

# Efficient high-order accurate Fresnel diffraction via areal quadrature and the nonuniform fast Fourier transform

Alex H. Barnett\*

Simons Foundation, Flatiron Institute, Center for Computational Mathematics, New York, New York, United States

**Abstract.** We present a fast algorithm for computing the diffracted field from arbitrary binary (sharp-edged) planar apertures and occulters in the scalar Fresnel approximation for up to moderately high Fresnel numbers ( $\lesssim 10^3$ ). It uses a high-order areal quadrature over the aperture and then exploits a single 2D nonuniform fast Fourier transform to evaluate rapidly at target points (on the order of  $10^7$  such points per second, independent of aperture complexity). It thus combines the high accuracy of edge integral methods with the high speed of Fourier methods. Its cost is  $\mathcal{O}(n^2 \log n)$ , where  $n$  is the linear resolution required in the source and target planes, to be compared with  $\mathcal{O}(n^3)$  for edge integral methods. In tests with several aperture shapes, this translates to between two and five orders of magnitude acceleration. In starshade modeling for exoplanet astronomy, we find that it is roughly  $10^4\times$  faster than the state-of-the-art in accurately computing the set of telescope pupil wavefronts. We provide a documented, tested MATLAB/Octave implementation. An appendix shows the mathematical equivalence of the boundary diffraction wave, angular integration, and line integral formulas and then the analysis of a new non-singular reformulation that eliminates their common difficulties near the geometric shadow edge. This supplies a robust edge integral reference against which to validate the main proposal. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JATIS.7.2.021211](https://doi.org/10.1117/1.JATIS.7.2.021211)]

**Keywords:** Fresnel; diffraction; quadrature; numerical; nonuniform fast Fourier transform; starshade.

Paper 20153SS received Oct. 12, 2020; accepted for publication Dec. 16, 2020; published online Jan. 11, 2021.

## 1 Introduction

The numerical modeling of wave diffraction from thin two-dimensional (2D) screens and apertures in the Fresnel regime has many applications in optics<sup>1</sup> and acoustics (Ref. 2, Sec. 8.4), including instrument modeling,<sup>3,4</sup> lithography mask design,<sup>5</sup> Fourier optics,<sup>6</sup> coherent x-rays,<sup>7</sup> acoustic emission,<sup>8</sup> computer-generated binary holograms,<sup>9</sup> starshades,<sup>10</sup> and Fresnel zone plate imagers.<sup>11,12</sup> This usually involves a plane or spherical wavefront hitting a binary (“0–1”) mask of given shape, although continuous opacity/phase variation is also possible. We confine ourselves to the former case, although the method can trivially accommodate the latter. Our point is to present a simple—yet seemingly overlooked—method that renders a high-accuracy numerical modeling at moderate Fresnel numbers orders of magnitude more efficient than prior methods.

One motivation is starshade design.<sup>10,13–17</sup> Given a space telescope, the goal is that a distant binary occulter blocks the direct light from a star, allowing for much dimmer exoplanets separated from it by only tens of milli-arcseconds to be imaged. The occulter shape and distance are thus optimized to give a deep shadow region, with a relative intensity on the order of  $10^{-10}$  across the telescope pupil, throughout a given wavelength range, while minimizing the occulter’s physical size (for practical reasons) and the angular size at the telescope. This has led to shapes with “petals” that emulate a continuous radial apodization, with radii on the order of 10 m and distances on the order of  $10^7$  m. Here the scalar<sup>18</sup> and Fresnel approximations are superb (Ref. 17,

\*Address all correspondence to Alex H. Barnett, [abarnett@flatironinstitute.org](mailto:abarnett@flatironinstitute.org)

App. A), with a small Fresnel number [defined in Eq. (2)] of typically 5 to 20. Numerical modeling is challenging, demanding at least 6-digit accuracy in amplitude to validate the shadow and many runs with different wavelengths and shapes to assess mechanical and thermal stability.<sup>16</sup>

Fixing a wavelength  $\lambda$  and propagation distance  $z$ , a point in the aperture (or occulter) plane is  $(x, y, 0)$  and a target point in the detector (or pupil) plane is  $(\xi, \eta, z)$  (see Fig. 1). We drop the constant  $z$ -coordinates hereafter, so the problem is in essence 2D. In the case of a unit amplitude plane wave with wavevector  $(0, 0, 2\pi/\lambda)$  incident on a planar aperture  $\Omega \subset \mathbb{R}^2$ , the mathematical task is to evaluate the Fresnel integral for the scalar potential:

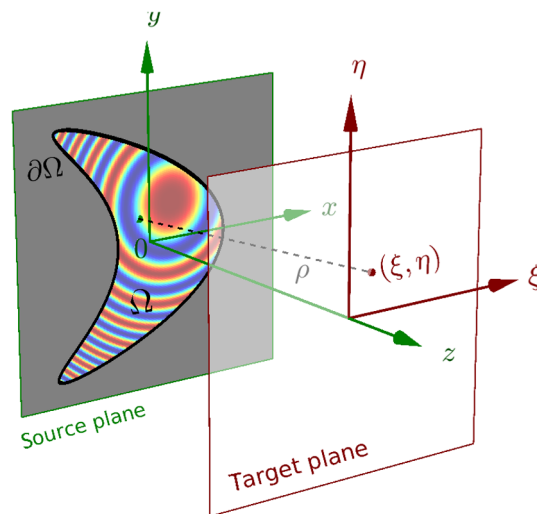
$$u^{\text{ap}}(\xi, \eta) = \frac{1}{i\lambda z} \iint_{\Omega} e^{\frac{i\pi}{\lambda z}[(\xi-x)^2 + (\eta-y)^2]} dx dy. \quad (1)$$

This takes the form of a 2D convolution of the aperture's characteristic function  $\chi_{\Omega}$  with a radially symmetric kernel (a complex Gaussian), with half-wave oscillation regions that are commonly called “zones” (Fig. 1). If  $R$  is an effective (or maximum) radius of  $\Omega$ , then the in-plane separation  $r := \sqrt{(\xi - x)^2 + (\eta - y)^2}$  is typically bounded by  $R$  times a small constant. Thus the number of zones inside  $\Omega$  is on the order of

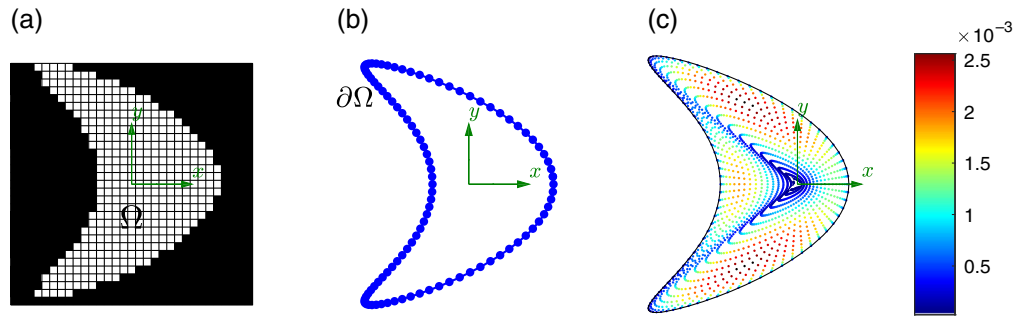
$$\mathfrak{f} := \frac{R^2}{\lambda z} \quad (\text{Fresnel number}), \quad (2)$$

and the finest oscillation scale of the integrand is  $\mathcal{O}(1/\mathfrak{f})$ . In Eq. (1), the prefactor  $1/i\lambda z$  ensures that  $u$  tends to unity in the limit of a large aperture, i.e., the unimpeded wave.

It is worth reviewing the origin of Eq. (1). It arises from the Kirchhoff diffraction approximation to the full Maxwell equations. This is good when aperture features are much larger than  $\lambda$  (Ref. 6, Ch. 3). Only the zeroth and first term in the Taylor expansion of the exponent in the free-space Green's function  $e^{2\pi i \rho/\lambda}/\rho$  are then kept,  $\rho = \sqrt{r^2 + z^2}$  being the source-target distance (dotted line in Fig. 1). The Fresnel approximation is thus valid to the extent that the next term is small, implying the condition  $r^4 \ll \lambda z^3$ . The denominator of the Green's function is approximated by  $z$ . We refer the reader to (Ref. 1, Sec. 8.3.3) (Ref. 6, Ch. 4) for details. Note that the zeroth term gave the plane propagation phase  $e^{2\pi i z/\lambda}$ , which is usually included as a prefactor in Eq. (1). We drop it for simplicity; it is trivial to insert. By replacing  $z^{-1}$  by  $z^{-1} + D^{-1}$ , Eq. (1) also applies when a point source at finite distance  $D$  produces a spherical incident wave.<sup>1,17,19</sup>



**Fig. 1** Geometry and notation for Fresnel diffraction, in the case of  $\Omega$  an aperture with boundary  $\partial\Omega$ . A plane wave is incident from behind, along the  $z$  axis. For the target  $(\xi, \eta)$ , the real part of the integrand in Eq. (1) is imaged in color (red positive, blue negative, and green around zero; each red or blue annular region is a Fresnel zone).



**Fig. 2** Sketch of three alternative methods for discretization in the source (aperture) plane, shown for a kite-shaped aperture  $\Omega$  with smooth boundary  $\partial\Omega$ . (a) Uniform 2D grid sampling; (b) line integral quadrature; and (c) high-order areal quadrature. (a) and (b) are well established. Our method is (c): areal nodes  $(x_j, y_j)$  are shown with their color indicating the weights  $w_j$  according to the scale at the right.

In the plane wave case, by Babinet’s principle (Ref. 1, Sec. 8.3.2) (Ref. 14, Sec. 2.2), the potential when  $\Omega$  defines an occulter rather than an aperture is simply given by

$$u^{\text{oc}}(\xi, \eta) = 1 - u^{\text{ap}}(\xi, \eta). \quad (3)$$

Analytical forms for the integral Eq. (1) are known only for the straight edge, infinite slit, and rectangles (all involving the same 1D special function) (Ref. 1, Sec. 8.7) (Ref. 6, Sec. 4.5.1) and the disc.<sup>20–22</sup> Semianalytical Bessel expansions can be useful for symmetric starshade design.<sup>10,14</sup> But for general shapes, one is left with fully numerical methods, which fall into two main categories [sketched in Figs. 2(a) and 2(b)]:

- (a) *2D fast Fourier transform (FFT) methods.* There are two types: (i) methods implementing the convolution theorem (forward then backward FFTs), useful only for large  $\mathfrak{f}$  (near-field), or (ii) methods exploiting the quadratic form in Eq. (1) via a single FFT,<sup>11,13,23</sup> useful from zero to moderate  $\mathfrak{f}$ . See Ref. 24 for a review, in which fractional FFTs are also considered. The FFT of course requires only  $\mathcal{O}(n^2 \log n)$  operations to transform a  $n \times n$  source to the target grid. However, the aperture or occulter must be sampled (quantized) on the source grid, and if this is done in a binary fashion<sup>11</sup> [as in Fig. 2(a)], the error convergence rate is a very low order, no faster than  $\mathcal{O}(1/n)$ . This is inadequate for starshade shadow modeling.<sup>14,15</sup> Subpixel averaging can improve accuracy,<sup>3,17</sup> but this can give at best<sup>25</sup>  $\mathcal{O}(1/n^2)$ , and for starshades, huge ( $n > 10^5$ ) sized FFTs are still needed to reach the required accuracy.<sup>17</sup> The underlying problem is that  $\chi_\Omega$  is not a bandlimited function, so it is always poorly represented on regular grids.
- (b) *Edge integral methods.* Such methods discretize a target-dependent integral over the aperture/occulter boundary  $\partial\Omega$ . The literature splits these methods into two formulas: (i) the “boundary diffraction wave” (BDW) of Miyamoto–Wolf,<sup>26</sup> arising from the Kirchhoff approximation, versus (ii) reduction of Eq. (1) to a 1D angular integral, a result of Dauger<sup>19</sup> and Dubra–Ferrari.<sup>22</sup> To resolve the Fresnel integrand, the number of discretization nodes must scale as  $n = \mathcal{O}(\mathfrak{f})$ , so the cost for (direct) evaluation on a resolved  $n \times n$  target grid is  $\mathcal{O}(n^3) = \mathcal{O}(\mathfrak{f}^3)$ . Both formulas have been applied to state-of-the-art starshade modeling: Cady<sup>15,27</sup> uses (i), while Cash<sup>14</sup> and Harness et al.<sup>17</sup> use (ii). Here second-order, i.e.,  $\mathcal{O}(1/n^2)$  accurate, midpoint quadrature rules are used, with up to about  $10^5$  nodes, to reach the needed 6-digit accuracy.

**Remark 1.** Such edge integral methods should not be confused with the (more involved) boundary integral equation method, which solves the 3D Helmholtz or Maxwell equations using surface unknowns and potential theory.<sup>28,29</sup>

Our proposal combines the best features of the above two categories, namely the speed of the FFT methods with the high (and potentially high order) accuracy of edge integral methods for binary apertures. The result is an acceleration over edge integral methods of between two and five orders of magnitude. In a nutshell, the idea is, realizing that a 2D regular grid guarantees

poor quadrature for integrals over  $\Omega$ , to replace it with a much better (high-order) areal quadrature scheme [see Fig. 2(c)]. Then exploiting the quadratic form in Eq. (1), as in “single FFT” methods (a)(ii), leaves one remaining problem: how to rapidly evaluate Fourier sums involving off-grid frequencies. Fortunately, fast algorithms for this task—nonuniform FFTs<sup>30,31</sup>—are quite mature and have speeds only one order of magnitude below those of plain FFTs.

High accuracy relies on constructing a good areal quadrature for  $\Omega$ , which depends on its geometric description. We show that such quadratures can easily piggyback off boundary quadratures or be built independently. Our proposal is in some way related to diffraction methods that subsample the FFT<sup>32</sup> or use chirp FFTs;<sup>4</sup> however, it is much simpler and more general than either.

Turning to the structure of this paper, Sec. 2 explains the method for arbitrary targets (Sec. 2.1) and then for gridded targets (Sec. 2.2), the latter being somewhat faster. In Sec. 3, we demonstrate the high accuracy and efficiency of the method for a smooth occulter (Sec. 3.1), two symmetric starshades (Sec. 3.2), and an aperture built from 67 million triangles (Sec. 3.3). We compare the method with the performance of a BDW edge integral code.<sup>15,27</sup> We draw conclusions, explain how the method can be applied to perturbed starshades, and propose extensions in Sec. 4.

Finally, validating the proposed nonuniform fast Fourier transform (NUFFT) method down to errors of  $10^{-12}$  or better demanded a high-accuracy reference edge integral method, which required new research, to which the Appendix is devoted. There we clarify that the BDW edge-integral (b)(i) and angular-integration (b)(ii) methods are equivalent and are equivalent to a more convenient line integral due to Cash.<sup>14</sup> However, as we show, existing edge integral methods suffer numerical breakdown as targets approach  $\partial\Omega$  (the geometric shadow edge) because they represent the (smooth) diffracted field as a sum of two discontinuous terms, one with a singular integrand. We instead present and analyze a simple, robust non-singular line integral (NSLI) that maintains close to machine accuracy for target points near or even on  $\partial\Omega$ , without extra work, yet takes only five lines to code.

**Remark 2.** We maintain a documented, tested, open-source MATLAB/Octave implementation of the proposed fast algorithm (using the FINUFFT library<sup>33</sup>), and the NSLI, on GitHub.<sup>34</sup> Queries to the author are welcome.

## 2 Proposed Method

Given a set of targets  $(\xi_k, \eta_k)$ ,  $k = 1, \dots, M$ , recall that the goal is to approximate Eq. (1) efficiently, i.e., to evaluate

$$u_k^{\text{ap}} := u^{\text{ap}}(\xi_k, \eta_k) = \frac{1}{i\lambda z} \iint_{\Omega} e^{\frac{i\pi}{\lambda z}[(\xi_k - x)^2 + (\eta_k - y)^2]} dx dy, \quad k = 1, \dots, M. \quad (4)$$

Suppose that an areal quadrature rule for the aperture  $\Omega$  has been found, that is, a set of nodes  $(x_j, y_j) \in \mathbb{R}^2$  and weights  $w_j$ ,  $j = 1, \dots, N$ , such that, for all sufficiently smooth functions  $f$ ,

$$\iint_{\Omega} f(x, y) dx dy \approx \sum_{j=1}^N f(x_j, y_j) w_j \quad (5)$$

holds to high accuracy. Specifically, one seeks a family of rules of increasing  $N$ , with a high order of convergence  $p$ , meaning that, for each  $C^\infty$ -smooth  $f$ , the error (difference between left and right sides) is  $\mathcal{O}(N^{-p})$ . This may even hold for all  $p > 0$ , in which case the convergence is said to be super-algebraic or spectral. For instance, such a quadrature over a rectangle is given by a tensor product of 1D Gauss–Legendre rules (Ref. 35, Ch. 19). By passing those nodes through a smooth mapping of  $\mathbb{R}^2$  to  $\mathbb{R}^2$  (and multiplying the weights by its Jacobean), such rules for triangles and distorted quadrilaterals are easily made, the union of which can approximate any domain with a piecewise smooth boundary to a high order. For more node-efficient quadratures, we refer the reader to recent works on triangles,<sup>36</sup> polygons,<sup>37,38</sup> and domains defined by piecewise rational curves.<sup>39</sup> Converting a general boundary into an areal quadrature for its interior is a

software engineering task beyond the scope of this paper. We will be content constructing areal quadratures for three types of domains: the interior of a smooth closed curve, symmetric star-shades, and unions of triangles.

Since any rule Eq. (5) must resolve the Fresnel integrand oscillations, for a fixed  $\Omega$ , its number of nodes must grow like

$$N = \mathcal{O}(f^2). \quad (6)$$

Now simply applying Eq. (5) to Eq. (4) gives a high-order accurate direct (slow) summation method that, as the results below show, can exceed accuracy requirements with reasonable numbers of nodes. (Our results show that claims such as “a single point in the shadow plane can require a trillion sine calculations at quadruple precision”<sup>14</sup> are overly pessimistic.) The cost of this direct sum to  $M$  targets would be  $\mathcal{O}(NM)$ . Our goal is now to reduce this to close to  $\mathcal{O}(N + M)$  via fast Fourier methods.

## 2.1 Fast Algorithm for Diffraction to Arbitrary Target Points

We apply Eq. (5) to Eq. (4) and then in the second line expand each quadratic term. (This trick, common to “single FFT” methods,<sup>11,13,23</sup> is similar to, but essentially the reverse of, that in the Bluestein method.<sup>40</sup>) This gives

$$u_k^{\text{ap}} \approx \frac{1}{i\lambda z} \sum_{j=1}^N e^{\frac{i\pi}{\lambda z}[(\xi_k - x_j)^2 + (\eta_k - y_j)^2]} w_j \quad (7)$$

$$= \frac{1}{i\lambda z} e^{\frac{i\pi}{\lambda z}(\xi_k^2 + \eta_k^2)} \cdot \sum_{j=1}^N e^{-\frac{2i\pi}{\lambda z}(\xi_k x_j + \eta_k y_j)} [e^{\frac{i\pi}{\lambda z}(x_j^2 + y_j^2)} w_j]. \quad (8)$$

This factorized form allows for a three-step “fast” (in the sense of quasi-optimal scaling) algorithm:

1. Compute all “strengths”  $c_j$  according to the following, which takes  $\mathcal{O}(N)$  effort:

$$c_j := e^{\frac{i\pi}{\lambda z}(x_j^2 + y_j^2)} w_j, \quad j = 1, \dots, N.$$

2. Evaluate

$$v_k = \sum_{j=1}^N e^{-\frac{2i\pi}{\lambda z}(\xi_k x_j + \eta_k y_j)} c_j, \quad k = 1, \dots, M, \quad (9)$$

which is precisely the task performed by the so-called 2D NUFFT of type 3,<sup>30,31</sup> a well-established algorithm with modern software implementations.<sup>33,41</sup> This takes  $\mathcal{O}(N + M + q^2 \log q)$  effort, where  $q = \mathcal{O}(f)$  is the largest magnitude of the exponent in Eq. (9).

3. Postmultiply all outputs by their quadratic phases, which takes  $\mathcal{O}(M)$  effort:

$$u_k^{\text{ap}} = \frac{1}{i\lambda z} e^{\frac{i\pi}{\lambda z}(\xi_k^2 + \eta_k^2)} v_k, \quad k = 1, \dots, M.$$

The overall cost is thus  $\mathcal{O}(N + f^2 \log f + M)$ . Recalling Eq. (6), this is  $\mathcal{O}(f^2 \log f + M)$ . The NUFFT requires a user-chosen error tolerance  $\varepsilon$ , which affects the prefactor of this run time scaling, but rather weakly.<sup>33</sup>

## 2.2 Fast Algorithm for Target Points Lying on a Grid

Often sampling the diffracted wave on a dense regular Cartesian grid is sufficient. In this special case, more speed can be gained. Specifically, let  $(\xi_k, \eta_k)$  be the 2D grid points defined by the product of  $n$ -point regular 1D grids

$$\{-nh/2, (-n/2 + 1)h, \dots, -h, 0, h, \dots, (n/2 - 1)h\}$$

in the  $\xi$  and  $\eta$  directions, with  $h$  being the grid spacing. We assume that  $n$  is even. This grid has  $M = n^2$  targets and (ignoring its left-most column and bottom row) is centered on the origin. If we relabel the grid points as  $(hk_1, hk_2)$  for integer indices  $-n/2 \leq k_1, k_2 < n/2$  and define rescaled source points  $\tilde{x}_j := (2\pi h/\lambda z)x_j$  and  $\tilde{y}_j := (2\pi h/\lambda z)y_j$ , the middle step Eq. (9) can be written

$$v_{k_1, k_2} = \sum_{j=1}^N e^{-i(k_1 \tilde{x}_j + k_2 \tilde{y}_j)} c_j, \quad -n/2 \leq k_1, k_2 < n/2,$$

which is precisely the so-called type 1 NUFFT<sup>30</sup> (also known as the ‘‘adjoint NFFT’’<sup>41</sup>). Its non-uniform points  $(\tilde{x}_j, \tilde{y}_j)$  are only defined modulo  $2\pi$  and may need to be ‘‘folded’’ back into a valid input domain such as  $[-\pi, \pi)^2$ . If the grid width is similar in size to  $\Omega$ , then it is easy to check that such a folding is only needed if  $n$  is less than the order of  $\mathfrak{f}$ , i.e., the target grid under-resolves the diffracted field  $u$ . This is probably not a common use case.

The total cost for the regular grid case is  $\mathcal{O}(N + M \log M)$ , which, recalling Eq. (6) and  $n = \mathcal{O}(\mathfrak{f})$ , is  $\mathcal{O}(\mathfrak{f}^2 \log \mathfrak{f})$ . In practice, we find that, for the same number  $M$  of targets spanning the same domain and the same tolerance  $\varepsilon$ , this regular grid version is around four times faster than the above arbitrary target version because the type 1 NUFFT is faster than the type 3 for the same space-bandwidth product.<sup>33</sup> The generalization to rectangular target grids with arbitrary translations is simple, achieved through prephasing in step 1, and we will not present it here.

### 3 Performance Tests and Results

We now test the accuracy and speed of the above method in several geometries and compare it with two edge integral methods. All codes are written in MATLAB R2017a, apart from FINUFFT, which is a parallel C++ library with MEX interface. All timings are reported for double-precision arithmetic on a 4-core i7-7700HQ laptop with 32 GB RAM, using 8 threads (full hyperthreading). All codes take advantage of the multiple threads, either via vectorized MATLAB statements or via OpenMP in FINUFFT. In each geometry, we first need to describe the areal quadrature used.

#### 3.1 Domains Defined by a Simple Smooth Closed Curve

We start by setting up a simple areal quadrature for the interior of a simple smooth closed curve. Suppose that we have good quadrature nodes  $(X_i, Y_i) \in \partial\Omega$  and weights  $(W_i, V_i)$ , indexed by  $i = 1, \dots, n$ , for vector line integrals on  $\partial\Omega$ , that is, for all sufficiently smooth vector-valued functions  $\mathbf{f}$  on  $\partial\Omega$ :

$$\int_{\partial\Omega} \mathbf{f}(x, y) \cdot d\mathbf{s} \approx \sum_{i=1}^n \mathbf{f}(X_i, Y_i) \cdot (W_i, V_i), \quad (10)$$

where  $d\mathbf{s}$  is the counter-clockwise vector line element on  $\partial\Omega$ . Now fix a number of ‘‘radial’’ nodes  $m$ , and let  $\{\alpha_l\}_{l=1}^m$  be the Gauss–Legendre (Ref. 35, Ch. 19) nodes and  $\{\tilde{w}_l\}_{l=1}^m$  be their weights for the interval  $(0, 1)$ . The integral over  $\Omega$  may be rewritten using a ‘‘dilation’’ parameterization  $\iint_{\Omega} f d\mathbf{x} dy = \int_{\partial\Omega} (\int_0^1 f(\alpha\mathbf{x}) \alpha d\alpha) \mathbf{x} \times d\mathbf{s}$ , where  $\mathbf{x} \in \partial\Omega$  and the 2D cross product is understood to give a scalar. Applying the  $m$ -node rule on  $(0, 1)$  to the inner integral and Eq. (10) to the outer integral gives the ‘‘tensor product’’ areal quadrature for  $\Omega$ , with  $N = nm$  nodes  $(x_j, y_j)$  and weights  $w_j$  given by

$$[x_{l+(i-1)m}, y_{l+(i-1)m}] = (\alpha_l X_i, \alpha_l Y_i), \quad \text{for } i = 1, \dots, n, \quad l = 1, \dots, m, \quad (11)$$

$$w_{l+(i-1)m} = \alpha_l \tilde{w}_l (X_i V_i - Y_i W_i), \quad \text{for } i = 1, \dots, n, \quad l = 1, \dots, m. \quad (12)$$

These nodes lie along “spokes” connecting the origin to the boundary nodes, as in Fig. 2(c). If  $\Omega$  is not star-shaped about the origin, then some of the nodes lie outside  $\Omega$  and some  $w_j$  are negative; however, we observe little loss of accuracy unless  $\Omega$  is highly non-convex or poorly centered on the origin.

**Remark 3.** The above “dilation” method automatically creates an areal quadrature for  $\Omega$  given only a (vector) line integral quadrature for  $\partial\Omega$  and the convergence parameter  $m$ . It thus also applies to boundaries with corners or cusps, including non-symmetric starshades. In practice, its construction time is negligible compared with that of the proposed NUFFT algorithm.

Finally, we must build a vector line integral rule Eq. (10). The simplest case is when  $\partial\Omega$  is parameterized in a counter-clockwise sense over  $t \in [0, 2\pi)$  by a smooth  $2\pi$ -periodic vector function  $\mathbf{x}(t) := [X(t), Y(t)]$ . Then

$$\int_{\partial\Omega} \mathbf{f} \cdot d\mathbf{s} = \int_0^{2\pi} \mathbf{f}(\mathbf{x}(t)) \cdot \mathbf{x}'(t) dt \approx \sum_{i=1}^n \mathbf{f}(\mathbf{x}(2\pi i/n)) \cdot \frac{2\pi}{n} \mathbf{x}'(2\pi i/n), \quad (13)$$

where  $\mathbf{x}'(t) := d\mathbf{x}/dt$  and we apply the  $n$ -point periodic trapezoid rule quadrature with nodes  $t = 2\pi i/n$  and equal weights  $2\pi/n$ . Comparing the right sides of Eqs. (10) and (13), one reads

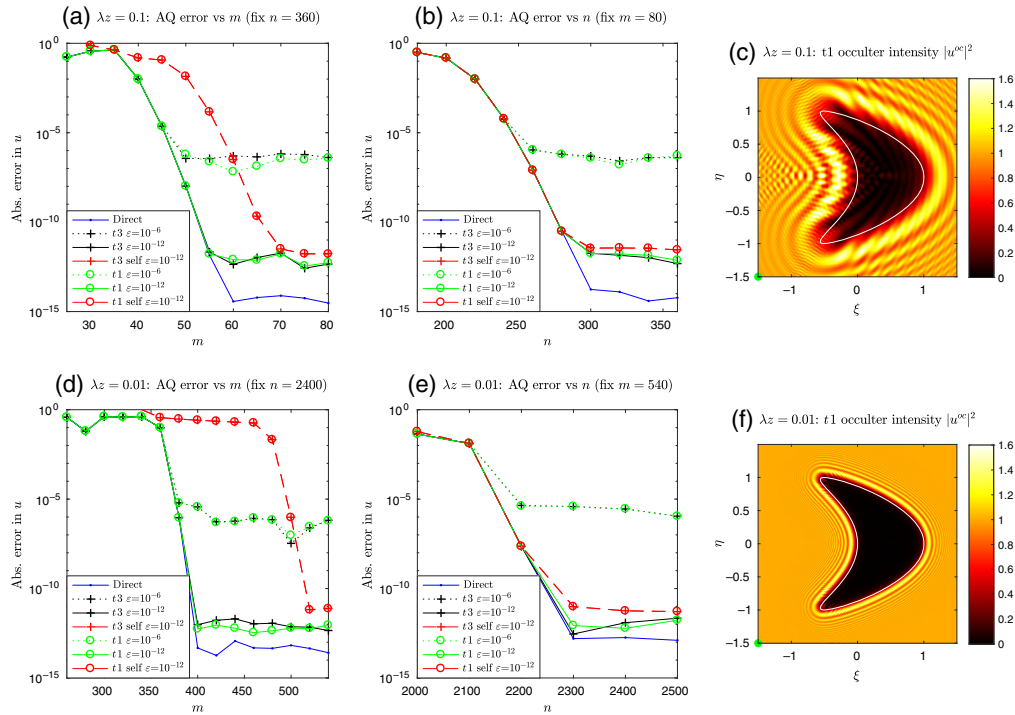
$$\begin{aligned} X_i &= X(2\pi i/n), & Y_i &= Y(2\pi i/n), & W_i &= \frac{2\pi}{n} X'(2\pi i/n), & V_i &= \frac{2\pi}{n} Y'(2\pi i/n), \\ & & & & i &= 1, \dots, n. \end{aligned} \quad (14)$$

We apply the above to build a family of areal quadratures for the kite domain with a smooth boundary  $[X(t), Y(t)] = (0.5 \cos t + 0.5 \cos 2t, \sin t)$ , shown in Fig. 2. Its maximum radius is  $R \approx 1.13$ . We then show multiple types of error convergence for the proposed method for Fresnel diffraction in Fig. 3. The graphs not labeled “self” (i.e., blue, black, and green) show absolute error in  $u^{\text{ap}}$  at a single point, using as a reference solution the (converged) NSLI edge integral method in the Appendix. The other self convergence (red) graphs show the maximum error in  $u^{\text{ap}}$  over  $10^6$  targets, using the converged values themselves as a reference. The “direct” (blue) graph simply sums Eq. (7) without the fast algorithm. The other graphs test two types of the proposed NUFFT method—arbitrary targets ( $t3$ , + signs) and gridded targets ( $t1$ ,  $\circ$  signs)—each at two different requested tolerances (6-digit and 12-digit).

In each row of panels, the first shows convergence in  $m$  (“radial” nodes), with fixed  $n$  (boundary nodes), and the second vice versa. The right-most panels show, on a  $M = 1000^2$  point grid, the converged intensity  $|u^{\text{oc}}|^2$ , applying Eq. (3) so that  $\Omega$  is an occulter. There are several observations as follows.

- In all cases, the convergence is very sudden, as is typical with a spectrally accurate quadrature rule applied to an oscillatory integrand.
- Comparing the top and bottom rows, where  $f$  has increased by a factor 10, the converged  $m$  and  $n$  have each become about 10 times larger. This matches Eq. (6).
- The “direct” application of Eq. (7) converges to 13 to 14 digits, while the NUFFT methods convergence bottoms out at within 1 digit of the requested  $\varepsilon$ , as expected.
- The self convergence (red graphs) in  $n$  [panels (b) and (e)] occurs in tandem with the independent error at the single test point  $(-3/2, -3/2)$ . However, for  $m$  [panels (a) and (d)] this is not quite true, even though the test point is the point in the target domain  $[-3/2, 3/2]^2$  maximizing the maximum source-target separation  $r$ . This is due to particulars of the angular variation in node density and reminds one that convergence must be tested at all target points.

Table 1 reports CPU timings and errors for these same tasks at converged  $n$  and  $m$  quadrature parameters. (The areal quadrature generation is not included as it was found to be negligible.) NSLI (known to achieve 13 to 14 digits with these parameters) is used as the reference for all errors. A state-of-the-art edge integral code BDWF<sup>15</sup> (as available in SISTER<sup>27</sup> and documented in Ref. 34) is also tested with the same  $n$ : while its median errors are as expected from its use of a second-order accurate midpoint rule, its maximum errors are larger than 1. These huge errors



**Fig. 3** Convergence the proposed method for the smooth kite domain (see Sec. 3.1), with respect to  $m$  (“radial” nodes) and  $n$  (boundary nodes). Unless labeled self, all errors are measured at the target  $(-3/2, -3/2)$ , shown as a green dot in the right-most plots, relative to the reference line integral method (NSLI). Those labeled self give the maximum error over  $M = 10^6$  target points relative to their converged values. “direct” (blue) uses (7); “t3” (+ signs) uses the arbitrary target type 3 NUFFT of Sec. 2.1 (random points lying in  $[-3/2, 3/2]^2$ ), and “t1” (o signs) uses the gridded target type 1 NUFFT of Sec. 2.2 (grid of  $n = 10^3$ ,  $nh = 3$ ). Two NUFFFT tolerances  $\epsilon = 10^{-6}$  and  $\epsilon = 10^{-12}$  are compared. The occulter Fresnel number for (a)–(c) is  $\bar{f} \approx 12.8$  and (d)–(f) is  $\bar{f} \approx 128$ .

appear to be confined to a few target points very near  $\partial\Omega$ , the geometric shadow edge. The main conclusions from Table 1 are as follows.

- In this setting, the proposed NUFFT-based methods are 100 to 300 times faster than the edge integral methods (for arbitrary targets) or 400 to 600 times faster (for gridded targets).
- The proposed methods robustly (uniformly at all targets) achieve close to the requested error.
- At the largest  $\lambda z$  ( $\bar{f} \approx 12.8$ ), the t1 achieves  $\sim 10^7$  targets/s, and the t3 achieves  $\sim 4 \times 10^6$  targets/s, with only a weak dependence on tolerance. These are only slightly slower for  $\bar{f}$  10 times larger.
- At the smallest  $\lambda z = 10^{-3}$  ( $\bar{f} \approx 1280$ ), the asymptotic  $\mathcal{O}(\bar{f}^2 \log \bar{f})$  cost has started to dominate. Note that 134 million nodes are being mapped to 10 million targets in around 10 s.

Since under the hood, the NUFFT uses FFTs, the reader might worry about their RAM usage. It is very mild: for t1 (gridded) cases, the FFT size is 5/4 times (for  $\epsilon \geq 10^{-9}$ , or twice otherwise) the requested grid size in each dimension. For t3, FFT dimensions scale like  $\bar{f}$ : for the smallest  $\bar{f} \approx 12.8$ , the FFT is a tiny  $216 \times 288$ . This is to be compared with the  $32768 \times 32768$  FFT needed for a subpixel sampling method to reach around 6-digit accuracy in  $u$  in various tests<sup>17</sup> at similar  $\bar{f}$ . In the penultimate row of the table  $\bar{f}$  is  $10^2$  times larger, yet the FFT is only  $7500 \times 9600$  (similar to the maximum  $r^2/\lambda z \approx 8900$  zones), and total RAM usage is 17 GB, about the largest the laptop can handle.

Another natural question is: does the method suffer at smaller  $\bar{f}$  than tested above? It does not: node numbers and CPU times only get smaller. In fact, by expanding the target grid in proportion



**Table 1** Absolute error in  $u^{\text{OC}}$  and run times of several methods for the smooth kite occulter with converged quadrature parameters. The domain, upper two  $\lambda z$  choices, and target region are as in Fig. 3. Blank entries in this table to be taken as repeated from above and “—” indicates not applicable. In both random and grid cases, errors are measured relative to the NSLI reference method (see Appendix). Timings for NSLI and BDWF<sup>15</sup> are not listed for the grid cases since they are identical to the random cases. The last two rows are close to the largest Fresnel number  $f$  that the laptop can handle (errors were checked at only  $10^4$  targets in those cases). See Sec. 3.1 for other details.

$\lambda z$	$f$	$n$ (bdry)	$m$ (radial)	$M$ (targets)	Method	Median err	Max err	CPU time (s)		
0.1	12.8	320	—	$10^6$ , random	NSLI	—	—	25.8		
					BDWF	$2.1 \times 10^{-3}$	$1.9 \times 10^1$	35.2		
					NUFFT $t3$ ( $\epsilon = 10^{-6}$ )	$6.2 \times 10^{-8}$	$1.0 \times 10^{-6}$	0.23		
				320	80	$10^6$ , grid	NUFFT $t3$ ( $\epsilon = 10^{-12}$ )	$3.3 \times 10^{-13}$	$2.8 \times 10^{-12}$	0.32
							NUFFT $t1$ ( $\epsilon = 10^{-6}$ )	$9.6 \times 10^{-9}$	$8.0 \times 10^{-7}$	0.06
							NUFFT $t1$ ( $\epsilon = 10^{-12}$ )	$1.4 \times 10^{-13}$	$2.6 \times 10^{-12}$	0.12
0.01	128	2400	—	$10^6$ , random	NSLI	—	—	79		
					BDWF	$1.1 \times 10^{-5}$	$1.1 \times 10^1$	115		
					NUFFT $t3$ ( $\epsilon = 10^{-6}$ )	$9.3 \times 10^{-8}$	$4.7 \times 10^{-6}$	0.29		
				2400	560	$10^6$ , grid	NUFFT $t3$ ( $\epsilon = 10^{-12}$ )	$3.4 \times 10^{-13}$	$9.5 \times 10^{-12}$	0.51
							NUFFT $t1$ ( $\epsilon = 10^{-6}$ )	$2.1 \times 10^{-8}$	$4.6 \times 10^{-6}$	0.17
							NUFFT $t1$ ( $\epsilon = 10^{-12}$ )	$1.8 \times 10^{-13}$	$9.6 \times 10^{-12}$	0.21
0.001	1280	24,000	5600	$10^7$ , random	NUFFT $t3$ ( $\epsilon = 10^{-6}$ )	$5.2 \times 10^{-8}$	$5.3 \times 10^{-6}$	16.2		
				$10^7$ , grid	NUFFT $t1$ ( $\epsilon = 10^{-6}$ )	$1.9 \times 10^{-8}$	$3.7 \times 10^{-6}$	10.4		

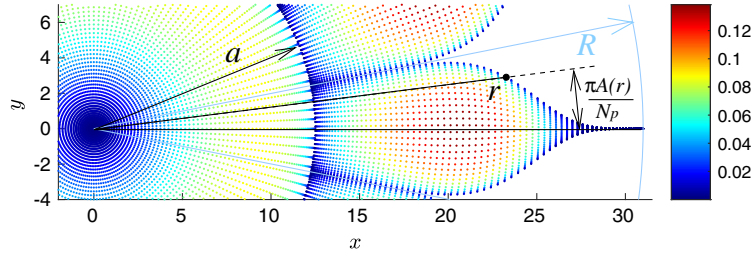
to  $z \rightarrow \infty$ , the Fraunhofer limit is reached in a stable fashion [here the first factor in Eq. (8) should be discarded, whereas the third factor tends to unity].

### 3.2 Application to Starshade Modeling

Idealized starshades are described<sup>10,14,15</sup> by a radial apodization function  $A(r)$ , where (in this section only) we use  $(r, \theta)$  as occulter-plane polar coordinates about the origin.  $A(r)$  is 1 (indicating a fully blocking disc) for  $r < a$  and drops in a carefully optimized fashion in  $r \in [a, R]$  to close to zero at  $R$ , the maximum occulter radius, and identically 0 beyond this. Apodization over  $[a, R]$  is realized via  $N_p$  identical binary petals, each of which has an angular width at radius  $r$  of  $2\pi A(r)/N_p$ . Let the function  $P(\alpha)$  denote  $\alpha + 2\pi n$  for the unique  $n \in \mathbb{Z}$  such that  $\alpha + 2\pi n \in [-\pi, \pi)$ , a common definition of the principal value of an angle. Then the occulter is the “flower” shape:

$$\Omega = \{(r, \theta) : 0 \leq r \leq R, P(N_p \theta) \in [-\pi A(r), \pi A(r)]\}. \quad (15)$$

See Fig. 4. Note that published  $A(r)$  designs are discontinuous at  $a$  (indicating a gap between petals) and at  $R$  (petal tips have finite widths). In early “analytic” designs, these discontinuities were required to be no larger than about  $10^{-5}$  in size to minimize Arago-spot-style diffraction into the deep shadow (Ref. 14, § 4.3), but designs generated by optimization over a  $\lambda z$  band<sup>10,27</sup> have much larger gap and tip discontinuities, on the order of  $10^{-2}$  to  $10^{-3}$ , and an Arago effect that is apparently cancelled out over the band by distributed “ripples” (Ref. 14, §5) in  $A(r)$ .



**Fig. 4** Idealized starshade geometry (zoom for clarity).  $A(r)$  is the apodization profile controlling petal angular width, here illustrated with the HG (offset hyper-Gaussian<sup>14</sup>) with  $N_p = 16$  petals. The colored dots show areal quadrature nodes  $(x_j, y_j)$  with weights  $w_j$  indicated by color as in the colorbar.  $m = 80$  Gauss radii cover the petal length, with  $n_p = 30$  Gauss nodes covering the petal angular width at each radius.

Recall that the task is simply to evaluate Eqs. (1) and (3) with errors in  $u^{\text{oc}}$  no worse than  $10^{-6}$ . To apply the proposed method, we build a high-order areal quadrature as follows. Since  $A(r)$  is discontinuous at  $r = a$ , we split  $\Omega$  into the disc of radius  $a$  plus each of  $N_p$  petals. Our disc quadrature simply applies Eqs. (11) and (12) to the uniform  $n_d$ -node line integral on its boundary, that is, Eq. (14) applied to the parameterization  $[X(t), Y(t)] = (a \cos t, a \sin t)$ . We use  $m_d$  radial nodes. We then cover each petal by  $m$  nodes to handle the (outer) radial integral and then handle the (inner) arc integral at each of their node radii by  $n_p$  angular nodes (see Fig. 4). Specifically, let  $\{r_l, \hat{w}_l\}_{l=1}^m$  be 1D Gauss–Legendre nodes and weights for the (outer) radial integral over  $(a, R)$ . Similarly, let  $\{t_i, \hat{\omega}_i\}_{i=1}^{n_p}$  be nodes and weights for the fixed interval  $[-\pi/N_p, \pi/N_p]$ . Then recalling the area element  $r dr d\theta$ , the resulting areal nodes (in Cartesians) and weights covering one petal are

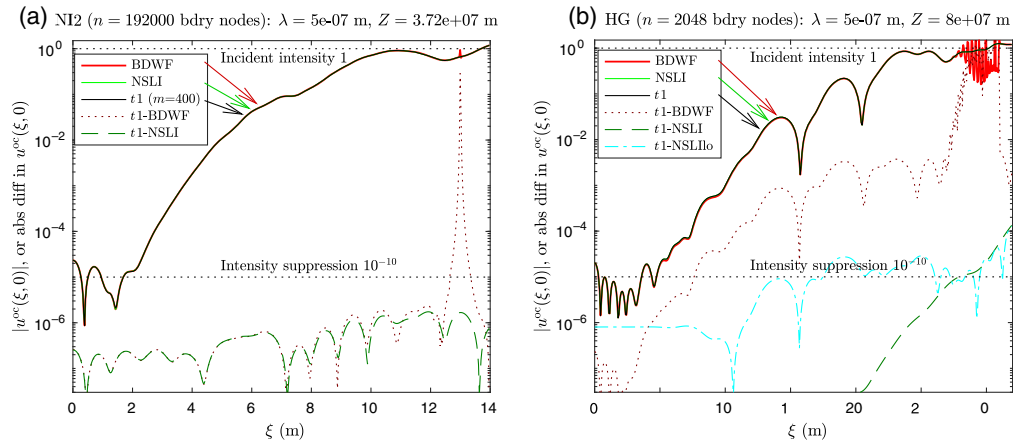
$$\begin{aligned} (x_{l+(i-1)m}, y_{l+(i-1)m}) &= (r_l \cos(A(r_l)t_i), r_l \sin(A(r_l)t_i)), \quad \text{for } i = 1, \dots, n_p, \quad l = 1, \dots, m, \\ w_{l+(i-1)m} &= r_l \hat{w}_l A(r_l) \hat{\omega}_i, \quad \text{for } i = 1, \dots, n_p, \quad l = 1, \dots, m. \end{aligned}$$

Other petals are obtained by rotating by multiples of  $2\pi/N_p$ . The total number of nodes is then  $N = n_d m_d + N_p n_p m$ . We fix  $m_d = m$  and  $n_d \approx 0.3 N_p n_p$ , leaving two (petal) convergence parameters  $m$  and  $n_p$ . In Sec. 4, we discuss applying this to perturbed (non-ideal) starshades.

### 3.2.1 Accuracy validation

In Fig. 5, we compare the wave amplitudes along a radial slice computed by three methods for two designs of starshade: “NI2” (a small occulter with rippled profile optimized for a blue-green  $\lambda$  range<sup>16</sup>) and “HG” (a large occulter with analytic “offset hyper-Gaussian” profile<sup>14</sup>). We choose  $\lambda$  within their designed wavelength windows. The parameters used, and some CPU timings, are listed in Table 2. Since they have different geometry descriptions, we treat the two designs in turn.

**NI2.** The optimized profile  $A(r)$  is available in the SISTER package<sup>27</sup> in the form of 2462 equispaced samples covering the petal radius range  $[a, R] = [5, 13]$  m. From these, we use piecewise cubic splines to interpolate  $A$  at  $m$  radial Gauss nodes in  $[a, R]$ . Since  $A''(r)$  appears to have at least 13 “bang-bang” type discontinuities (the discrete second derivative mostly takes values  $\pm\sigma$ , for some constant  $\sigma$ , or 0), this necessarily limits accuracy to around 6 to 7 digits. By a convergence study, we found that  $m = 400$  and  $n_p = 40$  nodes across each petal were sufficient for areal quadrature to match this accuracy. For BDWF, we used the  $n = 192,000$  boundary nodes as given and used in SISTER. These have 6000 nodes per petal edge, but no nodes covering the interpetal gaps or tips (each of which is 0.03 m wide). For NSLI, we used an  $n$ -node vector line integral quadrature matching the second-order accurate midpoint rule in BDWF (this match was needed to accurately handle the wide tips with a single segment). Figure 5(a) shows that the proposed NUFFT  $t1$  method matches both of these edge integral methods to



**Fig. 5** Validation of diffracted  $u^{\text{oc}}$  along a radial slice for two starshade designs: (a) NI2 (optimized function with ripples<sup>16</sup>) and (b) HG (offset hyper-Gaussian analytic function<sup>14</sup>). Both have  $N_p = 16$  petals. The proposed NUFFT  $t1$  method (black line) is compared against edge-integral methods BDWF (red) and NSLI (green). Labels such as “ $t1$ -BDWF” indicate the absolute difference in  $u^{\text{oc}}$  between two methods. BDWF and NSLI use the same boundary nodes, except for “NSLIlo”, which uses  $10\times$  the boundary nodes and second-order weights. For details, see Sec. 3.2.

**Table 2** Parameters and CPU times for the proposed NUFFT  $t1$  and the BDWF edge-integral to complete the same diffraction tasks, for two starshades. See Fig. 5 for comparisons of their answers. The Fresnel number  $\mathfrak{f}$  uses the maximum radius  $R$  in Eq. (2).  $N$  is the number of areal quadrature nodes and  $n$  is the number of boundary nodes.

Design	$\lambda$ (m)	$z$ (m)	$\mathfrak{f}$	$m$ (petal)	Total nodes	$M$ (targets)	Method	CPU time (s)
NI2	$5 \times 10^{-7}$	$3.72 \times 10^7$	9.1	6000	$n = 192,000$	$10^6$ , grid	BDWF	5361
				400	$N = 499,200$		NUFFT $t1$ ( $\epsilon = 10^{-8}$ )	0.076
HG	$5 \times 10^{-7}$	$8 \times 10^7$	24	60	$n = 2048$	$10^6$ , grid	BDWF	80.5
				60	$N = 37,440$		NUFFT $t1$ ( $\epsilon = 10^{-8}$ )	0.042

around  $3 \times 10^{-7}$  in  $u^{\text{oc}}$  in the shadow region where  $|u^{\text{oc}}| \leq 2 \times 10^{-5}$ . NSLI agrees with  $t1$  to around 6 digits everywhere. However, at  $\xi \approx 13$  m, the error of BDWF spikes to  $\mathcal{O}(1)$  as  $(\xi, 0)$  approaches  $\partial\Omega$ .

**Remark 4.** Since  $z/\lambda \sim 10^{14}$ , overall phase is meaningless; thus we fit the phase of BDWF to the other two methods at a single target. We then quote absolute differences in complex  $u^{\text{oc}}$ . This is a more predictable metric than the error in intensity  $|u^{\text{oc}}|^2$ , which is affected by local intensity.

**HG.** Here the profile is analytically known from Ref. 14:  $A(r) = e^{-[(r-a)/b]^p}$  in  $[a, R]$ , where  $a = b = 12.5$  m, the maximum radius is  $R = 31$  m, and  $p = 6$ . A convergence study shows that only  $m = 60$  and  $n_p = 30$  are needed, giving an areal quadrature of  $N = 37,440$  nodes. For NSLI, we used  $A(r)$  and  $A'(r)$  to generate a high-order line integral quadrature using the same  $m = 60$  radii per petal, plus four Gauss nodes across each gap and tip, giving  $n = 2048$  in total (see `starshadeliquad` in our repository<sup>34</sup>). We also fed these boundary nodes (but of course not their high-order weights) to BDWF. Figure 5(b) shows that the three methods again agree to the desired accuracy almost everywhere, apart from BDWF near  $\partial\Omega$  where again its errors hit  $\mathcal{O}(1)$ .

**Remark 5.** For HG with  $m = 60$ , the errors of BDWF are summarized by 2 to 3 digits of relative accuracy overall, giving 6 to 7 digits of absolute  $u^{\text{oc}}$  accuracy in the deep shadow. Yet NSLI, if fed the low-order midpoint rule used inside BDWF, gives only 4-digit absolute

accuracy; thus it is useless in deep shadow. Figure 5(b) thus also explores (dash-dot line) the errors of NSLI with this low-order rule and the larger  $m = 600$ : the absolute error now bottoms out at a useful  $10^{-6}$ . This highlights an advantage of BDWF over plain NSLI when deep shadows are modeled with poor quadrature, a subtle point explained in Remark 7.

### 3.2.2 Solution speed

Table 2 presents CPU times for the above converged experiments on gridded targets (we omit NSLI times since they are similar to BDWF). The time to construct the areal quadrature is not included for two reasons: (i) it is never more than half the NUFFT method run time and (ii) it would be amortized away over runs at multiple wavelengths. We make the following observations.

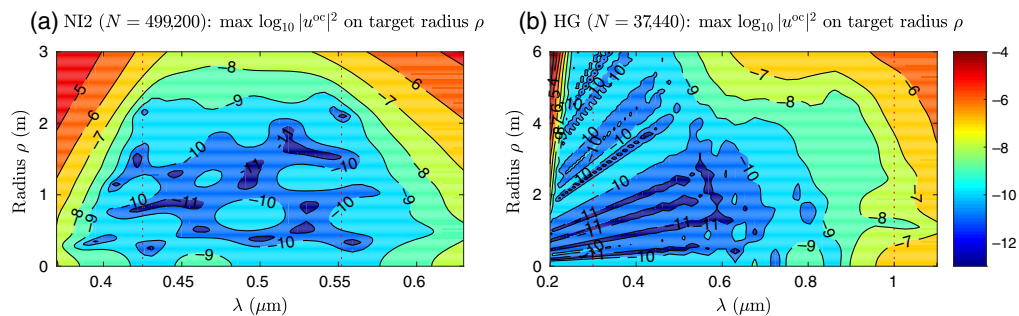
- For NI2, our proposal is around 70,000 $\times$  faster and for HG around is 2000 $\times$  faster than a state-of-the-art edge integral method.
- By reinterpolation of the NI2 profile and code changes to use a high-order quadrature, BDWF could probably be sped up by a factor of 15. BDWF is found also to gain a factor of about two when multiple  $\lambda$  are needed. Neither factor impacts the conclusions much.
- Since the number  $N$  of areal nodes is smaller than the large number  $M$  of targets, the cost of the NUFFT method is almost independent of  $N$ , hence of the starshade complexity, or  $f$ .

We now turn to arbitrary targets. In Fig. 6, the intensity suppression of the two starshade designs are studied, sweeping 50 wavelengths and taking the maximum intensity  $|u^{oc}|^2$  over circles of varying radii  $\rho$  ( $M = 60,000$  targets at each  $\lambda$ ), using the proposed NUFFT  $t3$  method. The narrow-band nature of NI2 and deterioration of HG above  $0.8 \mu\text{m}$  are apparent. The entire calculation for both starshades, including quadrature generation, totals 6 s on the laptop.

Finally, we report initial timing results upon having (rather crudely) inserted the NUFFT  $t3$  method in place of BDWF within the SISTER code base<sup>27</sup> (see `sister_mods` in our repository<sup>34</sup>). Running a standard SISTER “PSF basis” generation task for the non-spinning NI2 starshade, 14 wavelengths are needed covering  $[0.425, 0.555] \mu\text{m}$ , at each of which  $M = 806,144$  targets are needed. (Targets are organized into  $16 \times 16$  telescope pupil grids, translated to 3149 different  $(\xi, \eta)$  centers covering a sector with angle  $2\pi/N_p$ .) We had to reorganize the loop ordering since BDWF was called separately for each pupil while grouping wavelengths together for speed, whereas the NUFFT  $t3$  is most efficient with a single call to all targets at each wavelength.

The original SISTER run time was an estimated 6.5 h (since BDWF gets about  $6 \times 10^7$  node-target pairs per second). The NUFFT  $t3$  method, using parameters as in the previous section and including quadrature generation, took 2.6 s. The acceleration is thus about 10,000 $\times$ .

**Remark 6.** The above shows that the proposed algorithm excels in efficiency when the number of desired targets  $M$  is large, resolving a region similar in size to the occulter. Since its cost



**Fig. 6** Intensity (on  $\log_{10}$  scale indicated on the right) as a function of wavelength and target radius  $\rho$  from the center for the two starshade designs [(a) NI2 and (b) HG] of Fig. 5. At each of 200  $\rho$  values, the maximum over 300 angles is taken. The incident intensity is 1. The NUFFT  $t3$  method is used. Vertical dotted lines show the designed  $\lambda$  range.

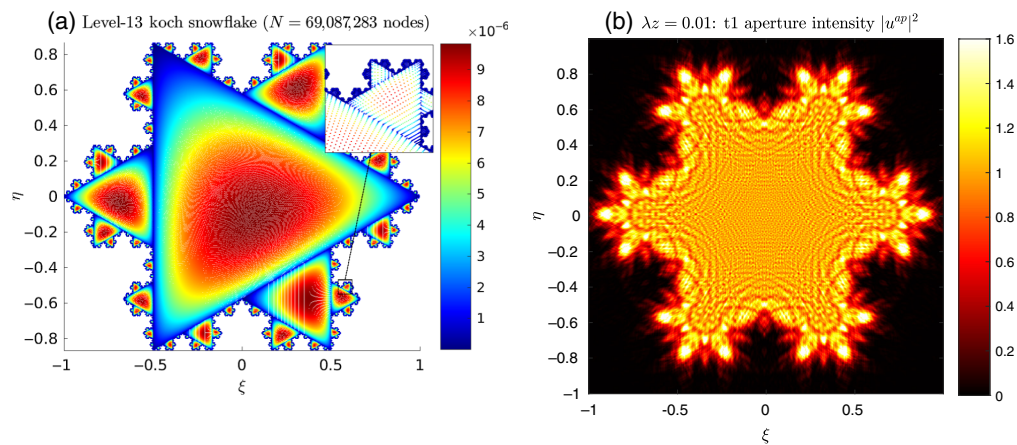
is close to  $\mathcal{O}(N + M)$ , dropping  $M$  does not reduce run time much:  $N$  then dominates, and the relative speed over edge integral methods drops in proportion. The natural question is: what is the crossover  $M$  such that there is no advantage? For the NI2 starshade, since BDWF gives about 250 targets/s, the answer is as small as  $M \approx 20$  for  $t1$  and 50 for  $t3$ . Thus whenever the user needs more targets than this, the NUFFT wins.

### 3.3 Complicated Domain

As a final example, we compute Fresnel diffraction for an aperture with a fractal boundary, specifically the standard Koch snowflake with a maximum radius  $R = 1$ . Let  $\Omega_0$  denote the equilateral triangle with side length  $\sqrt{3}$ ,  $\Omega_1$  is its union with three triangles of side  $\sqrt{3}/3$ ,  $\Omega_2$  is the union of  $\Omega_1$  with 12 triangles of side  $\sqrt{3}/9$ , etc., so that  $\Omega_L$  is the level- $L$  construction [see Fig. 7(a)]. To reach level  $L = 13$ ,  $n_{\text{tri}} = 1 + 3 \sum_{k=0}^{L-1} 4^k = 67,108,864$  triangles are needed. To build an areal quadrature, the integral over each triangle is approximated by a simple  $p \times p$  node product Gauss–Legendre quadrature, by translating one vertex to the origin and then applying Eqs. (11) and (12) to the discretized line integral connecting the other two vertices (see the inset of figure). To accurately handle the oscillatory integrand as in Eq. (6), the node spacing should not exceed  $\mathcal{O}(1/\bar{f})$ ; thus we designed a heuristic choice of  $p$  that varied from 217 for the largest triangle to  $p = 1$  at levels  $L \geq 10$  and checked  $p$ -convergence for Fresnel integrals for  $\lambda z \geq 0.01$ . For  $\Omega_{13}$ , the resulting total node number  $N$  is about 69 million, requiring about 4 min to build in our simple implementation.

The NUFFT  $t1$  method with  $\varepsilon = 10^{-6}$  is then applied to this areal quadrature to resolve the diffracted field for  $\lambda z = 0.01$  on a grid of  $10^6$  target points, as shown in Fig. 7(b). For each new  $\lambda z \geq 0.01$ , this takes 4.6 s. Since  $\Omega_{13}$  has  $3 \times 4^{13} \approx 2 \times 10^8$  edges, an edge integral method of similar accuracy is estimated to be around  $10^5$  to  $10^6 \times$  slower. We checked (via  $p$ -convergence at each level) that this  $u$  computed for  $\Omega_{13}$  has at least 6-digit accuracy.

However, we may also interpret the calculation as an approximation to one for the limit domain  $\Omega_\infty$  with a true fractal boundary. Since the smallest triangles in  $\Omega_{13}$  have sides of  $1.1 \times 10^{-6} \ll \lambda z$ , i.e., much smaller than any Fresnel zone, we are well into the regime of Richardson extrapolation in  $L$ , with differences from the limit scaling like  $(4/9)^L$ . The largest absolute change in  $u$  on the grid in going from  $\Omega_{12}$  to  $\Omega_{13}$  was  $1.6 \times 10^{-4}$ ; thus by extrapolation, the largest change in  $u$  between  $\Omega_{13}$  and the limiting domain  $\Omega_\infty$  is around 4/5 of this. Thus we may quote uniformly about 4-digit accuracy for diffraction from the limit fractal domain. (By applying Richardson to a sequence, one could in fact recover many more digits without much extra effort.)



**Fig. 7** Koch fractal aperture diffraction example from Sec. 3.3. (a) The areal quadrature constructed by a union of about 67 million triangles. The color of each node  $(x_j, y_j)$  indicates its weight  $w_j$  using the scale on the right. The inset shows a zoom into the region shown, resolving individual nodes. (b) Intensity (on  $\log_{10}$  scale indicated on the right) computed on a million-point grid by the NUFFT  $t1$  method in under 5 s.

## 4 Conclusion and Discussion

We have explained a fast algorithm for Fresnel diffraction from binary aperture or occulter shapes, which achieves high accuracy via flexible areal quadrature schemes over the planar domain yet with speeds close to regular-grid FFT propagation methods, via the nonuniform FFT. Extensive tests of error convergence and CPU timings show between two and five orders of magnitude acceleration over edge integral methods at comparable accuracies. Thus at moderate Fresnel numbers  $f \leq 10^2$ , one can evaluate accurate diffraction fields from complicated shapes such as starshades almost instantaneously (0.1 s for a grid of  $10^6$  targets). For higher  $f$ , the  $\mathcal{O}(f^2 \log f)$  cost starts to dominate. RAM usage starts to become a limitation only at  $f \sim 10^3$ . Along the way, we have reformulated edge integrals as an NSLI that is numerically robust (Appendix A).

Although we did not exploit it, the proposed NUFFT method can trivially include smooth source-plane phase/amplitude variations while retaining accurate edge diffraction simply by multiplying  $w_j$  by the source term at each areal node, in step 1 of Sec. 2.1. This is impossible for traditional edge integral methods, but we note exciting recent progress on including low-frequency phase variations with edge integrals (Ref. 17, Sec. 6).

Our findings highlight the importance of high-order accurate quadratures, both for edge integrals and, crucially, planar integrals. We have shown how the latter may be constructed for three classes of domain  $\Omega$  (those with an existing boundary quadrature, ideal starshades, and unions of triangles). Such construction on a case-by-case basis is possible, at least up to the tolerance with which a description of  $\partial\Omega$  is available. Yet to automate this procedure for a requested accuracy and  $f$ , given “any”  $\Omega$ , raises 2D geometry representation and meshing issues common throughout engineering and scientific computing. This is a huge topic with many available tools. Given the benefits that we show in optical simulation, automated high-order areal quadratures from CAD formats should be a useful future project.

### 4.1 Application to Non-Ideal (Perturbed) Starshades

Since the proposed method does not exploit symmetry, it could vastly accelerate Monte Carlo studies of optical stability under the various types of realistic shape perturbations. The latter include manufacturing tolerances, in-flight misalignment, thermal distortions, and damage over time. The topic is complicated by geometry descriptions, statistical correlations, and averaging due to spinning.<sup>16</sup> Numerical study of this is beyond the scope of this initial work. Yet, recall that once an areal quadrature exists for the desired shape  $\Omega$ , the proposed method is very simple. We explain three ways that such a quadrature can be generated as follows.

1. Section 3.1 showed how to automatically build an areal quadrature from an existing boundary quadrature for  $\partial\Omega$ , as in Fig. 2(c). In this way, the method may immediately be applied to a set of boundary nodes describing a perturbed starshade. This will not be as efficient as the quadratures of Sec. 3.2: e.g., for NI2, using the rather large supplied  $n = 192,000$  would give  $N \sim 10^8$ , giving CPU times around 10 s. Yet this is still two to three orders of magnitude faster than edge integrals in the case of  $M = 10^6$  targets.
2. For ideal rigid petals that are misaligned, a deformed areal quadrature may be built as follows: apply rigid motion to the nodes for each petal, without changing their weights, and then add new quadratures covering the new (signed) areas that connect the base arc of each petal with its “root” arc on the ideal disc. This would only add a few nodes to  $N$ , hence preserving the 0.1-s CPU times quoted.
3. A general non-ideal shape may be written as the ideal shape plus a narrow (signed) “ribbon” domain in the neighborhood of the boundary  $\partial\Omega$ . If its width is everywhere smaller than a Fresnel zone, as expected for realistic perturbations, the ribbon contribution is well approximated by a line integral on  $\partial\Omega$ . The resulting quadrature is that of Sec. 3.2 plus a smaller number of boundary nodes, preserving 0.1 s CPU times.

Which of the above methods, or whether another method, proves best in practice will depend on the number of runs required and spatial details of the perturbation.

## 4.2 Other Extensions

Beyond the geometry-handling extensions discussed above, fruitful future directions include the following.

- Further acceleration is probable by drop-in replacement of FINUFFT by a GPU library.
- In this initial work, Gauss–Legendre rules were used for all 1D quadratures. However, since the Fresnel integrands are oscillatory and band-limited, it is probable that substituting high-order 1D rules with quasi-uniform nodes<sup>42,43</sup> will allow  $N$  to be reduced for the same accuracy. Asymptotically at high  $f$ , one would expect a reduction by up to a factor  $(\pi/2)^2 \approx 2.5$ .
- It seems that a NUFFT replacing the first FFT in method (a)(i) from Sec. 1 could enable efficient new “angular spectrum of plane waves” (Ref. 6, Sec. 3.10) (one-way Helmholtz) methods for binary aperture diffraction beyond the Fresnel regime.

## 5 Appendix A: Fresnel Edge Integral Methods—Equivalence and Desingularization

### 5.1 Equivalence of Edge Integral Methods for Planar Waves and Apertures

Dauger<sup>19</sup> noticed that, using polar coordinates about the target  $(\xi, \eta)$ , i.e.,  $x = \xi + r \cos \theta$  and  $y = \eta + r \sin \theta$ , the Fresnel aperture integral Eq. (1) may be analytically integrated in  $r$ , for each  $\theta$ , as follows. Assuming that the target is in  $\Omega$  (geometric shadow) and that  $\Omega$  is strictly star-shaped about this target, Eq. (1) becomes

$$u^{\text{ap}}(\xi, \eta) = \frac{1}{i\lambda z} \int_0^{2\pi} \int_0^{R(\theta)} e^{\frac{ikr^2}{2z}} r dr d\theta = \frac{-1}{2\pi} \int_0^{2\pi} \left[ e^{\frac{ikR(\theta)^2}{2z}} - 1 \right] d\theta, \quad (16)$$

where  $R(\theta)$  is the distance  $r$  where the in-plane ray launched at angle  $\theta$  from the target exits  $\Omega$ . If these conditions are broken,  $R(\theta)$  becomes multivalued. For targets outside  $\Omega$ , the second term in the square brackets must be replaced by one similar to the first but involving the  $r$  value where the ray first enters  $\Omega$ . The resulting numerical method is cumbersome for a general  $\Omega$  because much effort is spent finding, and tracking as a function of  $\theta$ , these multiple ray intersection points.<sup>19</sup>

It is simpler to reformulate Eq. (1) in terms of a line integral over  $\partial\Omega$ . Cash [Ref. 14, (46)] provides such a formula but no rigorous derivation. To remedy this, fixing the target, we write  $\mathbf{r} := (x - \xi, y - \eta)$  and hence  $r^2 = \|\mathbf{r}\|^2$ , and we define the 2D vector field:

$$\mathbf{F}(x, y) := \frac{-1}{2\pi} \frac{\mathbf{r}}{r^2} e^{\frac{ikr^2}{2z}}, \quad (x, y) \neq (\xi, \eta). \quad (17)$$

After cancelling terms, its divergence is found to be simply the Fresnel integrand from Eq. (1) minus the unit 2D delta distribution at the target (the latter can be proven by excluding a small disk of radius  $r \rightarrow 0$  about the target), that is,

$$\nabla \cdot \mathbf{F}(x, y) = \frac{1}{i\lambda z} e^{\frac{ikr^2}{2z}} - \delta(\mathbf{r}). \quad (18)$$

Applying the divergence theorem in  $\Omega$ , with  $\mathbf{n}$  being the unit outward normal, and, as before, taking the 2D cross product as a scalar,

$$\iint_{\Omega} \nabla \cdot \mathbf{F} dx dy = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds = \int_{\partial\Omega} \mathbf{F} \times \mathbf{ds} = \frac{-1}{2\pi} \int_{\partial\Omega} e^{\frac{ikr^2}{2z}} \frac{\mathbf{r} \times \mathbf{ds}}{r^2}.$$

Substituting Eq. (18) and recalling Eq. (1) gives the line integral formula

$$u^{\text{ap}}(\xi, \eta) = \frac{-1}{2\pi} \int_{\partial\Omega} e^{\frac{ikr^2}{2z}} \frac{\mathbf{r} \times \mathbf{ds}}{r^2} + u_{\text{geom}}^{\text{ap}}(\xi, \eta), \quad \text{where } u_{\text{geom}}^{\text{ap}}(\xi, \eta) := \begin{cases} 1, & (\xi, \eta) \in \Omega \\ 0, & \text{otherwise} \end{cases}. \quad (19)$$

This is easily seen to be equivalent to Dager's formulas using the facts: (i)  $d\theta = (\mathbf{r} \times d\mathbf{s})/r^2$ , (ii)  $u_{\text{geom}}^{\text{ap}}(\xi, \eta) = \int_0^{2\pi} d\theta/2\pi$ , and (iii) the multiple values of  $R(\theta)$  correspond to  $\theta$  folding back as  $\partial\Omega$  is traversed.

We now show that, within the Fresnel approximation, the Miyamoto–Wolf [Ref. 26, Eqs. (5.1), (5.5)] BDW formulation used by Cady<sup>15</sup> is also equivalent to the above. Given  $u(x, y)$ , the plane incident wave at the aperture with unit direction vector  $\mathbf{p}$ , this states (noting that our  $u^{\text{ap}}$  definition excludes the phase of plane  $z$ -propagation) that

$$u^{\text{ap}}(\xi, \eta) = \frac{1}{4\pi} e^{-2\pi iz/\lambda} \int_{\partial\Omega} u(x, y) \frac{e^{2\pi i\rho/\lambda} \hat{\rho} \times \mathbf{p} \cdot d\mathbf{s}}{\rho (1 + \hat{\rho} \cdot \mathbf{p})} + u_{\text{geom}}^{\text{ap}}(\xi, \eta). \quad (20)$$

Here we recall that  $\rho = \sqrt{r^2 + z^2}$  is the target-source 3D distance and define  $\hat{\rho}$  to be the unit vector pointing from the target to the source ( $\rho$  is notated as  $s$  in standard references, but we reserve the latter for arclength). Since we are concerned with planar incidence,  $u(x, y) \equiv 1$  and  $\mathbf{p} = (0, 0, 1)$ . Since  $r \ll z$  is implicit in Eq. (1), we insert the leading-order small-angle approximations  $\rho \approx z$ ,  $\hat{\rho} \times \mathbf{p} \cdot d\mathbf{s} \approx (\mathbf{r} \times d\mathbf{s})/z$ , and  $1 + \hat{\rho} \cdot \mathbf{p} \approx r^2/(2z^2)$  and the usual Fresnel approximation  $e^{2\pi i\rho/\lambda} \approx e^{2\pi iz/\lambda} e^{i\pi r^2/z\lambda}$ . The result is precisely Eq. (19). Thus all three edge formulations are equivalent.

However, it is worth noting that BDW Eq. (20) and some formulas in Dubra–Ferrari<sup>22</sup> have a wider range of validity than Eq. (1), Dager's formulas, or Eq. (19) since they allow for out-of-plane apertures and more general incident waves.

## 5.2 Robust Non-Singular Line Integral Formulation

To our knowledge, all edge integral numerical codes use formulas shown in the previous section to be equivalent to Eq. (19) and are thus well known to be plagued by two serious problems as follows.<sup>14,15,17,19,22</sup>

1. Targets must be labeled as being inside or outside  $\Omega$  in a robust fashion, no matter how close they are to  $\partial\Omega$ ; otherwise  $\mathcal{O}(1)$  errors result.
2. When the target approaches  $\partial\Omega$ , the integrand on  $\partial\Omega$  becomes nearly singular, requiring increasingly refined quadrature near the target to retain accuracy. (Dager's  $\theta$ -parameterization conceals this but does not remove the difficulty since  $R(\theta)$  changes arbitrarily rapidly.)

For instance, in Secs. 3.1 and 3.2.1, we saw that the BDWF code loses all accuracy near  $\partial\Omega$ . However, once it is realized that the two problems are in fact facets of the same phenomenon, they can be made to “cancel out.”

This works as follows. It is well known [Ref. 44, (6.23)] [or combining facts (i) and (ii) above] that

$$u_{\text{geom}}^{\text{ap}}(\xi, \eta) = \frac{1}{2\pi} \int_{\partial\Omega} \frac{\mathbf{r} \times d\mathbf{s}}{r^2}. \quad (21)$$

Inserting this into Eq. (19) gives one formula that applies whether the target is inside or outside  $\Omega$ :

$$u^{\text{ap}}(\xi, \eta) = \frac{1}{2\pi} \int_{\partial\Omega} \left(1 - e^{i\pi r^2/z\lambda}\right) \frac{\mathbf{r} \times d\mathbf{s}}{r^2} \quad (\text{NSLI formula}). \quad (22)$$

This has no singularity as  $r \rightarrow 0$  (target approaching  $\partial\Omega$ ) because the term in square brackets is  $\mathcal{O}(r^2)$ , cancelling the denominator. The integrand is as smooth as the Fresnel zones, i.e., as smooth as the diffracted field in the target plane. We believe that Eq. (22) is new.

This leads to an incredibly simple yet robust code. For instance, in MATLAB, if  $\mathbf{bx}$  and  $\mathbf{by}$  are list coordinates of nodes on  $\partial\Omega$ , with  $\mathbf{wx}$  and  $\mathbf{wy}$  being the corresponding weights for a vector line integral as in Sec. 3.1, the entire NSLI code to output  $u^{\text{ap}}$  at a target  $(\mathbf{x}_i, \mathbf{y}_i)$  is five lines:



```

rx = bx - xi; ry = by - eta; % components of r displacement vector
r2 = rx.*rx + ry.*ry; % r^2
f = (1 - exp((1i*pi/lambdaz)*r2)) ./ r2;
f(r2==0.0) = 0.0; % kill NaNs (target hits node)
uap = sum((rx.*wy - ry.*wx) .* f) / (2*pi); % cross product, quadrature

```

Note that when a target hits a node to within machine error ( $r = 0$ ), any finite value of  $f$  may be inserted in line 4 because  $\mathbf{r} \times d\mathbf{s} = 0$  in line 5. Yet there is a subtlety here: numerical eyebrows should immediately be raised because  $f$  involves catastrophic cancellation as  $r \rightarrow 0$ . To understand why this is in fact barely a problem, we apply forward error analysis [Ref. 45, Ch. 1] and treat the real and imaginary parts separately (combining them leads to a pessimistic prediction).

The imaginary part of the  $\exp$  is  $\sin[(\pi/\lambda z)r^2]$ , which, given a rounded value of  $r2$ , is computed to relative accuracy  $\mathcal{O}(\epsilon_{\text{mach}})$ , where  $\epsilon_{\text{mach}} \approx 1.1 \times 10^{-16}$  is the usual double precision relative error. Subtraction from 1 does not change the imaginary part. The division by  $r2$  then results in absolute error  $\mathcal{O}(\epsilon_{\text{mach}})$ , which then gets multiplied by the  $\mathcal{O}(r)$  cross product, giving  $\mathcal{O}(\epsilon_{\text{mach}}r)$ . Note that this holds even though  $r2$  is necessarily inaccurate due to coordinate subtraction in line 1.

Now to the real part of the  $\exp$ , which is  $1 + \mathcal{O}(r^4)$ . Thus when  $r \lesssim \epsilon_{\text{mach}}^{1/4}$ , the real part of  $\exp$  is in machine arithmetic exactly 1, which cancels the other 1 exactly, leaving zero. Since the true answer is  $\mathcal{O}(r^3)$ , in this regime the final error is bounded by  $\mathcal{O}(\epsilon_{\text{mach}}^{3/4})$ . On the other hand, for  $r \gtrsim \epsilon_{\text{mach}}^{1/4}$ , catastrophic cancellation occurs: the error in the real part of  $\exp$  is  $\mathcal{O}(\epsilon_{\text{mach}})$ , so the final error is  $\mathcal{O}(\epsilon_{\text{mach}}/r)$ . In summary, uniformly in  $r$ , the final error is bounded by  $\mathcal{O}(\epsilon_{\text{mach}}^{3/4})$ . In practice, we find by comparison to the areal quadrature answers that this uniform bound is around  $10^{-14}$ , which is adequate. Replacing by a Taylor expansion for small  $r$  (e.g., via `cexpr1`<sup>46</sup>) could possibly gain a digit.

Equation (22), in the form of the above code looped over target points, serves as our reference direct method. A documented, tested MATLAB/Octave implementation is in the repository<sup>34</sup> `bdrymeths/nsli_pts.m`

**Remark 7.** When a poor quadrature (that is, low order and few nodes) is used with deep shadow regions, the usual line integral Eq. (19) has one advantage over NSLI Eq. (22): it can in shadows achieve relative accuracy in  $u$ , appropriate to the quadrature, because  $u_{\text{geom}} = 0$  exactly. NSLI merely achieves absolute accuracy in  $u$ ; thus it may require a better quadrature to resolve deep shadows than Eq. (19) (as implemented by, e.g., BDWF). In essence,  $u_{\text{geom}}^{\text{ap}} \approx 1$  [the “1” term in Eq. (22)] to limited accuracy, which is then poorly canceled in Eq. (3). To remedy this, our NSLI implementation also includes an option to use Eq. (19) for targets far from  $\partial\Omega$ , combining the robustness of Eq. (22) with the deep shadow relative accuracy of traditional edge integrals.

## Acknowledgments

This work benefited from the input of the anonymous reviewers and discussions with David Spiegel, Robert Vanderbei, Stuart Shaklan, and Eric Cady. The Flatiron Institute is a division of the Simons Foundation.

## References

1. M. Born and E. Wolf, *Principles of Optics*, 6th ed., Pergamon Press, Oxford (1980).
2. P. M. Morse and K. U. Ingard, *Theoretical Acoustics*, McGraw-Hill, New York (1968).
3. M. D. Perrin et al., “Simulating point spread functions for the James Webb Space Telescope with WebbPSF,” *Proc. SPIE* **8442**, 84423D (2012).
4. Y. Hu et al., “Efficient full-path optical calculation of scalar and vector diffraction using the Bluestein method,” *Light Sci. Appl.* **9**(1), 119 (2020).
5. A. J. Bourdillon et al., “A critical condition in Fresnel diffraction used for ultra-high resolution lithographic printing,” *J. Phys. D* **33**(17), 2133–2141 (2000).

6. J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed., McGraw-Hill (1996).
7. M. Ruiz-Lopez et al., “Coherent x-ray beam metrology using 2D high-resolution Fresnel-diffraction analysis,” *J. Synchrotron. Rad.* **24**(1), 196–204 (2017).
8. T. D. Mast, “Fresnel approximations for acoustic fields of rectangularly symmetric sources,” *J. Acoust. Soc. Am.* **121**, 3311–3322 (2007).
9. P. Tsang et al., “Computer generation of binary Fresnel holography,” *Appl. Opt.* **50**(7), B88–B95 (2011).
10. R. J. Vanderbei, E. J. Cady, and N. J. Kasdin, “Optimal occulter design for finding extrasolar planets,” *Astrophys. J.* **665**(1), 794–798 (2007).
11. D. Serre, “The Fresnel imager: instrument numerical model,” *Exp. Astron.* **30**, 111–121 (2011).
12. R. Wilhem and K. Laurent, “Improvements on Fresnel arrays for high contrast imaging,” *Exp. Astron.* **45**, 21–40 (2018).
13. A. S. Lo, T. Glassman, and C. Lillie, “New worlds observer optical performance,” *Proc. SPIE* **6687**, 668716 (2007).
14. W. Cash, “Analytic modeling of starshades,” *Astrophys. J.* **738**(1), 76 (2011).
15. E. J. Cady, “Boundary diffraction wave integrals for diffraction modeling of external occulters,” *Opt. Express* **20**(14), 15196–15208 (2012).
16. S. B. Shaklan, L. Marchen, and E. Cady, “Shape accuracy requirements on starshades for large and small apertures,” *Proc. SPIE* **10400**, 104001T (2017).
17. A. Harness et al., “Advances in edge diffraction algorithms,” *J. Opt. Soc. Am. A* **35**(2), 275–285 (2018).
18. A. Harness et al., “Modeling non-scalar diffraction in the Princeton starshade testbed,” *Proc. SPIE* **10698**, 1069865 (2018).
19. D. E. Dager, “Simulation and study of Fresnel diffraction for arbitrary two-dimensional apertures,” *Comput. Phys.* **10**(6), 591–604 (1996).
20. J. E. Harvey and J. L. Fordham, “The spot of Arago: new relevance for an old phenomenon,” *Am. J. Phys.* **52**(3), 243–247 (1984).
21. G. E. Sommargren and H. J. Weaver, “Diffraction of light by an opaque sphere. 1: Description and properties of the diffraction pattern,” *Appl. Opt.* **29**, 4646–4657 (1990).
22. A. Dubra and J. A. Ferrari, “Diffracted field by an arbitrary aperture,” *Am. J. Phys.* **67**(1), 87–92 (1999).
23. L. Junchang and W. Yanmei, “An indirect algorithm of Fresnel diffraction,” *Opt. Commun.* **282**, 455–458 (2009).
24. D. Mas et al., “Fast algorithms for free-space diffraction patterns calculation,” *Opt. Commun.* **164**, 233–245 (1999).
25. A. F. Oskooi et al., “MEEP: a flexible free-software package for electromagnetic simulations by the FDTD method,” *Comput. Phys. Commun.* **181**(3), 687–702 (2010).
26. K. Miyamoto and E. Wolf, “Generalization of the Maggi-Rubinowicz theory of the boundary diffraction wave—Part II,” *J. Opt. Soc. Am.* **52**, 626–636 (1962).
27. S. R. Hildebrandt et al., “Starshade imaging simulation toolkit for exoplanet reconnaissance (SISTER),” MATLAB software, <http://sister.caltech.edu/> (accessed 2 September 2020).
28. D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, 2nd ed., Vol. **93**, Applied Mathematical Sciences, Springer-Verlag, Berlin (1998).
29. O. P. Bruno and S. K. Lintner, “A high-order integral solver for scalar problems of diffraction by screens and apertures in three-dimensional space,” *J. Comput. Phys.* **252**, 250–274 (2013).
30. A. Dutt and V. Rokhlin, “Fast Fourier transforms for nonequispaced data,” *SIAM J. Sci. Comput.* **14**, 1368–1393 (1993).
31. J.-Y. Lee and L. Greengard, “The type 3 nonuniform FFT and its applications,” *J. Comput. Phys.* **206**, 1–5 (2005).
32. R. Soummer et al., “Fast computation of Lyot-style coronagraph propagation,” *Opt. Express* **15**(24), 15935–15951 (2007).
33. A. H. Barnett, J. F. Magland, and L. af Klinteberg, “A parallel non-uniform fast Fourier transform library based on an ‘exponential of semicircle’ kernel,” *SIAM J. Sci. Comput.* **41**(5), C479–C504 (2019).

34. A. H. Barnett, “FRESNAQ: MATLAB/Octave library for fast Fresnel diffraction from apertures and occulters,” 2020, <https://github.com/ahbarnett/fresnaq>.
35. L. N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, <http://www.chebfun.org/ATAP> (2013).
36. B. Vioreanu and V. Rokhlin, “Spectra of multiplication operators as a numerical tool,” *SIAM J. Sci. Comput.* **36**, A267–A288 (2014).
37. S. E. Mousavi, H. Xiao, and N. Sukumar, “Generalized Gaussian quadrature rules on arbitrary polygons,” *Int. J. Numer. Methods Eng.* **82**(1), 99–113 (2010).
38. H. Xiao and Z. Gimbutas, “A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions,” *Comput. Math. Appl.* **59**(2), 663–676 (2010).
39. D. Gunderman, K. Weiss, and J. A. Evans, “Spectral mesh-free quadrature for planar regions bounded by rational parametric curves,” *Comput. Aided Des.* **130**, 102944 (2021).
40. L. Bluestein, “A linear filtering approach to the computation of discrete Fourier transform,” *IEEE Trans. Audio Electroacoust.* **18**, 451–455 (1970).
41. J. Keiner, S. Kunis, and D. Potts, “Using NFFT 3—a software library for various nonequispaced fast Fourier transforms,” *ACM Trans. Math. Software* **36**(4), 1–30 (2009).
42. B. K. Alpert, “Hybrid gauss-trapezoidal quadrature rules,” *SIAM J. Sci. Comput.* **20**, 1551–1584 (1999).
43. N. Hale and L. N. Trefethen, “New quadrature formulas from conformal maps,” *SIAM J. Numer. Anal.* **46**(2), 930–948 (2008).
44. R. Kress, *Linear Integral Equations*, 2nd ed., Vol. **82**, Applied Mathematical Sciences, Springer, New York (1999).
45. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia (2002).
46. W. Fullerton, “cexprl.f: Fortran code for the complex relative error exponential,” SLATEC library, 1989, <https://www.netlib.org/slatec/fnlib/cexprl.f>.

**Alex H. Barnett** received his PhD in physics from Harvard University, followed by post-doctoral work in radiology at Massachusetts General Hospital and in applied mathematics at New York University. He then taught in the Department of Mathematics at Dartmouth College for 12 years, becoming a full professor in 2017. He is a group leader for numerical analysis at the Center for Computational Mathematics, Flatiron Institute, New York, USA. His research interests include computational partial differential equations, waves, fast algorithms, integral equations, neuroscience, imaging, and inverse problems. He has published more than 50 papers, received several grants from the National Science Foundation, won the XXI International Physics Olympiad, and received Dartmouth’s Karen E. Wetterhahn Memorial Award for Distinguished Creative or Scholarly Achievement.