achromatic Phase Retrieval

In this section, we narrow our consideration to the achromatic broadband case, where $A[m, n; \lambda_k] = A[m, n]$ and $W[m, n; \lambda_k] = W[m, n]$: the system pupil and aberrations are independent of wavelength. In this case, the only wavelength dependence of the PSF is through the changes in the propagation calculation with wavelength. We will assume in this section that the bandwidth is large enough that these changes are significant, and more than one discrete wavelength is required to accurately model the system.

For the sampling requirements given in Eq. (61), we will consider the shortest wavelength, λ_0, which in the achromatic case will carry the most demanding sampling requirements. If we take the minimum for M and N, we can plug back into Eq. (61) and get a simple expression for the FFT lengths of the individual transforms:

$$K_k = \frac{\lambda_k}{\lambda_0} Q^{\mathrm{I}} R, \qquad L_k = \frac{\lambda_k}{\lambda_0} Q^{\mathrm{I}} S. \tag{62}$$

If $R \approx S$, the cost to compute an FFT propagation for an individual wavelength will be

$$t_{\mathrm{FFT},k} \propto \left(\frac{\lambda_k}{\lambda_0} Q^{\mathrm{I}} R\right)^2 \log_2\left(\frac{\lambda_k}{\lambda_0} Q^{\mathrm{I}} R\right), \tag{63}$$

with the relative cost of the longer wavelengths increasing faster than the square of the wavelength. By contrast, the CZT calculation of the same DFT has a cost given by Eq. (52), independent of λ_k/λ_0. So we expect that, for problems with large enough bandwidth, the CZT form will outperform the FFT form, as larger values of λ_k/λ_0 increase the run time in Eq. (63) relative to that of Eq. (54). This effect will be demonstrated with benchmarks in Section 4.B.

C. Narrowband Phase Retrieval with Chromatic Aberrations

In this section, we consider scenarios where the choice to simulate multiple wavelengths is motivated not by a large bandwidth, but by the presence of large chromatic aberrations over a relatively narrow bandwidth, such as in the case of CPA lasers with misaligned grating stretchers or compressors [6,19–22]. If the propagation is computed with FFTs, then a small difference between adjacent spectral components is difficult to represent with Eq. (62), since array sizes are restricted to integer values. If $\Delta\lambda$ is the spacing required between adjacent wavelengths to adequately model the chromatic aberrations, and the shortest wavelength is λ_0, then increments in the array dimensions ΔK and ΔL for the FFTs are

$$\Delta K = \frac{\Delta\lambda}{\lambda_0} K_0, \qquad \Delta L = \frac{\Delta\lambda}{\lambda_0} S_0. \tag{64}$$

In order for ΔK and ΔL to be integers, the expressions to the right of the equal signs in Eq. (64) must be at least unity. Since the reference wavelength λ_0 and the necessary spectral sample spacing $\Delta\lambda$ are determined mainly by the physical system, the burden to meet the integer spacing requirement falls to K_0 and L_0:

$$K_0, L_0 \geq \frac{\lambda_0}{\Delta\lambda}. \tag{65}$$

We note that in some cases we could be flexible in our choice of λ_0, but this is of limited usefulness, since only a small range of wavelengths is of interest for a narrowband system.

The requirement in Eq. (65) tends to increase the computational cost for smaller wavelength spacings. Since the padded array size for the FFT must be greater than or equal to $\lambda_0/\Delta\lambda$, we modify Eq. (63) for narrowband polychromatic propagation to show that

$$t_{\mathrm{FFT},k} \propto \left[\frac{\lambda_k}{\lambda_0} \max\left(Q^{\mathrm{I}} R, \frac{\lambda_k}{\Delta\lambda}\right)\right]^2 \log_2\left[\frac{\lambda_k}{\lambda_0} \max\left(Q^{\mathrm{I}} R, \frac{\lambda_k}{\Delta\lambda}\right)\right]. \tag{66}$$

Comparison of Eq. (66) with the CZT complexity in Eq. (54) reveals how a CZT may be faster than the equivalent padded FFT for this application. For example, if the desired sample spacing is 1 nm and the reference wavelength is 1000 nm, then the minimum array size for a padded FFT is 1000 pixels, and the cost of propagating the reference wavelength component (where $\lambda_k = \lambda_0$) is $1000^2 \log_2 1000^2 \approx 20 \times 10^6$ operations. If, for instance, the region of interest in the image plane is 256 pixels wide and we use $Q^{\mathrm{I}} = Q^{\mathrm{P}}$, then the cost of one of the internal FFTs in the CZT is $256^2 \log_2 256^2 \approx 1 \times 10^6$ operations. Since there are two internal FFTs in a CZT (assuming $\hat{H}$ in Eq. (51) was precomputed), we can say that the FFT has to do on the order of $(20 \times 10^6)/(2 \times 10^6) = 10$ times more operations than the CZT. This argument based on the asymptotic complexity provides an intuitive explanation for cases where a CZT-based propagation is faster than an equivalent propagation using an FFT, but we also remember that these formulas neglect leading scaling factors and constant minimum values that will affect the run time in practice, and we will not attempt to determine these constants analytically.

Table 1. Summary of Computational Complexities

Algorithm	Complexity
FFT	
Monochromatic	$(Q^{\mathrm{I}}R)^2 \log_2(Q^{\mathrm{I}}R)$
Broadband	$\left(\frac{\lambda_k}{\lambda_0} Q^{\mathrm{I}}R\right)^2 \log_2\left(\frac{\lambda_k}{\lambda_0} Q^{\mathrm{I}}R\right)$
Narrowband	$\left(\frac{\lambda_k^2}{\lambda_0 \Delta\lambda}\right)^2 \log_2\left(\frac{\lambda_k^2}{\lambda_0 \Delta\lambda}\right)$
CZT	$\left[R\left(1+\frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}}\right)\right]^2 \log_2 R\left(1+\frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}}\right)$
MTP	$\frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}} R^3 \left(1+\frac{Q^{\mathrm{I}}}{Q^{\mathrm{P}}}\right)$

Instead, we will explore the real-world performance through comparative benchmarks in Section 4.

Finally, we remark that the restriction in Eq. (65) also drives requirements on the pupil array sample spacings (Δx and Δy). Combining the relations in Eqs. (3), (58), and (29) and evaluating them for λ_0 gives

$$\frac{\lambda_0 f}{\Delta x \Delta u} = K_0. \tag{67}$$

Since the values of λ_0, f, and Δu are driven by the physical system (Δu is typically chosen to match the pixel pitch of the image-plane detector), a large increase in K_0, like the factor of 4 increase in the example above, would need to be accompanied by a proportional change in the selection of Δx to preserve these physical parameters. This is especially inconvenient if the system model has an initial estimate of the wavefront that has been measured in the pupil plane in addition to data in the image plane, as in [7,22]. In this case, it is desirable to choose a pupil sample spacing that matches the initial estimate instead of interpolating the measured data to another grid. Matching Δx to the physical pixel pitch is trivial with the MTP or CZT, but requires some extra considerations for the FFT.

The computational complexities for each of the three algorithms, when applied to polychromatic Fraunhofer propagations with square arrays, are summarized in Table 1.

4. BENCHMARKS AND PERFORMANCE COMPARISON

This section shows the real-world performance of the previously discussed DFT algorithms. It is important to remember that the actual run time of any algorithm is dependent on choice of hardware, the underlying numerical libraries available in the software environment, and the details of a particular implementation. Results for other systems will vary, but with this example, we seek to show how the ideal choice of DFT algorithm can be influenced by the physical parameters of the optical simulation. The benchmarks that follow were run on an NVIDIA Tesla K20X graphics processing unit (GPU) (driver version 352.39, CUDA toolkit version 7.5), in MATLAB 2016a under Red Hat Enterprise Linux 6.6. MATLAB's parallel computing toolbox was used to execute code on the GPU. Run times were measured with MATLAB's built-in `timeit()` and `gputimeit()` functions for central processing unit (CPU) and GPU implementations, respectively, and all variables were stored in double-precision floating point.

It should also be noted again that, for a phase-retrieval application, the arrays $\hat{H}$, $\hat{a}$ [Eq. (47a)], and $\hat{b}$ [Eq. (47a)] used by the CZT and Ω_{x} and Ω_{y} used by the MTP are the same on every iteration, so they should be stored and reused. For this reason, the time taken to compute these arrays is not included in the timing measurements.

All of the benchmarks in this section were done with square arrays; that is, $R = S$, $M = S$, $Q_{\mathrm{x}}^{\mathrm{P}} = Q_{\mathrm{y}}^{\mathrm{P}} = Q^{\mathrm{P}}$, and $Q_{\mathrm{x}}^{\mathrm{I}} = Q_{\mathrm{y}}^{\mathrm{I}} = Q^{\mathrm{I}}$.

A. Monochromatic Models

We begin the section with a simple simulation of propagation for a monochromatic model. The run times for equivalent CZT, FFT, and MTP computations over a range of 2D array sizes are shown on a log scale in Fig. 2. In each trial, a single optical field represented by an array of complex numbers is propagated to the image plane with the given DFT algorithm. In this example, we set values for Q^{I} and Q^{P} to unity, so that the array sizes in the pupil and image planes are naturally the same. We can see from the figure that the two FFT-based algorithms (FFT and CZT) scale better with array size than the MTP, as expected, with the FFT being a clear winner at large sizes. At small array sizes, the curves are flat, which we interpret as the run times being dominated by fixed overhead cost of communication between the MATLAB interpreter and the GPU, rather than the numerical calculation itself. In that range, the MTP does the best. The CZT times show a "stair-step" behavior, which is a result of the way this CZT implementation chooses array sizes for its internal FFTs. Since the two internal FFTs in Eq. (51) can be padded up to any array size without regard to the physical parameters of the overall DFT, our implementation pads the data arrays to whatever size yields the best performance (based on previous benchmarks). For small ranges of PSF array sizes, there tends to be one nearby array size that has the fastest time and is chosen instead of a range of nearby sizes. Since the nearby DFTs with smaller PSF array sizes are all using the same internal FFT size, they have nearly the same run time. In this example, the FFT is the most efficient of the three in terms of handling large arrays. However,

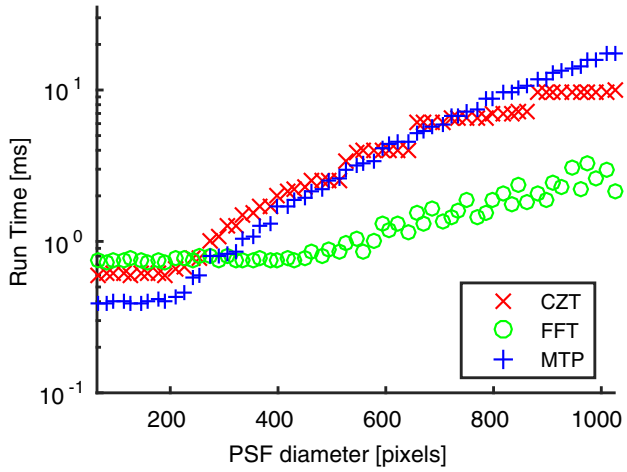

Fig. 2. Run time comparison for $Q^{\mathrm{I}} = Q^{\mathrm{P}} = 1$ as a function of PSF size R for a monochromatic simulation.

this simple case is the most advantageous case for the FFT, since the data array sizes in both domains are the same, there is only one wavelength component, and $Q^{\mathrm{P}} = Q^{\mathrm{I}} = 1$. In the following examples, we will see how the flexibility of the arbitrarily sampled methods allow them to maintain good performance with a wide variety of simulation parameters.

As discussed in Section 2.A, it is common to use $Q^{\mathrm{I}} = 2$ to reduce aliasing effects, and $Q^{\mathrm{P}} = 2$ to ensure that the intensity of the PSF is Nyquist-sampled. Figure 3 shows the timing results for this case. For a given PSF diameter, the larger Q^{P} corresponds to a smaller physical pupil size compared to the previous example. However, the increase in Q^{I} requires finer sample spacing in the pupil. Since Q^{P} and Q^{I} are both increased by a factor of 2, the net effect is that the CZT and MTP have the same cost in this case as in the previous example. The FFT, on the other hand, is unable to take advantage of the smaller physical pupil size represented by the increase in Q^{P}, since it must pad the pupil array up to the same size as the image plane array. The array size is doubled with $Q^{\mathrm{I}} = 2$, so the performance of the FFT is worsened by a factor of 4 compared to the previous case. As a result, the performance of the CZT is comparable to that of the FFT in this case.

In order to explore the dependence on Q^{P} in more detail, we now consider cases where the region of interest in the pupil plane is smaller than the overall aperture size. In particular, this is relevant to phase retrieval on optical systems with segmented apertures and to transverse translation diversity using with a subaperture mask, where subregions of the pupil will be propagated, while the detector array was designed to adequately sample a PSF from the full aperture. In these cases, the DFT period K given in Eq. (59) and region of interest in the image plane R stay the same, but a smaller region of interest in the pupil plane means propagation with a higher Q^{P}. Figure 4 shows the FFT time holding constant, since its run time does not change as Q^{P} increases: even though the number of pixels needed to represent the pupil is smaller, the pupil-plane data array is always the size of the DFT period. The run time of the CZT changes in accordance with Eqs. (52) and (54). As the number of pixels in the pupil decreases, the padded array size $K' \times L'$ may decrease

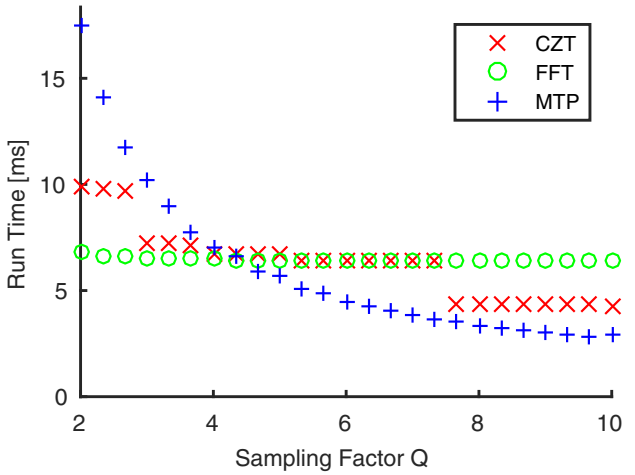

Fig. 4. Time versus Q^{P}, for PSF size of 1024 × 1024 pixels. Image size is held fixed and pupil sampling is varied to accommodate the change in Q^{P}.

until it is nearly equal to the array size R in the image plane. The MTP has a run time driven by Eq. (34), or equivalently by Eq. (40), since the arrays are square. The decreasing array size in the pupil plane allows the cost to continuously decrease, which makes the MTP the most efficient for large Q^{P}.

B. Polychromatic Models

Next, we examine performance of the three algorithms for polychromatic simulations. The first example is an optical system with a spectrum ranging from 500 nm to 1.5 μm, which is modeled at five evenly spaced discrete wavelengths. The model uses $Q^{\mathrm{I}} = Q^{\mathrm{P}} = 2$. The run times shown are the total time needed to propagate all five wavelengths. The timing results in Fig. 5 show the CZT and MTP performing similarly, as they did for the monochromatic case. The FFT, however, performs poorly for all array sizes, consistent with expectations based on Eq. (63). The FFT can only maintain consistent sample spacing for each wavelength component by padding the data arrays significantly. The largest wavelength, for example,

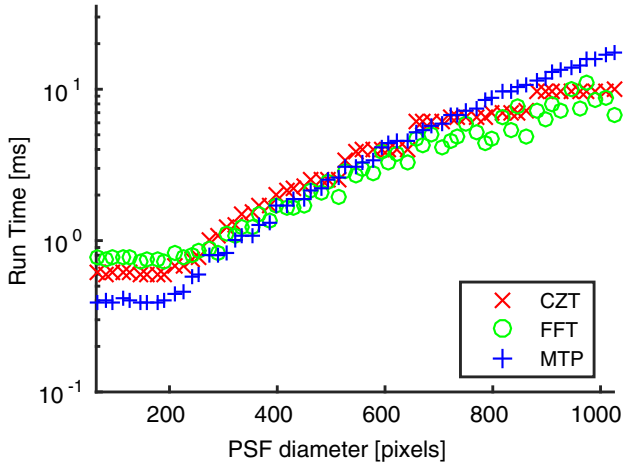

Fig. 3. Run time comparison for $Q^{\mathrm{I}} = Q^{\mathrm{P}} = 2$ as a function of PSF size R for a monochromatic simulation.

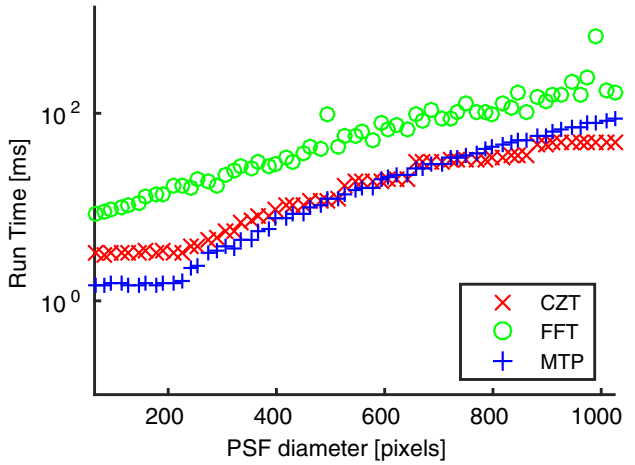

Fig. 5. Timing comparison for a broadband simulation.

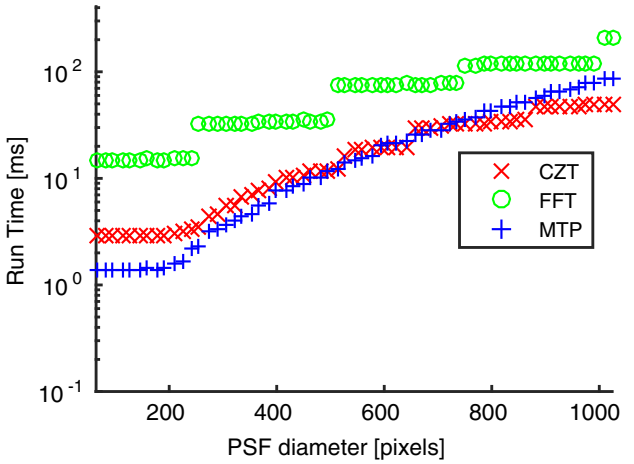

Fig. 6. Timing comparison for a narrowband polychromatic simulation.

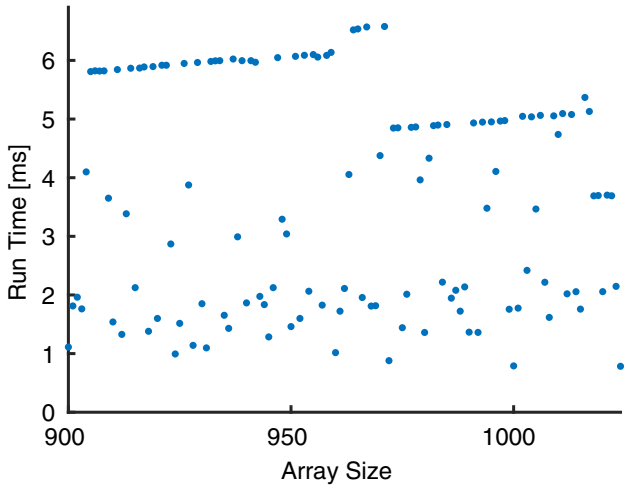

Fig. 7. 2D FFT times on the GPU, including highly noncomposite array sizes.

requires padding by a factor of 3 (since the wavelength in Eq. (62) is 3 times larger than the reference), which increases the run time for that wavelength by a factor of 9. This is more than enough to overwhelm the FFT's previous advantage over the other two methods. For example, we observe that for the 112 × 112 array size, the FFT is 6.6× slower than the MTP, and 3.1× slower than the CZT. At the 1008 × 1008 array size, the FFT is 2.1× slower than the MTP, and 3.6× slower than the CZT.

The next polychromatic simulation models a narrowband spectrum of only 10 nm wide, centered at 1 μm. Again, five evenly spaced wavelength components are propagated, and $Q^{\mathrm{I}} = Q^{\mathrm{P}} = 2$. The total run time for all five wavelengths is shown in Fig. 6. Again, the CZT and MTP take the same time to propagate each wavelength, as if the simulation were monochromatic (the total time shown on the plot includes the cost of running all five wavelengths). The FFT's performance is greatly degraded by the padding issues discussed in Section 3.C. The FFT must use a highly oversampled representation of the pupil plane, so that an integer change in the array size corresponds with the needed small fractional changes in wavelength. This is evidenced by the "stepped" nature of the curve in Fig. 6. Each step corresponds with a jump to the next smallest array size that permits integer changes to correspond with the fractional wavelength differences. As in the previous case, the padding requirements are enough to make the FFT slower than the MTP and CZT. For the 112 × 112 array size, it is 10.7× and 5.2× slower than the MTP and CZT, respectively, and at the 1008 × 1008 array size it is 2.4× slower and 4.2× slower.

C. Advantageous Array Sizes

An important factor in the CZT's good performance in polychromatic simulations is its freedom to choose advantageous array sizes without affecting the physical parameters of the simulation. To see the significance of this freedom, we can look at how the performance of an FFT implementation varies with array size.

Figure 7 shows benchmarks of the FFT for a small range of 2D square array sizes (the axis labels show the length of one side). It is well known that FFTs are more efficient for highly composite array sizes, for example [23]. The plot shows that, in this case, the difference in run time between a good, highly composite array size and the poorest performers is roughly a factor of 6. This highlights the disadvantage faced by the FFT when it is forced to use a less-than-ideal array size for some wavelengths in a broadband simulation. Additionally, we see that the performance does not follow the theoretical $N^2 \log N$ curve locally; the actual performance is very array-size- and library-dependent.

Figure 8 shows run times for the same sizes when executed on a CPU. On this hardware, the range from best to worst array sizes is comparable to the GPU, but the times are more evenly dispersed in between. The libraries available for the CPU (FFTW via MATLAB in this example) are likely to be more refined than libraries available for a GPU, which could explain how the FFT handles noncomposite array sizes more gracefully on a CPU.

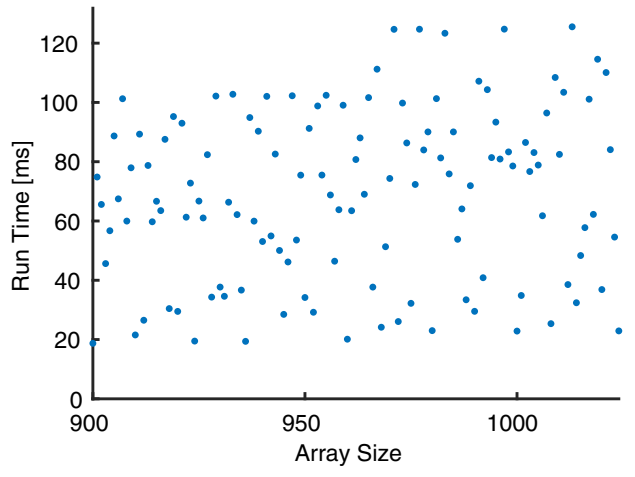

Fig. 8. 2D FFT times on the CPU (FFTW), including highly noncomposite array sizes.

In the polychromatic simulations of Section 4.B, regularly spaced wavelength components were chosen to sample the spectrum, which led to some padded array sizes for the FFT that are noncomposite. One may improve the performance of the FFT by shifting the choice of wavelengths to simulate so that the array sizes are closer to composite numbers, but then adjustments to the spectral weights are required [4]. The best possible outcome of this would be to recover the FFT's performance from the monochromatic cases, as in Fig. 2 or 3. One could also decide to allow slight errors in Q^{P} for the sake of using more desirable array sizes, but such approximations should be made with care. The arbitrarily sampled methods allow any choice of wavelengths without affecting performance or accuracy.

5. CONCLUSION

In this work, we considered in detail three methods for computing optical propagations: the commonly-used FFT, the MTP, and the CZT. We explored, both through theoretical considerations in Section 2 and benchmarks in Section 4, the computational trade-offs and sampling issues in choosing one or the other of these methods. The computational performance is more easily understood in terms of sampling factors Q^{P} and Q^{I}, from Section 3.B. Summarizing the results of these benchmarks:

- For monochromatic models with $Q^{\mathrm{P}} = Q^{\mathrm{I}} = 1$, the FFT performs best (see Fig. 2).
- For monochromatic models with $Q^{\mathrm{P}} = Q^{\mathrm{I}} = 2$, the CZT and FFT both have good performance (see Fig. 3).
- For small array sizes in general, the MTP tends to outperform the other methods, contrary to its less favorable asymptotic complexity.
- When Q^{P} is large, the CZT and MTP can substantially outperform the FFT. For very large Q^{P} with fixed PSF size, the MTP greatly outperforms the other methods (see Fig. 4).
- For polychromatic problems, the MTP and CZT generally outperform the FFT, sometimes by substantial margins, as shown in Figs. 5 and 6.
- Outside of the case where $Q^{\mathrm{P}} = Q^{\mathrm{I}} = 1$, the FFT does not have a large performance advantage over the CZT.

From these results we make some general observations: Contrary to popular practice, a padded FFT is often not the fastest choice for computing oversampled DFTs. The 2D CZT is a good candidate for a general purpose replacement for the padded FFT; it offers reasonable asymptotic complexity in all regimes and flexible control over image and pupil sampling. For small array sizes, the MTP tends to outperform the other methods.

In applications where the absolute fastest performance is required, all three methods should be benchmarked on the particular hardware and array sizes of interest; the crossover points where one method or the other offers the best performance will depend on the particular hardware and libraries in use. Getting the best possible performance out of the FFT or CZT requires careful choice of transform lengths to optimize performance of the particular FFT libraries on the available hardware.

APPENDIX A

Symbols used throughout this paper are defined in Table 2.

Table 2. Mathematical Symbols

Continuous	Discrete	Description
x, y	m, n	pupil coordinates
f_x, f_y	r, s	Fourier domain coordinates
u, v	r, s	image coordinates
$g(x, y)$	$g[m, n]$	pupil field
$G(u, v)$	$G[r, s]$	image field
D_x, D_y	M, N	pupil size
–	R, S	image size
–	K, L	lengths of DFT periods
Q^{P}	–	sampling ratio (as defined by [9])
Q^{I}	–	conjugate sampling ratio
–	$\Omega_x[r, m]$	MTP kernel (x)
–	$\Omega_y[n, s]$	MTP kernel (y)
α_x, α_y	–	CZT sampling factors
–	K', L'	FFT lengths (internal to CZT)
–	$a[r, s]$	CZT outer factor
–	$b[m, n]$	CZT inner factor
–	$h[m, n]$	CZT convolution kernel
–	$\hat{b}$, $\hat{h}$, $\hat{g}$	padded arrays for CZT
–	$\hat{H}$	Fourier transform of $\hat{h}$
f	–	focal length
λ	λ_k	wavelength
$w(\lambda)$	w_k	spectral weight
$A(x, y; \lambda)$	–	pupil amplitude
$W(x, y; \lambda)$	–	pupil wavefront
–	Q_k^{P}	sampling ratio for λ_k
$I(u, v)$	$I[r, s]$	intensity image (PSF)
t_{XYZ}	–	running time for method XYZ

Funding. Goddard Space Flight Center (GSFC); U.S. Department of Energy (DOE).

Acknowledgment. Portions of this work were presented at the OSA Computational Optical Sensing and Imaging Conference in 2014, paper CTh3C.1, "Optical Propagations with Arbitrary Sample Spacing." This research was supported by NASA Goddard Space Flight Center and a Frank Horton Research Fellowship from the Laboratory for Laser Energetics.

REFERENCES

1. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. **19**, 297–301 (1965).
2. M. Frigo and S. Johnson, "FFTW: an adaptive software architecture for the FFT," in *IEEE International Conference on Acoustics, Speech and Signal Processing* (1998).
3. M. Frigo and S. Johnson, "The design and implementation of FFTW3," Proc. IEEE **93**, 216–231 (2005).
4. J. R. Fienup, "Phase retrieval for undersampled broadband images," J. Opt. Soc. Am. A **16**, 1831–1837 (1999).

5. J. R. Fienup, "Phase retrieval with broadband light," in *Frontiers in Optics 2011/Laser Science XXVII* (Optical Society of America, 2011), paper FThD5.
6. M. D. Bergkoetter and J. R. Fienup, "Phase retrieval with linear chromatic dispersion," in *Imaging and Applied Optics* (Optical Society of America, 2016), paper CT4C.5.
7. M. D. Bergkoetter, "Phase retrieval for chromatic aberrations and wide-field detectors," Ph.D. dissertation (University of Rochester, 2017).
8. J. W. Goodman, *Introduction to Fourier Optics*, 3rd ed. (Roberts, 2005).
9. R. D. Fiete, "Image quality and λFN/p for remote sensing systems," Opt. Eng. **38**, 1229–1240 (1999).
10. R. Soummer, L. Pueyo, A. Sivaramakrishnan, and R. J. Vanderbei, "Fast computation of Lyot-style coronagraph propagation," Opt. Express **15**, 15935–15951 (2007).
11. M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," Opt. Lett. **33**, 156–158 (2008).
12. A. J. Stothers, "On the complexity of matrix multiplication," Ph.D. dissertation (University of Edinburgh, 2010).
13. N. Brenner, "Fast Fourier transform of externally stored data," IEEE Trans. Audio Electroacoust. **17**, 128–132 (1969).
14. G. Rivard, "Direct fast Fourier transform of bivariate functions," IEEE Trans. Acoust. Speech Signal Process. **25**, 250–252 (1977).
15. M. Andrews and R. E. Boring, "Architectural study of adaptive algorithms for adaptive beam communication antennas," Technical Report DTIC Document ADA206912, 1988.
16. L. Rabiner, R. Schafer, and C. Rader, "The chirp z-transform algorithm," IEEE Trans. Audio Electroacoust. **17**, 86–92 (1969).
17. L. Bluestein, "A linear filtering approach to the computation of discrete Fourier transform," IEEE Trans. Audio Electroacoust. **18**, 451–455 (1970).
18. D. H. Bailey and P. N. Swarztrauber, "The fractional Fourier transform and applications," SIAM Rev. **33**, 389–404 (1991).
19. K. Osvay, A. Kovacs, Z. Heiner, G. Kurdi, J. Klebniczki, and M. Csatari, "Angular dispersion and temporal change of femtosecond pulses from misaligned pulse compressors," IEEE J. Sel. Top. Quantum Electron. **10**, 213–220 (2004).
20. G. Pretzler, A. Kasper, and K. Witte, "Angular chirp and tilted light pulses in CPA lasers," Appl. Phys. B **70**, 1–9 (2000).
21. H.-M. Heuck, P. Neumayer, T. Kühl, and U. Wittrock, "Chromatic aberration in petawatt-class lasers," Appl. Phys. B **84**, 421–428 (2006).
22. B. E. Kruschwitz, S.-W. Bahk, J. Bromage, M. D. Moore, and D. Irwin, "Accurate target-plane focal-spot characterization in high-energy laser systems using phase retrieval," Opt. Express **20**, 20874–20883 (2012).
23. Nvidia Corporation, "CUDA toolkit documentation," http://docs.nvidia.com/cuda/cufft/#introduction.