Eulerian Geometrical Optics and Fast Huygens Sweeping Methods for Three-Dimensional Time-Harmonic High-Frequency Maxwell's Equations in Inhomogeneous Media

Jianliang Qian [*] Wangtao Lu [†] Lijun Yuan [‡] Songting Luo [§]

Robert Burridge [¶]

March 18, 2015

Abstract

In some applications, it is reasonable to assume that geodesics (rays) have a consistent orientation so that Maxwell's equations may be viewed as an evolution equation in one of the spatial directions. With such applications in mind, we propose a new Eulerian geometrical-optics method, dubbed the fast Huygens sweeping method, for computing Green's functions of Maxwell's equations in inhomogeneous media in the high-frequency regime and in the presence of caustics. The first novelty of the fast Huygens sweeping method is that a new dyadic-tensor type geometrical-optics ansatz is proposed for Green's functions which is able to utilize some unique features of Maxwell's equations. The second novelty is that the Huygens-Kirchhoff secondary source principle is used to integrate many locally valid asymptotic solutions to yield a globally valid asymptotic solution so that caustics associated with the usual geometrical-optics ansatz can be treated automatically. The third novelty is that a butterfly algorithm

[*] Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA. Email: qian@math.msu.edu

[†] Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA. Email: wangtaol@math.msu.edu

[‡] College of Math & Statistics, Chongqing Technology and Business University, Chongqing 400067, PRC. Email: yuanlijun.ma@gmail.com

[§] Department of Mathematics, Iowa State University, Ames, IA 50011, USA. Email: luos@iastate.edu

[¶] Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131, USA. Email: burridge137@gmail.com

is adapted to carry out the matrix-vector products induced by the Huygens-Kirchhoff integration in $O(N \log N)$ operations, where N is the total number of mesh points, and the proportionality constant depends on the desired accuracy and is independent of the frequency parameter. To reduce the storage of the resulting traveltime and amplitude tables, we compress each table into a linear combination of tensor-product based multivariate Chebyshev polynomials so that the information of each table is encoded into a small number of Chebyshev coefficients.

The new method enjoys the following desired features: (1) it precomputes a set of local traveltime and amplitude tables; (2) it automatically takes care of caustics; (3) it constructs Green's functions of Maxwell's equations for arbitrary frequencies and for many point sources; (4) for a specified number of points per wavelength it constructs each Green's function in nearly optimal complexity in terms of the total number of mesh points, where the prefactor of the complexity only depends on the specified accuracy and is independent of the frequency parameter.

Three-dimensional numerical experiments are presented to demonstrate the performance and accuracy of the new method.

1 Introduction

We consider the time-harmonic frequency-domain Maxwell's equations in a linear, isotropic and nonmagnetic dielectric medium,

$$\nabla \times \nabla \times \boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') - k_0^2 m^2(\boldsymbol{r}) \boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') = \boldsymbol{I}\delta(\boldsymbol{r}-\boldsymbol{r}'), \quad \boldsymbol{r}', \boldsymbol{r} \in \Omega,$$

where the Sommerfeld radiation condition is imposed at infinity, Ω is a volume, $m(\boldsymbol{r})$ is the index of refraction, k_0 is a large parameter, $\boldsymbol{I}$ is the 3 by 3 identity matrix, δ is the Dirac-delta function, ∇ is the gradient operator, and the dyad $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')$ as a 3 by 3 matrix-valued function mapping a 3-component vector to another 3-component vector is the electric Green's function at a point $\boldsymbol{r} = (x, y, z)^T$ due to the source at $\boldsymbol{r}'$. Since this equation arises in a variety of applications, ranging from optics, microwaves, antennas and radars, it is highly desirable to develop efficient and accurate numerical methods for Maxwell's equations. When the frequency parameter k_0 is large, the solution of Maxwell's equations is highly oscillatory so that it is very costly for a direct method to resolve these oscillations due to so-called numerical dispersion or pollution errors [3]. Therefore, we seek alternative methods, such as asymptotic methods, to solve Maxwell's equations in the high frequency regime. In this paper, we propose a new Eulerian geometrical-optics method, which we call the fast Huygens sweeping method, for Maxwell's equations in the high frequency regime and in the presence of caustics. The new method consists of three critical elements: a novel Eulerian geometrical-optics ansatz, the Huygens secondary source principle, and a modified butterfly algorithm for constructing low-rank matrix approximations, and the resulting algorithm can compute the Green's function of Maxwell's equations in almost linear complexity in terms of the total number of mesh points when the underlying wave field is sampled with a fixed number of points per wavelength.

In [29, 41] we have applied the standard geometrical-optics ansatz to Helmholtz equations in the process of designing fast Huygens sweeping methods in inhomogeneous media in the high frequency regime. Unlike the Helmholtz equation which is a scalar equation, Maxwell's equations are a system of equations for electric-magnetic fields with some unique features. It turns out that these features enable us to design a novel Eulerian geometrical-optics ansatz. This ansatz for the Green's function consists of three non-oscillatory functions, a phase (traveltime) function satisfying an eikonal equation, an amplitude function solving a transport equation, and a polarization dyadic function satisfying a system of advection equations. Since it is very complicated to directly compute the polarization dyad by solving the system of advection equations, we propose a reformulation so that the polarization dyad can be constructed from some auxiliary quantities which can be computed easily.

As is well known and is illustrated in many contexts [5, 14, 26, 29, 41], classical geometrical-optics ansatz fails at caustics; moreover, caustics will occur with high probability for wave propagation in inhomogeneous media [53]. A natural question arises: how to treat caustics when applying geometrical optics to computing the Green's function for Maxwell's equations in a linear, isotropic and nonmagnetic dielectric medium? To treat caustics we will use the Huygens secondary source principle. It has been shown in [2, 33, 47] that the isotropic eikonal equation with a point-source boundary condition has a locally smooth solution near the point source except at the source point itself (but the squared eikonal is smooth at the source [29, 28, 41]). This implies that caustics will not develop right away from the source point in isotropic media, such as isotropic, dielectric media, which are considered here. Based on this observation and the Huygens secondary source principle, we can integrate locally valid asymptotic Green's functions into a globally valid asymptotic Green's function by using the Huygens-Kirchhoff integral so that caustics associated with the usual geometrical-optics ansatz can be treated automatically.

To go beyond caustics, we assume that geodesics (rays) have a consistent orientation so that Maxwell's equations may be viewed as an evolution equation in one of the spatial directions, and this is a reasonable assumption for many applications such as lasers [6, 7], nano-optics [35], and electro-magnetic methods for geophysical exploration [59]. Thus along the consistent orientation, we partition the computational domain into subdomains or layers satisfying certain properties and then develop a layer-by-layer sweeping solver for the wave field by the Huygens principle. Specifically, the first layer is identified as a local neighborhood of the primary source except the source-point itself, where the asymptotic Green's function is locally valid. Next, we set up secondary sources on the boundary of the first layer which is regarded as the first secondary-source plane, and we further identify the second layer as the region where asymptotic Green's functions excited by secondary sources are valid. Then, we apply the Huygens-Kirchhoff integral identity to carry out an integration along the first secondary-source plane, so that we can synthesize information from those secondary sources to construct the globally valid Green's function for the primary source in the second layer. Repeating this process, we can sweep through the whole domain to obtain the globally valid Green's function for the primary source, and caustics are automatically taken care of during this sweeping process.

So the question now is how to implement the above sweeping strategy efficiently. To tackle this challenging problem, we must surmount several obstacles. The first obstacle is that the traveltime and amplitude functions for the eikonal and transport equations with point-source conditions have upwind singularities at the source point, making it difficult to compute these two functions with high-order accuracy; moreover, the occurrence of the Laplacian of the traveltime in the transport equation makes solving the transport equation an even more delicate task. To deal with this obstacle, we use our newly developed high-order schemes for computing the first-arrival traveltimes and amplitudes as illustrated in [26, 30, 29, 28].

The second obstacle is how to store many tables of traveltime, amplitude, and polarization dyad that our sweeping strategy will generate for those specified secondary sources. This storage issue is critical as we are aiming at solving both 3-D Maxwell's equations in the high frequency regime. To reduce data storage, we compress each table into a linear combination of tensor-product based multivariate Chebyshev polynomials so that the information of each table is encoded into a small number of Chebyshev coefficients. Computationally, such compression leads to a significant storage reduction and efficient memory accesses.

The third obstacle is how to carry out efficiently the dense matrix-vector products induced by the Huygens-Kirchhoff integration. Let n be the number of mesh points along each coordinate direction of the computational domain, so that the total number of mesh points is $N = n^d$ in the d-dimensional case. Because we are interested in the asymptotic solution everywhere in the computational domain, the solution at observation points (receivers) in the 3-D case corresponds to the result of some matrix-vector products. In the 3-D case, straightforward implementation of the above matrix-vector products will lead to $O(N^{\frac{4}{3}})$ operations for each 2-D plane of receivers, and the total computation cost will be $O(N^{\frac{5}{3}})$ as we need to carry out such matrix-vector products for roughly $N^{\frac{1}{3}}$ planes of receivers. Such computational cost is far too high to make our strategy practical. Therefore, to tackle this difficulty, we adapt to our application the multilevel matrix decomposition based butterfly algorithm [32, 36, 55, 10, 13, 28, 41]. The resulting butterfly algorithm allows us to carry out the required matrix-vector products with the total computational cost of $O(N \log N)$ complexity, where the proportionality constant depends only on the specified accuracy and is independent of the frequency parameter k_0. Such low-rank rapid matrix-vector products are responsible for the adjective "fast" in the name "fast Huygens-sweeping method" of our method.

The fast Huygens-sweeping method also has two unique merits which may be attributed to the precomputed tables of traveltime, amplitude, and polarization dyad. The first merit is that because the traveltime, amplitude, and polarization dyad functions are independent of the frequency parameter, those tables can be used to construct asymptotic Green's functions at a given primary source for arbitrary frequencies. The second merit is that those tables can be used to construct asymptotic Green's functions at many other primary sources for arbitrary frequencies as well. These two merits are much desired in many applications, such as nano-optics and electro-magnetic methods for geophysical exploration.

1.1 Related work

The high-order schemes for the eikonal and transport equations that we are using here were developed in [26, 30, 29], which in turn are based on Lax-Friedrichs sweeping [19, 58, 51, 57, 46], weighted essentially non-oscillatory (WENO) finite difference approximation [37, 24, 18, 17], and factorization of the upwind source singularities [38, 56, 16, 26, 30, 27, 29]. To treat the upwind singularity at the point source, an adaptive method for the eikonal and transport equations has been proposed in [42] as well. See [5, 14] for reviews on Eulerian geometrical optics.

The idea of compressing a traveltime table into a small number of coefficients in a certain basis has been used frequently in seismic imaging by the geophysical community. Here we use the tensor-product based Chebyshev polynomials as the basis to compress the tables of traveltime, amplitude and polarization dyad, as inspired by the work in [1].

To construct global asymptotic solutions for the Helmholtz equation or Maxwell's equations even in the presence of caustics, there exist three possible approaches in the literature. The first approach is based on Ludwig's uniform asymptotic expansions at caustics [25, 9] which require that the caustic structure be given. The second approach is based on the Maslov canonical operator theory [31]. Although the Maslov theory is beautiful, it is not so useful as it requires identifying where caustics occur first before the theory can be applied; in practice, caustics can occur anywhere along a central ray in an inhomogeneous medium with a high probability as shown in [53]. The third approach is based on Gaussian beam methods [11, 39, 45, 54, 22, 49, 48]. Although Gaussian beam methods can treat caustics automatically along a central ray, the method itself suffers from expensive beam summation and exponential growth of beam width as analyzed and illustrated in [22, 43, 44, 20, 21, 34, 48], and such shortcomings sometimes have hindered applications of Gaussian beam methods to complicated inhomogeneous media. In addition, Benamou et al. [6, 7] designed an Eulerian geometrical-optics method for computing high-frequency electromagnetic fields in the vicinity of fold caustics. Our proposed new method is different from the above approaches.

The origin of the multilevel-matrix decomposition based butterfly algorithm can be traced back to the work [32], and it has been further developed in [36, 55, 10, 13]. In this work, we are using the version of the fast butterfly algorithm first developed in [10] and further analyzed in [13]. In [28, 41], a version of the fast butterfly algorithm was implemented for Helmholtz equations. The significance of the fast butterfly algorithm for high-frequency wave computation was illustrated in a recent work [15]. Engquist and Zhao [15] showed that the ϵ-approximate separability of $G(\boldsymbol{r};\boldsymbol{r}') = A(\boldsymbol{r};\boldsymbol{r}')e^{ik_0\tau(\boldsymbol{r};\boldsymbol{r}')}$ has k_0-dependent lower and upper bounds when $\boldsymbol{r} \in X$ and $\boldsymbol{r}' \in Y$ with $X \subset R^3$ and $Y \subset R^3$ disjoint and compact. This result has two implications. The first implication is that when k_0 is fixed, the matrix corresponding to the discretized Green's function will have finite ϵ-numerical ranks no matter how dense the sampling is; consequently, low-rank structures exist in the corresponding matrix when the wavefield is over-resolved per wavelength, which is not optimal in practice. The second implication is that when k_0 increases, the ϵ-numerical rank for the corresponding matrix increases as well so that no obvious low-rank structure exists when the wavefield is resolved with a fixed number of points per wavelength. To create low-rank structures in the corresponding matrix, we have to set up the two sets X and Y in a k_0-dependent manner which is exactly the departure point for fast butterfly algorithms in [10, 28, 41] and in the current paper.

1.2 Layout

The rest of the paper is organized as follows. In section 2, we present details of constructing a locally valid dyadic Green's function. In section 3, we present our fast Huygens-sweeping method for constructing a globally valid dyadic Green's function, along with the details of implementation and the complexity analysis. In section 4, we study several numerical examples to illustrate the effectiveness of our method.

2 Eulerian geometrical optics for Maxwell's equations

2.1 Maxwell's equations

The time-harmonic frequency-domain Maxwell's equations in a linear, isotropic, and non-magnetic dieletric medium bounded by a volume Ω can be written as the following,

$$\nabla \times \boldsymbol{H} = -i\omega\epsilon_0\epsilon\boldsymbol{E} + \boldsymbol{J}, \tag{1a}$$

$$\nabla \times \boldsymbol{E} = i\omega\mu_0\boldsymbol{H}, \tag{1b}$$

where the electric field $\boldsymbol{E}$, the magnetic field $\boldsymbol{H}$, and the current density $\boldsymbol{J}$ are 3-component vector functions in $\boldsymbol{r} = (x, y, z)^T$, ω is the angular frequency, μ_0 is the magnetic permeability of vacuum, ϵ_0 is the permittivity of vacuum, $\epsilon(\boldsymbol{r}) = m^2(\boldsymbol{r})$ is the dielectric function of the medium with $m(\boldsymbol{r})$ being the index of refraction, and $\nabla = [\partial_x, \partial_y, \partial_z]^T$ is the gradient operator at point $\boldsymbol{r}$.

Since the electric field $\boldsymbol{E}$ in Ω satisfies

$$\nabla \times \nabla \times \boldsymbol{E}(\boldsymbol{r}) - k_0^2 m^2(\boldsymbol{r})\boldsymbol{E}(\boldsymbol{r}) = i\omega\mu_0\boldsymbol{J}(\boldsymbol{r}), \quad \boldsymbol{r} \in \Omega, \tag{2}$$

we have

$$\boldsymbol{E}(\boldsymbol{r}) = i\omega\mu_0 \int_\Omega \boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}')\boldsymbol{J}(\boldsymbol{r}')d\boldsymbol{r}', \tag{3}$$

where the Green's function $\boldsymbol{G}$ satisfies

$$\nabla \times \nabla \times \boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') - k_0^2 m^2(\boldsymbol{r}) \boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') = \boldsymbol{I}\delta(\boldsymbol{r}-\boldsymbol{r}'), \quad \boldsymbol{r}', \boldsymbol{r} \in \Omega, \tag{4}$$

as well as the Sommerfeld radiation condition at infinity. Here the free-space wavenumber $k_0 = \frac{2\pi}{\lambda} = \omega\sqrt{\epsilon_0 \mu_0}$ is a large parameter depending on the angular frequency ω, and λ is the free-space wavelength. So the question is how to compute the dyadic Green's function G.

2.2 Geometrical-optics ansatz for dyadic Green's functions

We are interested in the dyadic Green's function $\boldsymbol{G}$ in equation (4) with a primary source $\boldsymbol{r}' = \boldsymbol{r}_0 = (x_0, y_0, z_0)^T$ in the high frequency regime.

In a homogeneous medium, say $m(\boldsymbol{r}) = m_0$, the dyadic Green's function $\boldsymbol{G}$ has the following explicit form [12]

$$\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0) = \left(\boldsymbol{I} + \frac{\nabla\nabla}{k_0^2 m_0^2}\right)\left(\frac{e^{ik_0 m_0 |\boldsymbol{r}-\boldsymbol{r}_0|}}{4\pi|\boldsymbol{r}-\boldsymbol{r}_0|}\right), \tag{5}$$

which can be expanded to

$$\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0) = \frac{e^{ik_0 m_0 |\boldsymbol{r}-\boldsymbol{r}_0|}}{4\pi|\boldsymbol{r}-\boldsymbol{r}_0|}\left[\boldsymbol{I} - \tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T + \frac{i(\boldsymbol{I} - 3\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T)}{k_0\ m_0|\boldsymbol{r}-\boldsymbol{r}_0|} - \frac{(\boldsymbol{I} - 3\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T)}{k_0^2\ m_0^2|\boldsymbol{r}-\boldsymbol{r}_0|^2}\right], \tag{6}$$

where $\boldsymbol{I}$ is the identity dyad and

$$\tilde{\boldsymbol{r}} = \frac{\boldsymbol{r}-\boldsymbol{r}_0}{|\boldsymbol{r}-\boldsymbol{r}_0|}.$$

In the high frequency regime, k_0 is considered to be large so that we can drop the last two terms inside the square brackets of equation (6) and obtain

$$\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0) \approx \frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}_0|} e^{ik_0 m_0 |\boldsymbol{r}-\boldsymbol{r}_0|}(\boldsymbol{I} - \tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T). \tag{7}$$

Motivated by this, in an inhomogeneous medium, we are looking for the solution of the Green's function $\boldsymbol{G}$ in the form of geometrical-optics (GO) ansatz

$$\begin{aligned} \boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0) &= A(\boldsymbol{r};\boldsymbol{r}_0) e^{ik_0\tau(\boldsymbol{r};\boldsymbol{r}_0)} \boldsymbol{P}(\boldsymbol{r};\boldsymbol{r}_0) + \mathcal{O}\left(\frac{1}{k_0}\right) \\ &\approx A(\boldsymbol{r};\boldsymbol{r}_0) e^{ik_0\tau(\boldsymbol{r};\boldsymbol{r}_0)} \boldsymbol{P}(\boldsymbol{r};\boldsymbol{r}_0), \end{aligned} \tag{8}$$

where A is the amplitude function, τ is the traveltime function, and $\boldsymbol{P}$ is the polarization dyad which is a 3 by 3 dyad of rank 2.

Since the Green's function $\boldsymbol{G}$ is composed of three electric fields, $\boldsymbol{E}_j$, due to the current density $\boldsymbol{J}_j$,

$$\boldsymbol{J}_j(\boldsymbol{r}';\boldsymbol{r}_0) = \frac{\delta(\boldsymbol{r}'-\boldsymbol{r}_0)}{i\omega\mu_0}\boldsymbol{e}_j, \tag{9}$$

where $\boldsymbol{e}_j$ is the j-th column of $\boldsymbol{I}$ for $j = 1,2,3$, we study equations for asymptotic components of each electric field separately. Letting the j-th column of polarization dyad $\boldsymbol{P}$ be the vector $\boldsymbol{p}_j$, we have by equation (8),

$$\boldsymbol{E}_j \approx A(\boldsymbol{r};\boldsymbol{r}_0)e^{ik_0\tau(\boldsymbol{r};\boldsymbol{r}_0)}\boldsymbol{p}_j(\boldsymbol{r};\boldsymbol{r}_0).$$

In the region away from the source $\boldsymbol{r}_0$, plugging the above asymptotic ansatz into equation (2), we obtain

$$\begin{aligned}&(ik_0)^2\left(A\nabla\tau\cdot\nabla\tau\boldsymbol{p}_j - Am^2\boldsymbol{p}_j\right) + \\ &ik_0\left(2\nabla\tau\cdot\nabla A\boldsymbol{p}_j + 2A(\nabla\tau\cdot\nabla)\boldsymbol{p}_j + A\Delta\tau\boldsymbol{p}_j + 2A(\nabla\log m\cdot\boldsymbol{p}_j)\nabla\tau\right) + \\ &\Delta A\boldsymbol{p}_j + 2(\nabla A\cdot\nabla)\boldsymbol{p}_j + A\Delta\boldsymbol{p}_j + 2A\nabla(\nabla\log m\cdot\boldsymbol{p}_j) + 2(\nabla\log m\cdot\boldsymbol{p}_j)\nabla A = 0.\end{aligned} \tag{10}$$

Meanwhile, since $\nabla\cdot(m^2\boldsymbol{E}_j) = 0$ in the source-free region, we have

$$ik_0(m^2A)\nabla\tau\cdot\boldsymbol{p}_j + \nabla(m^2A)\cdot\boldsymbol{p}_j + m^2A\nabla\cdot\boldsymbol{p}_j = 0. \tag{11}$$

Equating to zero the coefficients of k_0^2 and k_0 in equations (10) and (11), we obtain the eikonal equation for the traveltime function τ,

$$|\nabla\tau(\boldsymbol{r};\boldsymbol{r}_0)| = m(\boldsymbol{r}), \tag{12}$$

and equations for A and $\boldsymbol{p}_j$,

$$2(\nabla\tau\cdot\nabla A)\boldsymbol{p}_j + 2A(\nabla\tau\cdot\nabla)\boldsymbol{p}_j + A\Delta\tau\boldsymbol{p}_j + 2A(\nabla m\cdot\boldsymbol{p}_j)\frac{\nabla\tau}{m} = 0, \tag{13}$$

and

$$\nabla\tau\cdot\boldsymbol{p}_j = 0. \tag{14}$$

By applying the method of characteristics to the eikonal equation (12), the ray trajectory emanating from the source $\boldsymbol{r}_0$ and passing through $\boldsymbol{r}$ satisfies

$$\dot{\boldsymbol{r}} = \frac{\nabla\tau(\boldsymbol{r};\boldsymbol{r}_0)}{m(\boldsymbol{r})} = \boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0), \tag{15}$$

$$\frac{d}{ds}[\nabla\tau(\boldsymbol{r};\boldsymbol{r}_0)] = \nabla m(\boldsymbol{r}), \tag{16}$$

where the over-dot means differentiation with respect to the arclength s, and $\boldsymbol{t}^{(1)}$ denotes the ray tangent and also the arrival direction at $\boldsymbol{r}$. By equations (14) and (15), we see that the polarization vector $\boldsymbol{p}_j$ along each ray is perpendicular to the ray tangent. Assuming further that the polarization vector $\boldsymbol{p}_j(\boldsymbol{r};\boldsymbol{r}_0)$ has a constant norm along $\boldsymbol{r}(s)$, we have

$$\boldsymbol{p}_j \cdot \left(\frac{\nabla\tau}{m} \cdot \nabla\right) \boldsymbol{p}_j = \boldsymbol{p}_j \cdot (\dot{\boldsymbol{r}} \cdot \nabla)\, \boldsymbol{p}_j = \boldsymbol{p}_j \cdot \dot{\boldsymbol{p}}_j = 0. \tag{17}$$

Therefore, taking dot product of both sides of equation (13) with $\boldsymbol{p}_j$ and using equations (14) and (17), we obtain the transport equation for the amplitude function A,

$$\nabla\tau(\boldsymbol{r};\boldsymbol{r}_0) \cdot \nabla A(\boldsymbol{r};\boldsymbol{r}_0) + \frac{1}{2} A(\boldsymbol{r};\boldsymbol{r}_0)\Delta\tau(\boldsymbol{r};\boldsymbol{r}_0) = 0. \tag{18}$$

It follows that equation (13) can be reduced to

$$m(\boldsymbol{r})(\nabla\tau(\boldsymbol{r};\boldsymbol{r}_0) \cdot \nabla)\boldsymbol{p}_j(\boldsymbol{r};\boldsymbol{r}_0) + (\nabla m(\boldsymbol{r}) \cdot \boldsymbol{p}_j(\boldsymbol{r};\boldsymbol{r}_0))\nabla\tau(\boldsymbol{r};\boldsymbol{r}_0) = 0; \tag{19}$$

due to equations (14) and (15), this can be further written as

$$\dot{\boldsymbol{p}}_j = -(\dot{\boldsymbol{t}}^{(1)} \cdot \boldsymbol{p}_j)\boldsymbol{t}^{(1)}, \tag{20}$$

indicating that along the ray $\boldsymbol{r}(s)$, $\dot{\boldsymbol{p}}_j$ is parallel to the ray tangent $\boldsymbol{t}^{(1)}$.

In a small neighborhood of the primary source $\boldsymbol{r}_0$, $m(\boldsymbol{r})$ approximately equals $m(\boldsymbol{r}_0)$ so that we expect that the GO ansatz (8) is consistent with the asymptotic ansatz (7) for a constant index of refraction $m(\boldsymbol{r}_0)$. Therefore, we enforce the following initial conditions at the primary source $\boldsymbol{r}_0$:

$$\lim_{\boldsymbol{r}\to\boldsymbol{r}_0} \left(\frac{\tau(\boldsymbol{r};\boldsymbol{r}_0)}{|\boldsymbol{r}-\boldsymbol{r}_0|} - m(\boldsymbol{r}_0)\right) = 0, \tag{21}$$

$$\lim_{\boldsymbol{r}\to\boldsymbol{r}_0} \left(A(\boldsymbol{r};\boldsymbol{r}_0) - \frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}_0|}\right) = 0, \tag{22}$$

and

$$\lim_{\boldsymbol{r}\to\boldsymbol{r}_0} \left(\boldsymbol{p}_j(\boldsymbol{r};\boldsymbol{r}_0) - (\boldsymbol{I} - \tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T)\boldsymbol{e}_j\right) = 0, \tag{23}$$

respectively. In the above limits, $\boldsymbol{r} \to \boldsymbol{r}_0$ should be understood in the sense that $\boldsymbol{r}$ approaches $\boldsymbol{r}_0$ along its ray $\boldsymbol{r}(s)$ as $s \to 0^+$. Consequently, imposing these three source-point conditions, we can solve equation (12) for traveltime τ, equation (18) for amplitude A, and equations (14), (17) and (19) for polarization vectors $\boldsymbol{p}_j$.

Since coefficients in equations (14), (17), (18), and (19) depend on traveltime τ, we compute τ first, and then the amplitude A and polarization vectors $\boldsymbol{p}_j$. However, the complicated PDE system of equations (14), (17) and (19) with the boundary condition (23) make computing $\boldsymbol{p}_j$ difficult.

Extending an approach due to Popov and Psencik (Section 2 in [40]), we next show how to efficiently compute the polarization dyad $\boldsymbol{P}$ by first introducing an easily computed auxiliary moving frame.

2.3 Constructing the polarization dyad $\boldsymbol{P}$

The two properties of $\boldsymbol{p}_j$, the perpendicularity to the ray tangent (14) and the norm invariance (17) along the ray $\boldsymbol{r}(s)$, imply that we can rewrite the polarization vector $\boldsymbol{p}_j$ as follows:

$$\boldsymbol{p}_j(\boldsymbol{r};\boldsymbol{r}_0) = p_j\left(\boldsymbol{B}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0)\cos\phi(\boldsymbol{r};\boldsymbol{r}_0) - \boldsymbol{C}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0)\sin\phi(\boldsymbol{r};\boldsymbol{r}_0)\right), \tag{24}$$

where $p_j \geq 0$ is a constant. Here $\boldsymbol{B}^{(1)}$ and $\boldsymbol{C}^{(1)}$ are vectors defined through

$$\boldsymbol{B}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0) = \frac{\boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0)\times\boldsymbol{\ell}}{|\boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0)\times\boldsymbol{\ell}|}, \quad \boldsymbol{C}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0) = \boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0)\times\boldsymbol{B}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0), \tag{25}$$

with the reference vector $\boldsymbol{\ell}$ selected according to certain rules which will be detailed later, so that $\{\boldsymbol{t}^{(1)}, \boldsymbol{B}^{(1)}, \boldsymbol{C}^{(1)}\}$ form a moving orthogonal coordinate system along the ray $\boldsymbol{r}(s)$, and the unknown ϕ is the rotation angle from $\boldsymbol{B}^{(1)}$ to $\boldsymbol{p}_j$ in a counterclockwise direction in the plane perpendicular to $\boldsymbol{t}^{(1)}$.

Clearly, $\boldsymbol{p}_j$ in equation (24) of a constant norm p_j is perpendicular to the ray tangent $\boldsymbol{t}^{(1)}$. To determine the scalar unknown $\phi(\boldsymbol{r};\boldsymbol{r}_0)$, we make use of the third property, equation (20).

Along $\boldsymbol{r}(s)$, the derivative of $\boldsymbol{p}_j$ with respect to the arclength s satisfies

$$\dot{\boldsymbol{p}}_j = p_j\left(-\boldsymbol{B}^{(1)}\sin\phi\,\dot{\phi} + \dot{\boldsymbol{B}}^{(1)}\cos\phi - \boldsymbol{C}^{(1)}\cos\phi\,\dot{\phi} - \dot{\boldsymbol{C}}^{(1)}\sin\phi\right). \tag{26}$$

From equation (20), we have $\dot{\boldsymbol{p}}_j\cdot\boldsymbol{B}^{(1)} = 0$ and $\dot{\boldsymbol{p}}_j\cdot\boldsymbol{C}^{(1)} = 0$. Using the following relations,

$$\begin{aligned}
&\boldsymbol{B}^{(1)}\cdot\dot{\boldsymbol{B}}^{(1)} = 0, \quad \boldsymbol{C}^{(1)}\cdot\dot{\boldsymbol{C}}^{(1)} = 0,\\
&\boldsymbol{B}^{(1)}\cdot\boldsymbol{C}^{(1)} = 0, \quad \boldsymbol{B}^{(1)}\cdot\dot{\boldsymbol{C}}^{(1)} = -\boldsymbol{C}^{(1)}\cdot\dot{\boldsymbol{B}}^{(1)},
\end{aligned}$$

we obtain

$$\dot{\phi} = \boldsymbol{C}^{(1)}\cdot\dot{\boldsymbol{B}}^{(1)}. \tag{27}$$

Using equations (15) and (16), we simplify the above equation and arrive at the following governing equation for ϕ,

$$\frac{\nabla\tau}{m}\cdot\nabla\phi = \dot{\phi} = \frac{\left[-(\boldsymbol{t}^{(1)}\cdot\boldsymbol{\ell})\nabla\log m + \dot{\boldsymbol{\ell}}\right]\cdot(\boldsymbol{t}^{(1)}\times\boldsymbol{\ell})}{|\boldsymbol{t}^{(1)}\times\boldsymbol{\ell}|^2}. \tag{28}$$

Since $\phi(\boldsymbol{r};\boldsymbol{r}_0)$ at the source $\boldsymbol{r}_0$, say $\phi^{(0)}(\boldsymbol{r};\boldsymbol{r}_0) = \lim_{s\to 0^+}\phi(\boldsymbol{r};\boldsymbol{r}_0)$, varies depending on ray $\boldsymbol{r}(s)$, we compute the difference $\Phi(\boldsymbol{r};\boldsymbol{r}_0) = \phi(\boldsymbol{r};\boldsymbol{r}_0) - \phi^{(0)}$ instead of ϕ, satisfying the same equation (28) with the zero initial condition $\lim_{\boldsymbol{r}\to\boldsymbol{r}_0}\Phi(\boldsymbol{r};\boldsymbol{r}_0) = 0$.

To simplify the problem, we further assume that the vector $\boldsymbol{\ell}$ does not change along each ray $\boldsymbol{r}(s)$ so that Φ is governed by

$$\nabla\tau\cdot\nabla\Phi = m\dot{\Phi} = \frac{\left[-(\boldsymbol{t}^{(1)}\cdot\boldsymbol{\ell})\nabla m\right]\cdot(\boldsymbol{t}^{(1)}\times\boldsymbol{\ell})}{|\boldsymbol{t}^{(1)}\times\boldsymbol{\ell}|^2}. \tag{29}$$

To establish the relation between $\boldsymbol{p}_j$ and Φ, we derive the following equation. Along the ray $\boldsymbol{r}=\boldsymbol{r}(s)$, by (15), (21), and the L'hopital's rule, we have

$$\begin{aligned}
&\lim_{s\to 0^+}\tilde{\boldsymbol{r}}(s)=\lim_{s\to 0^+}\frac{\boldsymbol{r}(s)-\boldsymbol{r}_0}{|\boldsymbol{r}(s)-\boldsymbol{r}_0|}=m(\boldsymbol{r}_0)\lim_{s\to 0^+}\frac{\boldsymbol{r}(s)-\boldsymbol{r}_0}{\tau(\boldsymbol{r}(s);\boldsymbol{r}_0)}\\
&=m(\boldsymbol{r}_0)\lim_{s\to 0^+}\frac{\dot{\boldsymbol{r}}(s)}{\nabla\tau(\boldsymbol{r}(s);\boldsymbol{r}_0)\cdot\dot{\boldsymbol{r}}(s)}=\lim_{s\to 0^+}\frac{\dot{\boldsymbol{r}}(s)}{\dot{\boldsymbol{r}}(s)\cdot\dot{\boldsymbol{r}}(s)}=\lim_{s\to 0^+}\dot{\boldsymbol{r}}(s),
\end{aligned}$$

which is the take-off direction of the ray $\boldsymbol{r}=\boldsymbol{r}(s)$ from $\boldsymbol{r}_0$, denoted by $\boldsymbol{t}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0)$.

Since the take-off direction $\boldsymbol{t}^{(0)}$ varies depending on rays and does not change along each ray $\boldsymbol{r}(s)$, i.e. $\dot{\boldsymbol{t}}^{(0)}=0$, we have

$$(\nabla\tau\cdot\nabla)\boldsymbol{t}^{(0)}=0, \tag{30}$$

with the initial condition

$$\lim_{\boldsymbol{r}\to\boldsymbol{r}_0}\left[\boldsymbol{t}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0)-\tilde{\boldsymbol{r}}\right]=\lim_{\boldsymbol{s}\to 0^+}\left[\boldsymbol{t}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0)-\tilde{\boldsymbol{r}}\right]=0. \tag{31}$$

Once $\boldsymbol{t}^{(0)}$ is found, we have by the initial condition (23)

$$\lim_{s\to 0^+}\boldsymbol{p}_j(\boldsymbol{r}(s);\boldsymbol{r}_0)=\lim_{s\to 0^+}(\boldsymbol{I}-\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T)\boldsymbol{e}_j=[\boldsymbol{I}-\boldsymbol{t}^{(0)}\boldsymbol{t}^{(0)^T}]\boldsymbol{e}_j. \tag{32}$$

Let vectors $\boldsymbol{B}^{(0)}$ and $\boldsymbol{C}^{(0)}$ be defined through (25) with $\boldsymbol{t}^{(1)}$ replaced by $\boldsymbol{t}^{(0)}$. By equation (24),

$$\lim_{s\to 0^+}\boldsymbol{p}_j(\boldsymbol{r}(s);\boldsymbol{r}_0)=p_j\left(\boldsymbol{B}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0)\cos\phi^{(0)}-\boldsymbol{C}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0)\sin\phi^{(0)}\right). \tag{33}$$

Thus, by equation (32), we have

$$[\boldsymbol{I}-\boldsymbol{t}^{(0)}\boldsymbol{t}^{(0)^T}]\boldsymbol{e}_j=p_j\left[\begin{array}{ccc}\boldsymbol{t}^{(0)} & \boldsymbol{B}^{(0)} & \boldsymbol{C}^{(0)}\end{array}\right]\left[\begin{array}{c}0\\ \cos\phi^{(0)}\\ -\sin\phi^{(0)}\end{array}\right]. \tag{34}$$

Therefore, equation (24) becomes

$$\begin{aligned}
\boldsymbol{p}_j &= p_j\left[\begin{array}{ccc}\boldsymbol{t}^{(1)} & \boldsymbol{B}^{(1)} & \boldsymbol{C}^{(1)}\end{array}\right]\left[\begin{array}{c}0\\ \cos(\Phi+\phi^{(0)})\\ -\sin(\Phi+\phi^{(0)})\end{array}\right]\\
&= p_j\left[\begin{array}{ccc}\boldsymbol{t}^{(1)} & \boldsymbol{B}^{(1)} & \boldsymbol{C}^{(1)}\end{array}\right]\left[\begin{array}{ccc}1 & 0 & 0\\ 0 & \cos\Phi & \sin\Phi\\ 0 & -\sin\Phi & \cos\Phi\end{array}\right]\left[\begin{array}{c}0\\ \cos\phi^{(0)}\\ -\sin\phi^{(0)}\end{array}\right]
\end{aligned}$$

$$= \begin{bmatrix} \boldsymbol{t}^{(1)} & \boldsymbol{B}^{(1)} & \boldsymbol{C}^{(1)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \begin{bmatrix} \boldsymbol{t}^{(0)^T} \\ \boldsymbol{B}^{(0)^T} \\ \boldsymbol{C}^{(0)^T} \end{bmatrix} [\boldsymbol{I} - \boldsymbol{t}^{(0)}\boldsymbol{t}^{(0)^T}]\boldsymbol{e}_j \tag{35}$$

so that the polarization dyad $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3]^T$ satisfies

$$\begin{aligned} \boldsymbol{P} &= \begin{bmatrix} \boldsymbol{t}^{(1)} & \boldsymbol{B}^{(1)} & \boldsymbol{C}^{(1)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \begin{bmatrix} \boldsymbol{t}^{(0)^T} \\ \boldsymbol{B}^{(0)^T} \\ \boldsymbol{C}^{(0)^T} \end{bmatrix} [\boldsymbol{I} - \boldsymbol{t}^{(0)}\boldsymbol{t}^{(0)^T}] \\ &= \left(\boldsymbol{B}^{(1)}\boldsymbol{B}^{(0)^T} + \boldsymbol{C}^{(1)}\boldsymbol{C}^{(0)^T}\right)\cos\Phi + \left(\boldsymbol{B}^{(1)}\boldsymbol{C}^{(0)^T} - \boldsymbol{C}^{(1)}\boldsymbol{B}^{(0)^T}\right)\sin\Phi. \end{aligned} \tag{36}$$

Consequently, the electric Green's function $\boldsymbol{G}$ takes the following form

$$\boldsymbol{G} \approx Ae^{ik_0\tau}\left[\left(\boldsymbol{B}^{(1)}\boldsymbol{B}^{(0)^T} + \boldsymbol{C}^{(1)}\boldsymbol{C}^{(0)^T}\right)\cos\Phi + \left(\boldsymbol{B}^{(1)}\boldsymbol{C}^{(0)^T} - \boldsymbol{C}^{(1)}\boldsymbol{B}^{(0)^T}\right)\sin\Phi\right]. \tag{37}$$

In summary, to compute $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ by equation (37), we need to carry out the following steps: first, solve the eikonal equation (12) with the initial condition (21) for traveltime τ and obtain the arrival direction $\boldsymbol{t}^{(1)}$ by equation (15), and the related two orthonormal vectors $\boldsymbol{B}^{(1)}$ and $\boldsymbol{C}^{(1)}$; next, solve the transport equation (18) with the initial condition (22) for amplitude A, and solve equation (30) with the initial condition (31) for the take-off direction $\boldsymbol{t}^{(0)}$ so as to obtain the two orthonormal vectors $\boldsymbol{B}^{(0)}$ and $\boldsymbol{C}^{(0)}$; finally, solve equation(29) with the zero initial condition for Φ.

2.4 Rule of choosing the reference vector $\boldsymbol{\ell}$

According to the previous subsection, we enforce the following properties on the vector $\boldsymbol{\ell}$: the vector $\boldsymbol{\ell}$ is not allowed to be parallel to $\boldsymbol{t}^{(i)}$ so that $\boldsymbol{B}^{(i)}$ and $\boldsymbol{C}^{(i)}$ in equation (37) are well-defined for $i = 0$, 1, and the vector $\boldsymbol{\ell}$ is not allowed to change along each ray so that equation (29) holds.

In practice, we choose the vector $\boldsymbol{\ell}$ to be some $\boldsymbol{e}_j$ for $j = 1, 2, 3$ by the following rule: at any point $\boldsymbol{r} \neq \boldsymbol{r}_0$, along the ray traced out from $\boldsymbol{r}_0$ to $\boldsymbol{r}$:

1. if $|\boldsymbol{t}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0) \cdot \boldsymbol{e}_1| \le \sqrt{2}/2$ and $|\boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0) \cdot \boldsymbol{e}_1| \le \sqrt{2}/2$, then $\boldsymbol{\ell}(\boldsymbol{r};\boldsymbol{r}_0) = \boldsymbol{e}_1$;
2. else if $|\boldsymbol{t}^{(0)}(\boldsymbol{r};\boldsymbol{r}_0) \cdot \boldsymbol{e}_2| \le \sqrt{2}/2$ and $|\boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}_0) \cdot \boldsymbol{e}_2| \le \sqrt{2}/2$, then $\boldsymbol{\ell}(\boldsymbol{r};\boldsymbol{r}_0) = \boldsymbol{e}_2$;
3. else, $\boldsymbol{\ell}(\boldsymbol{r};\boldsymbol{r}_0) = \boldsymbol{e}_3$.

Consequently, it is not difficult to check that $\boldsymbol{\ell}$ is never parallel to $\boldsymbol{t}^{(0)}$ and $\boldsymbol{t}^{(1)}$ at any point $\boldsymbol{r}$. However, two potential problems may occur: first, along different rays, the vector $\boldsymbol{\ell}$ may vary; second, along a certain ray which is very curved, $\boldsymbol{\ell}$ may change from one direction to another so that the vector $\boldsymbol{\ell}$ becomes piecewisely discontinuous.

Therefore, equation (29) holds except at discontinuities so that a difficulty arises for the computation of Φ. To tackle this difficulty, we propose an effective numerical algorithm for solving equation (29).

For simplicity, we will use $\Phi(\boldsymbol{r})$ to denote $\Phi(\boldsymbol{r};\boldsymbol{r}_0)$ in the following. Suppose that at a certain stage we need to use the available value of $\Phi(\boldsymbol{r}_{j-1})$ to compute $\Phi(\boldsymbol{r}_j)$, but $\boldsymbol{\ell}(\boldsymbol{r}_{j-1})$ is different from $\boldsymbol{\ell}(\boldsymbol{r}_j)$ due to one of the two mentioned problems; for instance, $\boldsymbol{\ell}(\boldsymbol{r}_{j-1}) = \boldsymbol{e}_1$ and $\boldsymbol{\ell}(\boldsymbol{r}_j) = \boldsymbol{e}_2$. We cannot use $\Phi(\boldsymbol{r}_{j-1})$ directly since the function Φ depends on the choice of $\boldsymbol{\ell}$ and represents different functions for different $\boldsymbol{\ell}$; for the sake of clarity we rewrite it as $\Phi(\boldsymbol{r};\boldsymbol{\ell})$ in the following. Our approach is applying a transformation to obtain $\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_2)$ from $\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_1)$, and this can be further used to compute $\Phi(\boldsymbol{r}_j;\boldsymbol{e}_2)$.

Assume that $\{\boldsymbol{t}^{(\nu)}, \boldsymbol{B}^{(\nu)}, \boldsymbol{C}^{(\nu)}\}$ and $\{\boldsymbol{t}^{(\nu)}, \tilde{\boldsymbol{B}}^{(\nu)}, \tilde{\boldsymbol{C}}^{(\nu)}\}$ are two orthogonal coordinate systems at $\boldsymbol{r}_{j-1}$ constructed from equation (25) with $\boldsymbol{\ell} = \boldsymbol{e}_1$ and $\boldsymbol{\ell} = \boldsymbol{e}_2$, respectively, for $\nu = 0, 1$. Then, there exist two angles $\alpha^{(0)}$ and $\alpha^{(1)}$ such that

$$\begin{bmatrix} \boldsymbol{t}^{(\nu)} & \boldsymbol{B}^{(\nu)} & \boldsymbol{C}^{(\nu)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}^{(\nu)} & \tilde{\boldsymbol{B}}^{(\nu)} & \tilde{\boldsymbol{C}}^{(\nu)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha^{(\nu)} & \sin\alpha^{(\nu)} \\ 0 & -\sin\alpha^{(\nu)} & \cos\alpha^{(\nu)} \end{bmatrix}, \tag{38}$$

for $\nu = 0, 1$, where $\alpha^{(\nu)}$ represents the rotation angle from $\tilde{\boldsymbol{B}}^{(\nu)}$ to $\boldsymbol{B}^{(\nu)}$ in the plane perpendicular to $\boldsymbol{t}^{(\nu)}$.

Since the polarization dyad does not depend on the choice of the vector $\boldsymbol{\ell}$, we have by equations (36) and (38) that

$$\begin{aligned} \boldsymbol{P} \; = \; & \begin{bmatrix} \boldsymbol{t}^{(1)} & \tilde{\boldsymbol{B}}^{(1)} & \tilde{\boldsymbol{C}}^{(1)} \end{bmatrix} \\ & \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_1) + \alpha^{(1)} - \alpha^{(0)}) & \sin(\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_1) + \alpha^{(1)} - \alpha^{(0)}) \\ 0 & -\sin(\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_1) + \alpha^{(1)} - \alpha^{(0)}) & \cos(\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_1) + \alpha^{(1)} - \alpha^{(0)}) \end{bmatrix} \\ & \begin{bmatrix} \boldsymbol{t}^{(0)^T} \\ \tilde{\boldsymbol{B}}^{(0),T} \\ \tilde{\boldsymbol{C}}^{(0),T} \end{bmatrix} [\boldsymbol{I} - \boldsymbol{t}^{(0)}\boldsymbol{t}^{(0)^T}], \end{aligned} \tag{39}$$

indicating that

$$\Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_2) = \Phi(\boldsymbol{r}_{j-1};\boldsymbol{e}_1) + \alpha^{(1)} - \alpha^{(0)}. \tag{40}$$

Therefore, even though equation (29) does not hold at the discontinuities of $\boldsymbol{\ell}$ according to the rule of choosing the reference vector $\boldsymbol{\ell}$, the above transformation provides an effective way to change values of $\boldsymbol{\ell}$ in a small neighborhood of each discontinuity so that all discontinuities can be removed. Consequently, our iterative numerical scheme works everywhere in the computational domain.

2.5 Locally valid asymptotic solutions

Due to the Laplacian $\Delta\tau$ in equation (18), we have to compute a third-order accurate traveltime field so that a first-order accurate amplitude field can be obtained [42]. Furthermore, all four unknowns, traveltime, amplitude, take-off direction, and polarization angle, have upwind singularities at the primary source $\boldsymbol{r}_0$ as the distance-like traveltime is not differentiable at $\boldsymbol{r}_0$. Without special treatment, this upwind singularities can induce large truncation error nearby the source, which can propagate over the entire computational domain so that high-order methods can be influenced to have at most first-order accuracy. Therefore, to treat the upwind singularities, we utilize the factorization ideas in [26, 30, 29, 28] and the references therein to develop high-order WENO based Lax-Friedrichs schemes for computing τ, A, $\boldsymbol{t}^{(0)}$, and Φ. Consequently, with the GO large-k_0 ansatz (8) at our disposal, we can construct an asymptotic GO solution for the Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ in the whole computational domain.

However, since the viscosity-solution for the first-arrival traveltime field does not contain later-arriving traveltimes [26, 30, 29, 28, 23], the constructed asymptotic Green's function cannot capture the interference effects due to multivaluedness of the traveltime field and therefore is only valid before the occurrence of caustics. To get back the interference effects, we utilize the Huygens secondary source principle as the backbone to relay the information carried by locally valid asymptotic solutions, and the Huygens-Kirchhoff integration formula serves as the relay station.

As the polarization angle Φ depends closely on the ray-dependent vector $\boldsymbol{\ell}$, we will use the $\boldsymbol{\ell}$-independent polarization dyad $\boldsymbol{P}$ instead of Φ and will refer to traveltime τ, amplitude A, take-off direction $\boldsymbol{t}^{(0)}$, and polarization dyad $\boldsymbol{P}$ as the four GO asymptotic ingredients in the following.

3 Huygens' principle based globally valid Green's functions

3.1 Huygens-Kirchhoff formula

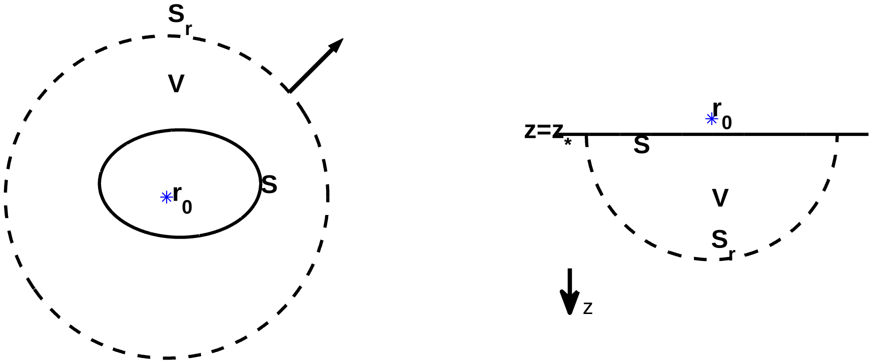
Consider an exterior region V bounded by two closed surfaces, an inner surface S and an outer sphere S_r of radius r centered at $\boldsymbol{r}_0=(x_0,y_0,z_0)^T$, both of which enclose the primary source $\boldsymbol{r}_0$, as shown in Figure 1(a).

By equation (9), the j-th column of the Green's function $\boldsymbol{G}$, the electric field $\boldsymbol{E}_j$ due to the current density $\boldsymbol{J}_j=0$ in V, satisfies

$$\nabla\times\nabla\times\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)-k_0^2m^2(\boldsymbol{r})\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)=0,\quad \boldsymbol{r}\in V, \tag{41}$$

for $j=1,2,3$. Thus, for any $\boldsymbol{r}'\in V$, we have by equation (4)

$$\begin{aligned}\boldsymbol{E}_j(\boldsymbol{r}';\boldsymbol{r}_0) &= \int_V \boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\delta(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r},\\ &= \int_V \boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\left[\nabla\times\nabla\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')-k_0^2m^2(\boldsymbol{r})\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\right]d\boldsymbol{r},\end{aligned}$$



(a) (b)

Figure 1: The schematic view of Huygens' Principle.

$$\begin{aligned} &= \int_V \left[\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\nabla\times\nabla\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') - \nabla\times\nabla\times\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\right] d\boldsymbol{r} \\ &= -\oint_{S+S_r} \boldsymbol{n}(\boldsymbol{r})\cdot\left[\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\times\nabla\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') + \nabla\times\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\right] dS(\boldsymbol{r}), \end{aligned} \tag{42}$$

where the Gauss divergence theorem is applied in the last equality, and $\boldsymbol{n}(\boldsymbol{r})$ denotes the outward unit-normal vector to the boundary of V. Next, by letting $r\to\infty$ and by virtue of the Sommerfeld radiation condition, the surface integration over S_r vanishes so that we obtain the Huygens-Kirchhoff formula [12, 52]

$$\boldsymbol{E}_j(\boldsymbol{r}';\boldsymbol{r}_0) = -\oint_S \boldsymbol{n}(\boldsymbol{r})\cdot\left[\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\times\nabla\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') + \nabla\times\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\right] dS(\boldsymbol{r}), \tag{43}$$

for $\boldsymbol{r}'\in\Omega$, the exterior domain outside the surface S. This formula indicates that the electric field $\boldsymbol{E}_j(\boldsymbol{r}';\boldsymbol{r}_0)$ away from S can be determined if $\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)$ and the electric Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')$ are available. By using the GO large-k_0 ansatz for the source $\boldsymbol{r}'$ to approximate $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')$, the curl of the electric Green's function can be approximated as,

$$\begin{aligned} \nabla\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') &\approx ik_0\nabla\tau(\boldsymbol{r};\boldsymbol{r}')\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') + e^{ik_0\tau(\boldsymbol{r};\boldsymbol{r}')}\nabla\times[A(\boldsymbol{r};\boldsymbol{r}')\boldsymbol{P}(\boldsymbol{r};\boldsymbol{r}')] \\ &\approx ik_0\nabla\tau(\boldsymbol{r};\boldsymbol{r}')\times\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}'), \end{aligned} \tag{44}$$

where we retain the leading order term only. Furthermore, by using the following vector relations

$$\begin{aligned} \boldsymbol{c}\cdot(\boldsymbol{a}\times\boldsymbol{b}) &= (\boldsymbol{c}\times\boldsymbol{a})\cdot\boldsymbol{b}, \\ (\boldsymbol{a}\times\boldsymbol{b})\cdot(\boldsymbol{c}\times\boldsymbol{d}) &= (\boldsymbol{a}\cdot\boldsymbol{c})(\boldsymbol{b}\cdot\boldsymbol{d}) - (\boldsymbol{a}\cdot\boldsymbol{d})(\boldsymbol{b}\cdot\boldsymbol{c}), \end{aligned}$$

for $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ and $\boldsymbol{d} \in \mathbb{R}^3$, equation (43) becomes

$$\begin{aligned} \boldsymbol{E}_j(\boldsymbol{r}';\boldsymbol{r}_0) \;\approx\; & -\oint_S \Big[\, ik_0\big(\boldsymbol{n}(\boldsymbol{r})\cdot\nabla\tau(\boldsymbol{r};\boldsymbol{r}')\big)\big(\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\big) \\ & -ik_0\big(\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\nabla\tau(\boldsymbol{r};\boldsymbol{r}')\big)\big(\boldsymbol{n}(\boldsymbol{r})\cdot\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\big) \\ & +\big(\boldsymbol{n}(\boldsymbol{r})\times\nabla\times\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\big)\cdot\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')\Big]dS(\boldsymbol{r}). \end{aligned} \tag{45}$$

Apparently, it is numerically expensive to compute $\nabla\tau(\boldsymbol{r};\boldsymbol{r}')$ and $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}')$ for $\boldsymbol{r} \in S$ excited by sources $\boldsymbol{r}' \in \Omega$ since Ω is a manifold of higher dimension than S. However, we can use the reciprocity to interchange the two arguments $\boldsymbol{r}$ and $\boldsymbol{r}'$ so that sources on S are involved instead. Specifically, the reciprocity states that

$$\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}') = \boldsymbol{G}^T(\boldsymbol{r}';\boldsymbol{r}), \tag{46}$$

and the arrival direction of the ray from the source $\boldsymbol{r}'$ to the receiver $\boldsymbol{r}$ is opposite to the take-off direction of the ray from the source $\boldsymbol{r}$ to the receiver $\boldsymbol{r}'$, yielding

$$\frac{\nabla\tau(\boldsymbol{r};\boldsymbol{r}')}{m(\boldsymbol{r})} = \boldsymbol{t}^{(1)}(\boldsymbol{r};\boldsymbol{r}') = -\boldsymbol{t}^{(0)}(\boldsymbol{r}';\boldsymbol{r}). \tag{47}$$

Thus, we have

$$\begin{aligned} \boldsymbol{E}_j(\boldsymbol{r}';\boldsymbol{r}_0) \;\approx\; & \oint_S \Big[\, ik_0 m(\boldsymbol{r})\big(\boldsymbol{n}(\boldsymbol{r})\cdot\boldsymbol{t}^{(0)}(\boldsymbol{r}';\boldsymbol{r})\big)\big(\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\boldsymbol{G}^T(\boldsymbol{r}';\boldsymbol{r})\big) \\ & -ik_0 m(\boldsymbol{r})\big(\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\cdot\boldsymbol{t}^{(0)}(\boldsymbol{r}';\boldsymbol{r})\big)\big(\boldsymbol{n}(\boldsymbol{r})\cdot\boldsymbol{G}^T(\boldsymbol{r}';\boldsymbol{r})\big) \\ & -\big(\boldsymbol{n}(\boldsymbol{r})\times\nabla\times\boldsymbol{E}_j(\boldsymbol{r};\boldsymbol{r}_0)\big)\cdot\boldsymbol{G}^T(\boldsymbol{r}';\boldsymbol{r})\Big]dS(\boldsymbol{r}), \end{aligned} \tag{48}$$

for $\boldsymbol{r}' \in \Omega$ and for $j = 1, 2, 3$. This is the Huygens-Kirchhoff formula for a bounded surface S enclosing the source $\boldsymbol{r}_0$.

To build up the Huygens-Kirchhoff formula for a wave field excited by the primary source $\boldsymbol{r}_0 = [x_0, y_0, z_0]^T$ when S is unbounded, such as a plane $z = z^*$ with $z^* > z_0$ as shown in Figure 1(b), we take V to be the region bounded by the disk S at $z = z^*$ centered at (x_0, y_0, z^*) of radius r and the half sphere S_r above $z = z^*$ centered at (x_0, y_0, z^*) of the same radius r. Using the same analysis as for the case of S being bounded, we obtain the same formula (48), where S becomes the infinite plane $z = z^*$, Ω becomes the region above S: $z > z^*$, and $\boldsymbol{n} = [0, 0, -1]^T$. Consequently, by equation (48), the (k, j) entry of the Green's function $\boldsymbol{G}(\boldsymbol{r}';\boldsymbol{r}_0)$, the k-th element of $\boldsymbol{E}_j(\boldsymbol{r}';\boldsymbol{r}_0)$, can be expressed as

$$\begin{aligned} \boldsymbol{G}_{kj}(\boldsymbol{r}';\boldsymbol{r}_0) \;\approx\; & \oint_S \Big[ik_0 m(\boldsymbol{r})\ \big(\boldsymbol{G}_{k3}(\boldsymbol{r}';\boldsymbol{r})\ \boldsymbol{t}_1^{(0)}(\boldsymbol{r}';\boldsymbol{r}) - \boldsymbol{G}_{k1}(\boldsymbol{r}';\boldsymbol{r})\ \boldsymbol{t}_3^{(0)}(\boldsymbol{r}';\boldsymbol{r})\big)\ \boldsymbol{G}_{1j}(\boldsymbol{r};\boldsymbol{r}_0) \\ & +ik_0 m(\boldsymbol{r})\ \big(\boldsymbol{G}_{k3}(\boldsymbol{r}';\boldsymbol{r})\ \boldsymbol{t}_2^{(0)}(\boldsymbol{r}';\boldsymbol{r}) - \boldsymbol{G}_{k2}(\boldsymbol{r}';\boldsymbol{r})\ \boldsymbol{t}_3^{(0)}(\boldsymbol{r}';\boldsymbol{r})\big)\ \boldsymbol{G}_{2j}(\boldsymbol{r};\boldsymbol{r}_0) \\ & +\boldsymbol{G}_{k1}(\boldsymbol{r}';\boldsymbol{r})\ \big(\partial_x\boldsymbol{G}_{3j}(\boldsymbol{r};\boldsymbol{r}_0) - \partial_z\boldsymbol{G}_{1j}(\boldsymbol{r};\boldsymbol{r}_0)\big) \\ & +\boldsymbol{G}_{k2}(\boldsymbol{r}';\boldsymbol{r})\ \big(\partial_y\boldsymbol{G}_{3j}(\boldsymbol{r};\boldsymbol{r}_0) - \partial_z\boldsymbol{G}_{2j}(\boldsymbol{r};\boldsymbol{r}_0)\big)\Big]dS(\boldsymbol{r}), \end{aligned} \tag{49}$$

for $k, j = 1, 2, 3$, where $\boldsymbol{t}_j^{(0)}$ denotes the j-th element of the take-off direction $\boldsymbol{t}^{(0)}$. Formulas for $z^* < z_0$ can be derived similarly.

3.2 Huygens principle based sweeping method

Based on the Huygens-Kirchhoff formula (48), we now develop a layer-by-layer sweeping method to construct the globally valid Green's function.

Since the viscosity solution based GO large-k_0 ansatz (8) is valid in a local neighborhood of $\boldsymbol{r}_0$, regarded as the first layer Ω_1, the Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ and its gradient $\nabla\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ are available on the boundary of Ω_1, denoted by S_1. Then, we set up secondary source points on S_1, and identify a narrow layer where the asymptotic Green's function $\boldsymbol{G}$ excited by every selected secondary source point $\boldsymbol{r} \in S_1$ is valid. The method of identifying the narrow layer will be addressed later below. We define this narrow layer as the second layer Ω_2 and denote its boundary by S_2. We can compute the four GO ingredients everywhere in Ω_2 at every secondary source point on S_1, so that we can construct the Green's function $\boldsymbol{G}(\boldsymbol{r}';\boldsymbol{r})$ and the take-off direction $\boldsymbol{t}^{(0)}(\boldsymbol{r}';\boldsymbol{r})$ for $\boldsymbol{r}' \in \Omega_2$ and $\boldsymbol{r} \in S_1$. Consequently, we can apply the Huygens-Kirchhoff formula (48) on S_1 to compute $\boldsymbol{G}(\boldsymbol{r}';\boldsymbol{r}_0)$ for all points $\boldsymbol{r}' \in \Omega_2$, and the Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ and its gradient $\nabla\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ become available on S_2. Repeating such a process, we can compute the Green's function everywhere by sweeping through the whole domain in a layer-by-layer fashion.

As in [28], the way of dividing the whole domain into layers is based on the first-arrival traveltime solution for the eikonal equation with a point-source condition. We first solve the eikonal equation (12) with the boundary condition (21) for the first-arrival traveltime $\tau(\boldsymbol{r};\boldsymbol{r}_0)$ in the whole domain. The first layer Ω_1 can be defined as the local neighborhood of $\boldsymbol{r}_0$ where τ is smooth since the first-arrival solution coincides with the classic solution wherever it is smooth. Once we have found the first layer Ω_1 and its boundary S_1, we set up a few secondary-source points on S_1, compute the first-arrival traveltimes for each individual source point, and determine the common region where all first-arrival traveltimes are smooth. We regard this common region as the second layer Ω_2 and its boundary as the next secondary-source surface S_2. This process can be repeated so that all such layers cover the whole computational domain. We remark that this domain-partition process needs to be done only once.

From the boundary conditions (22) and (23), we can see that both the amplitude and the polarization dyad are singular at the point source so that the asymptotic Green's function will lose accuracy nearby the point source. Thus, we have to move each secondary source surface S_j slightly closer to the primary source so that the receiver points in the next layer Ω_{j+1} and the secondary source surface S_j are well-separated. Therefore, the Huygens-Kirchhoff formula (48) can be applied to compute $\boldsymbol{G}(\boldsymbol{r}';\boldsymbol{r}_0)$ in the next layer Ω_{j+1} since the four GO ingredients at sources on S_j are accurate in Ω_{j+1}. The Green's function $\boldsymbol{G}$ in those missing portions is actually already computed by either using the asymptotic GO ansatz for the primary source or using the Huygens-Kirchhoff formula in the previous layer. In practice, since we are interested in designing methods with efficiency independent of frequency, we will set a fixed distance $d_f > 0$ to separate the source surface S_j from the receiver region Ω_{j+1} for all j.

3.3 Planar-layer based Huygens sweeping

In the following, considering that the wave field under consideration directs along the positive z direction, we assume that those non-overlapping layers Ω_j are cuboids with infinite lengths in x- and y- directions and all secondary-source surfaces S_j are planar and perpendicular to the z- direction so that the Huygens-Kirchhoff formula is reduced to equation (49).

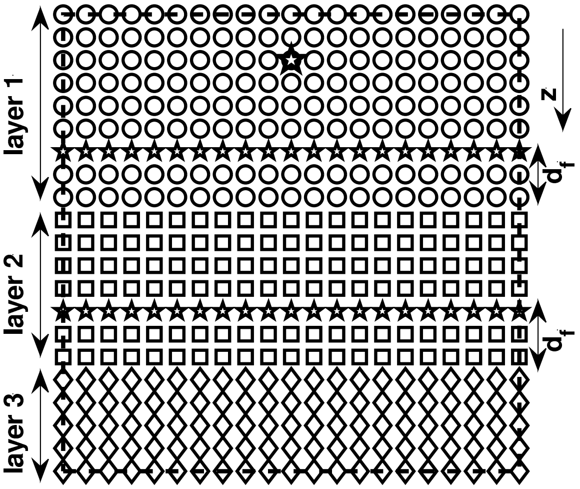


Figure 2: A 2-D sketch of Huygens sweeping method. "large star": primary source; "small star": secondary source; d_f: specify the distance from a secondary source plane in one layer to the next layer; Green's function excited by the large star at "circle", "squared", and "diamond" points is computed by different approaches.

We use Figure 2 to illustrate the sweeping process, where the whole computational domain, i.e., the dashed rectangle, is partitioned into three layers, the large star is the primary source, smaller stars are secondary sources selected as mesh points on the same secondary source plane, and each of the two secondary-source planes is within one layer and is $d_f > 0$ wide away from the next layer. In layer 1, we directly use the asymptotic GO method to compute the primary-source Green's function at those circle points (including the smaller stars) so that the value of the primary-source Green's function at those smaller stars (secondary sources) is available. Next, in layer 2, we compute secondary-source Green's functions excited by those smaller stars in layer 1 at the squared points so that the Huygens-Kirchhoff formula (49) can be applied to compute the primary-source Green's function at those squared points. Once the primary-source Green's function at those smaller stars (secondary sources) in layer 2 is available, a similar process can be used to compute the primary-source Green's function at those diamond points in layer 3. Consequently, the primary-source Green's function in the entire computational domain becomes available.

In general, the planar-based Huygens sweeping method can be summarized as follows.

Algorithm 1.

- **Offline Stage** Precomputing asymptotic ingredients.
  - Partition the computational domain into layers Ω_j and identify secondary source planes S_j.
  - For each secondary source on S_j, compute the table of the four GO ingredients: traveltime, amplitude, take-off direction, and polarization dyad at each secondary source on a coarse mesh in the layer Ω_{j+1}. In practice, we only compute those tables for a coarse set of secondary sources since we can interpolate the coarse tables to generate tables for a dense set of secondary sources.
  - The tables for the coarse set of secondary sources are stored (on a hard drive) and can be used to construct the wave field for all high frequencies.
- **Online Stage** Given a free-space wavenumber k_0, construct the primary-source Green's function layer by layer.
  - At all secondary sources on S_j, the tables of the four ingredients, traveltime, amplitude, take-off direction, and polarization dyad, are loaded from the hard drive to construct the Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ in the layer Ω_{j+1}.
  - For each table, first interpolate the data onto a finer mesh to resolve $\boldsymbol{G}$ and then compute $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ by Huygens-Kirchhoff formula (49) with a quadrature rule.
  - If the sampling of secondary sources on the source plane S_j is not dense enough, we can interpolate the tables from given source locations onto the region bounded by these source locations. This is feasible because asymptotic ingredients are continuous functions of source locations. For instance, if the four source points s_A, s_B, s_C and s_D are vertices of a rectangular region $ABCD$ of the mesh on S_j, and if tables at these four source points are computed and are already interpolated onto the finer mesh in the corresponding layer, we can interpolate the four tables to find the table onto the same finer mesh at any source point inside the rectangular region $ABCD$.

To implement **Algorithm 1**, we need to overcome several obstacles. The first significant obstacle is how to store many tables of the four ingredients generated by our Huygens sweeping method as we are dealing with the nine-element electric Green's function for the 3-D Maxwell's equations that involve the computation of four ingredients at many secondary sources. The second obstacle is how to carry out the matrix-vector multiplications induced by the discretization of integrals in the Huygens-Kirchhoff formula (49), where matrices are dense. We will address the two issues in the subsequent sections.

3.4 Data tables and compression

To reduce data storage, we will follow the approach in [1, 28, 41] to compress each data table into a linear combination of tensor-product based multivariate Chebyshev polynomials so that the information in each table is encoded into a small number of Chebyshev coefficients. In the setting here, there are 14 scalar tables in total to be compressed, including traveltimes, amplitudes, take-off directions, and polarization dyads.

To expedite reconstruction of information from those compressed tables, we will follow the low-rank-matrices based approach in [28] which is equivalent to the Orszag partial summation method [8].

3.5 Discretization of Huygens-Kirchhoff integral

Computationally, to evaluate the (k, j)-entry of the electric Green's function $\boldsymbol{G}(\boldsymbol{r'}; \boldsymbol{r}_0)$ by equation (49) for $k, j = 1, 2, 3$, first we need to truncate the unbounded integration domain S to a finite, bounded domain $\tilde{S}$ according to the specified computational domain for Maxwell's equations. Fortunately, our method is based on geometrical optics so that the numerical domain of dependence is mainly determined by outgoing rays, and the corresponding asymptotic solutions satisfy the Sommerfeld radiation boundary condition at infinity automatically. Thus, the truncation only affects accuracy of the solution nearby the boundary of $\tilde{S}$. After truncation, we can discretize the bounded domain $\tilde{S}$ so that a quadrature rule can be applied to approximate the integral in equation (49).

In the high frequency regime, the dyadic Green's function is highly oscillatory so that we have to specify sufficient mesh points to sample the overall solution. In principle, the optimal number of sampling points is arguably four to six mesh points per wavelength in each direction. While it is difficult for direct methods such as finite-difference or finite-element methods to obtain accurate numerical solutions in the high frequency regime with such an optimal number of mesh points due to pollution or dispersion errors [4, 3], geometrical-optics based Eulerian methods only require four to six mesh points per wavelength to resolve the overall solution in the high frequency regime and the latter has been verified numerically [28] and in the following examples.

Since, given the index of refraction m and the frequency parameter k_0, the smallest wavelength can be estimated to be $\lambda_{\min} = \frac{2\pi}{k_0 m_{\max}}$, where $m_{\max}$ is the largest value of m in the computational domain, we may estimate the number of waves in each direction in the bounded computational domain, and we specify 4 to 6 mesh points per wavelength in each direction accordingly. We remark that because of the scale separation inherent in the geometrical-optics ansatz, the asymptotic ingredients can be computed on very coarse meshes as they are independent of the frequency parameter k_0. Only when we construct the overall wave field do we need to specify enough sampling points accordingly to capture each wave accurately.

Based on the above considerations, we are ready to discretize the integral (49). Assume that the primary source $\boldsymbol{r}_0$ is given, and the truncated rectangular region $\tilde{S}$ is discretized into a set of $M_S = M_x \times M_y$ uniform grid points with the same mesh size h in both x- and y-directions so that the above sampling requirement of four to six points per wavelength is satisfied. We apply the trapezoidal rule to approximate the integral over $\tilde{S}$ and obtain

$$\begin{aligned}
& \boldsymbol{G}_{kj}(\boldsymbol{r}';\boldsymbol{r}_0) \\
\approx\ & h^2 \sum_{m=1}^{M_x}{}'' \sum_{n=1}^{M_y}{}'' \\
& \Big[\big(\boldsymbol{G}_{k3}(\boldsymbol{r}';\boldsymbol{s}_{mn})\ \boldsymbol{t}_1^{(0)}(\boldsymbol{r}';\boldsymbol{s}_{mn}) - \boldsymbol{G}_{k1}(\boldsymbol{r}';\boldsymbol{s}_{mn})\ \boldsymbol{t}_3^{(0)}(\boldsymbol{r}';\boldsymbol{s}_{mn})\big) ik_0 m(\boldsymbol{s}_{mn})\ \boldsymbol{G}_{1j}(\boldsymbol{s}_{mn};\boldsymbol{r}_0) \\
+\ & \big(\boldsymbol{G}_{k3}(\boldsymbol{r}';\boldsymbol{s}_{mn})\ \boldsymbol{t}_2^{(0)}(\boldsymbol{r}';\boldsymbol{s}_{mn}) - \boldsymbol{G}_{k2}(\boldsymbol{r}';\boldsymbol{s}_{mn})\ \boldsymbol{t}_3^{(0)}(\boldsymbol{r}';\boldsymbol{s}_{mn})\big) ik_0 m(\boldsymbol{s}_{mn})\ \boldsymbol{G}_{2j}(\boldsymbol{s}_{mn};\boldsymbol{r}_0) \\
+\ & \boldsymbol{G}_{k1}(\boldsymbol{r}';\boldsymbol{s}_{mn})\ \big(\partial_x \boldsymbol{G}_{3j}(\boldsymbol{s}_{mn};\boldsymbol{r}_0) - \partial_z \boldsymbol{G}_{1j}(\boldsymbol{s}_{mn};\boldsymbol{r}_0)\big) \\
+\ & \boldsymbol{G}_{k2}(\boldsymbol{r}';\boldsymbol{s}_{mn})\ \big(\partial_y \boldsymbol{G}_{3j}(\boldsymbol{s}_{mn};\boldsymbol{r}_0) - \partial_z \boldsymbol{G}_{2j}(\boldsymbol{s}_{mn};\boldsymbol{r}_0)\big)\Big], \qquad (50)
\end{aligned}$$

for $k, j = 1, 2, 3$, where $\sum''$ means that the first and last terms have a factor $1/2$.

In the above, partial derivatives of $\boldsymbol{G}_{kj}(\boldsymbol{s}_{mn};\boldsymbol{r}_0)$ are estimated by numerical differentiation since the Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ is already known near the secondary-source plane $\tilde{S}$; M_x and M_y are the number of grid points in the x- and y- directions, respectively; $\{\boldsymbol{s}_{mn}\}$ are the M_S mesh points (secondary sources) on $\tilde{S}$, and are re-enumerated as $\{\boldsymbol{s}_m\}_{m=1}^{M_S}$. According to the partitioning strategy in Section 3.2, we are interested in evaluating the Green's function $\boldsymbol{G}(\boldsymbol{r}';\boldsymbol{r}_0)$ in a narrow layer Ω which is about d_f far away from the source plane $\tilde{S}$, as shown in Figure 2. Therefore, we evaluate $\boldsymbol{G}_{kj}(\boldsymbol{r}';\boldsymbol{r}_0)$ at those mesh points (receivers) lying in the narrow layer Ω, enumerated as $\{\boldsymbol{r}_n\}_{n=1}^{N_R}$, and we obtain

$$\boldsymbol{g}_{kj} = \boldsymbol{U}_{k1}\ \boldsymbol{f}_{1j} + \boldsymbol{U}_{k2}\ \boldsymbol{f}_{2j} + \boldsymbol{U}_{k3}\ \boldsymbol{f}_{3j} + \boldsymbol{U}_{k4}\ \boldsymbol{f}_{4j}, \qquad (51)$$

for $k, j = 1, 2, 3$, where

$$\begin{aligned}
\boldsymbol{g}_{kj} &= [\boldsymbol{G}_{kj}(\boldsymbol{r}_1;\boldsymbol{r}_0),\ \cdots,\ \boldsymbol{G}_{kj}(\boldsymbol{r}_{N_R};\boldsymbol{r}_0)]^T, \\
\boldsymbol{U}_{k1} &= [\boldsymbol{t}_1^{(0)}(\boldsymbol{r}_n;\boldsymbol{s}_m)\ \boldsymbol{G}_{k3}(\boldsymbol{r}_n;\boldsymbol{s}_m) - \boldsymbol{t}_3^{(0)}(\boldsymbol{r}_n;\boldsymbol{s}_m)\ \boldsymbol{G}_{k1}(\boldsymbol{r}_n;\boldsymbol{s}_m)]_{1\le n\le N_R, 1\le m\le M_S}, \\
\boldsymbol{U}_{k2} &= [\boldsymbol{t}_2^{(0)}(\boldsymbol{r}_n;\boldsymbol{s}_m)\ \boldsymbol{G}_{k3}(\boldsymbol{r}_n;\boldsymbol{s}_m) - \boldsymbol{t}_3^{(0)}(\boldsymbol{r}_n;\boldsymbol{s}_m)\ \boldsymbol{G}_{k2}(\boldsymbol{r}_n;\boldsymbol{s}_m)]_{1\le n\le N_R, 1\le m\le M_S}, \\
\boldsymbol{U}_{k3} &= [\boldsymbol{G}_{k1}(\boldsymbol{r}_n;\boldsymbol{s}_m)]_{1\le n\le N_R, 1\le m\le M_S}, \\
\boldsymbol{U}_{k4} &= [\boldsymbol{G}_{k2}(\boldsymbol{r}_n;\boldsymbol{s}_m)]_{1\le n\le N_R, 1\le m\le M_S}, \\
\boldsymbol{f}_{1j} &= ik_0h^2[m(\boldsymbol{s}_1)\ \boldsymbol{G}_{1j}(\boldsymbol{s}_1;\boldsymbol{r}_0),\ \cdots,\ m(\boldsymbol{s}_{M_S})\ \boldsymbol{G}_{1j}(\boldsymbol{s}_{M_S};\boldsymbol{r}_0)]^T, \\
\boldsymbol{f}_{2j} &= ik_0h^2[m(\boldsymbol{s}_1)\ \boldsymbol{G}_{2j}(\boldsymbol{s}_1;\boldsymbol{r}_0),\ \cdots,\ m(\boldsymbol{s}_{M_S})\ \boldsymbol{G}_{2j}(\boldsymbol{s}_{M_S};\boldsymbol{r}_0)]^T, \\
\boldsymbol{f}_{3j} &= [\partial_x \boldsymbol{G}_{3j}(\boldsymbol{s}_1;\boldsymbol{r}_0) - \partial_z \boldsymbol{G}_{1j}(\boldsymbol{s}_1;\boldsymbol{r}_0),\ \cdots,\ \partial_x \boldsymbol{G}_{3j}(\boldsymbol{s}_{M_S};\boldsymbol{r}_0) - \partial_z \boldsymbol{G}_{1j}(\boldsymbol{s}_{M_S};\boldsymbol{r}_0)]^T, \\
\boldsymbol{f}_{4j} &= [\partial_y \boldsymbol{G}_{3j}(\boldsymbol{s}_1;\boldsymbol{r}_0) - \partial_z \boldsymbol{G}_{2j}(\boldsymbol{s}_1;\boldsymbol{r}_0),\ \cdots,\ \partial_y \boldsymbol{G}_{3j}(\boldsymbol{s}_{M_S};\boldsymbol{r}_0) - \partial_z \boldsymbol{G}_{2j}(\boldsymbol{s}_{M_S};\boldsymbol{r}_0)]^T.
\end{aligned}$$

In the high frequency regime, we have to set up enough secondary sources on $\tilde{S}$ so that the trapezoidal rule yields desired accuracy. Since it is expensive to compute those data tables, we make use of the fact that the geometrical-optics ingredients are independent of the frequency parameter and are continuous functions of the source locations so that we just need to compute those data tables on relatively coarser mesh which can be further interpolated onto relatively finer meshes if needed.

Overall, to compute the 3 by 3 matrix-valued Green's function $\boldsymbol{G}$ at the N_R receivers in the narrow layer Ω, i.e., $\{\boldsymbol{g}_{kj}\}_{1\leq k,j\leq 3}$, we need to compute 12 matrices and 12 vectors $\boldsymbol{U}_{kq}$ and $\boldsymbol{f}_{qj}$ for $1 \leq k,j \leq 3$ and $1 \leq q \leq 4$, and then carry out 36 matrix-vector multiplications arising in equation (51) with matrices of size $N_R \times M_S$ and vectors of size $M_S \times 1$. In practice, N_R and M_S can be extremely large so that direct multiplication of complexity $\mathcal{O}(M_S N_R)$ is expensive and impractical. In the next subsection, we will use a multilevel matrix decomposition based butterfly algorithm [10, 13, 28, 41] to speed up the multiplications.

3.6 A butterfly algorithm

Equation (51) involves a total of 36 matrix-vector multiplications that can be reformulated as

$$g_{kj}(\boldsymbol{r}) = \sum_{q=1}^{4} \sum_{\boldsymbol{s}\in\boldsymbol{X_s}} U_{kq}(\boldsymbol{r};\boldsymbol{s}) f_{qj}(\boldsymbol{s}), \quad \boldsymbol{r} \in \boldsymbol{X_r} \subset \Omega_{\boldsymbol{r}}, \tag{52}$$

for $1 \leq k,j \leq 3$. Here, $\boldsymbol{X}_S$ and $\boldsymbol{X}_R$ are input source points in the domain Ω_S and output receiver points in the domain Ω_R, respectively, where Ω_S and Ω_R are d_f apart from each other; $f_{qj}(\boldsymbol{s})$ is the representative function of $\boldsymbol{f}_{qj}$ in the sense that $f_{qj}(\boldsymbol{s}_m)$ is the m-th element of $\boldsymbol{f}_{qj}$ for $1 \leq m \leq M_S$; $U_{kq}(\boldsymbol{r};\boldsymbol{s})$ is the representative function of $\boldsymbol{U}_{kq}$, taking the following form due to the GO ansatz of $\boldsymbol{G}$,

$$U_{kq}(\boldsymbol{r};\boldsymbol{s}) = A_{kq}(\boldsymbol{r};\boldsymbol{s}) e^{ik_0\tau(\boldsymbol{r};\boldsymbol{s})}, \tag{53}$$

where the amplitude $A_{kq}(\boldsymbol{r};\boldsymbol{s})$ and the traveltime or phase $\tau(\boldsymbol{r};\boldsymbol{s})$ are available for $\boldsymbol{r} \in \Omega_{\boldsymbol{r}}$ and $\boldsymbol{s} \in \Omega_{\boldsymbol{s}}$. Based on such decomposition (53), we can adopt the low-rank-separation-based butterfly algorithm [10, 13, 28] to speed up the 36 matrix-vector products in equation (52).

Although the 36 products involve 12 matrices and 12 vectors, the 12 matrices share the same phase function τ so that we are able to tailor the butterfly algorithm to our specific applications, resulting in rapid computation of the 36 matrix-vector products.

To begin with, we first introduce the multi-dimensional Lagrange basis with respect to Chebyshev nodes. For a given integer $p > 0$, the Chebyshev nodes of order p on the standard one-dimensional box $[-1,1]$ are defined as

$$X = \left\{ x_j = \cos\left(\frac{(j-1)\pi}{p-1}\right) \right\}_{j=1}^{p}.$$

We denote the j-th Lagrange basis function at $x \in [-1,1]$ with nodes X by $L_X(x;x_j)$, taking 1 at x_j and 0 elsewhere in X, for $j = 1,\cdots,p$. On the standard d-dimensional box $[-1,1]^d$, the Chebyshev nodes of order p are d tensor products of X as

$$X^d = \{x_{j_1}\}_{j_1=1}^{p} \times \cdots \times \{x_{j_d}\}_{j_d=1}^{p}.$$

Thus, the $\boldsymbol{j} = (j_1,\cdots,j_d)$-th Lagrange basis function with nodes X^d at $\boldsymbol{x} = (x^1,\cdots,x^d)^T \in [-1,1]^d$ satisfies

$$L_{X^d}(\boldsymbol{x};\boldsymbol{x_j}) = L_X(x^1;x_{j_1})\cdots L_X(x^d;x_{j_d}),$$

where $\boldsymbol{x_j} = (x_{j_1}, \cdots, x_{j_d})^T$.

For a general one-dimensional interval $[a, b]$, the Chebyshev nodes of order p satisfy

$$Y = \left\{ y_j = \frac{a+b}{2} + \frac{b-a}{2} x_j \right\}_{j=1}^{p},$$

so that the j-th Lagrange basis function at $y \in [a, b]$ is denoted by $L_Y(y; y_j)$. Meanwhile, on a general d-dimensional box $[a_1, b_1] \times \cdots \times [a_d, b_d]$, the Chebyshev nodes of order p become the following tensor products

$$\boldsymbol{Y} = Y_1 \times \cdots \times Y_d,$$

where

$$Y_i = \{y_{j_i} = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} x_{j_i}\}_{j_i=1}^{p}.$$

Thus, the $\boldsymbol{j}$-th Lagrange basis function at $\boldsymbol{y} = (y^1, \cdots, y^d)$ is

$$L_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{y_j}) = L_{Y_1}(y^1; y_{j_1}) \cdots L_{Y_d}(y^d; y_{j_d}).$$

In the following, we will denote by $\boldsymbol{C}^B$ the set of p^d d-dimensional Chebyshev nodes in a d-dimensional box B. Following closely the butterfly algorithm in [28], we present the algorithm as follows.

Algorithm 2. (The butterfly algorithm)

1. Construct the cluster trees for both receivers and sources. Assume that the domain of receivers is a cube $\Omega_R = [\mathcal{L}^{\boldsymbol{r}}_{\min}, \mathcal{L}^{\boldsymbol{r}}_{\max}]^3$, and the domain of sources is a square $\Omega_s = [\mathcal{L}^{\boldsymbol{s}}_{\min}, \mathcal{L}^{\boldsymbol{s}}_{\max}]^2$. The domains are discretized such that the number of sampling points per wavelength is fixed, such as 4 to 6 points per wavelength. The cluster trees for the receivers and sources are an octree and a quadtree, respectively.

   At the root level (denoted as level 0), the boxes for both the source and receiver cluster trees are assigned to be the corresponding domain directly. Then the tree construction goes by dyadically subdividing the boxes: for an octree (quadtree), each box is equally divided into 8 (4) boxes. The construction reaches and stops at the leaf level (denoted as level L) where the size of each box is about 2 minimum wavelengths so that approximately $\mathcal{O}(p)$ sampling points are used each dimension with p the order of the Chebyshev nodes. Hence, except for the leaf level, each box B of an octree (quadtree) has 8 (4) children boxes, denoted as B^c, and except for the root level, each box B has a parent box, denoted as B^p. We denote the resulting two trees as T_s (the source tree) and T_R (the receiver tree), respectively. From now on, we will use the superscript $(\cdot)^B$ to denote the dependence on the box B.

   The butterfly algorithm traverses through the two cluster trees in the following way: for $\ell = L, \cdots, 0$, visit level ℓ in T_s and level $L - \ell$ in T_r by considering each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ with $B_{\boldsymbol{r}} \in T_r$ and $B_{\boldsymbol{s}} \in T_s$, $l(B_{\boldsymbol{s}}) = \ell$ and $l(B_{\boldsymbol{r}}) = L - \ell$, where $l(B)$ indicates the level of B in a tree.

Moreover, at the root level of the receiver tree and at the leaf level of the source tree, each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ satisfies

$$w(B_{\boldsymbol{r}})w(B_{\boldsymbol{s}}) = (\mathcal{L}^{\boldsymbol{r}}_{\max} - \mathcal{L}^{\boldsymbol{r}}_{\min})\mathcal{O}(2\lambda_{\min}) = \mathcal{O}\left(\frac{4\pi(\mathcal{L}^{\boldsymbol{r}}_{\max} - \mathcal{L}^{\boldsymbol{r}}_{\min})}{k_0 m_{\max}}\right) = \mathcal{O}\left(\frac{1}{k_0}\right),$$

where $w(B)$ is the size of box B and $m_{\max}$ is the maximum refractive index. As moving downward the receiver tree T_r by one level and simultaneously moving upward the source tree T_s by one level, $w(B_{\boldsymbol{r}})$ is divided by 2 while $w(B_{\boldsymbol{s}})$ is multiplied by 2 so that $w(B_{\boldsymbol{r}})w(B_{\boldsymbol{s}}) = \mathcal{O}(1/k_0)$ is automatically satisfied. For the motivation of such conditions on all box pairs $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$, please see [10, 28] for details. In the following, we mean by the equivalent sources the Chebyshev nodes on a source box $B_{\boldsymbol{s}}$ and by equivalent points the Chebyshev nodes on a receiver box $B_{\boldsymbol{r}}$.

2. The **Upward Pass** starts at the leaf level (level L) of the source tree T_s and ends at the level (denotes as L_s), where the size of the boxes $w(B_s) \geq \mathcal{O}(\frac{1}{\sqrt{k_0}})$. Correspondingly, the level of the receiver tree T_r varies from the root level (level 0) to level $L_r \equiv L - L_s$.

   (1) **Initialization:** For each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ with $B_{\boldsymbol{r}}$ traversing all boxes at the root level of the receiver tree and $B_{\boldsymbol{s}}$ traversing all boxes at the leaf level of the source tree, interpolate 12 equivalent densities $\{\bar{f}_{qj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}\}$ at the equivalent sources $\boldsymbol{C}^{B_{\boldsymbol{s}}} = \{\boldsymbol{s}_n^{B_{\boldsymbol{s}}}\}_{n=1}^{p^2}$ in $B_{\boldsymbol{s}}$ from the given 12 input densities $\{f_{qj}\}$ at all sources in $\boldsymbol{X}_S \cap B_{\boldsymbol{s}}$:

   $$\bar{f}_{qj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{s}_n^{B_{\boldsymbol{s}}}) = \sum_{\boldsymbol{s}\in\boldsymbol{B}_s\cap\boldsymbol{X}_S} e^{-ik_0\tau(\boldsymbol{r}_c^{B_{\boldsymbol{r}}};\boldsymbol{s}_n^{B_{\boldsymbol{s}}})} L^2_{\boldsymbol{C}^{B_{\boldsymbol{s}}}}(\boldsymbol{s};\boldsymbol{s}_n^{B_{\boldsymbol{s}}}) e^{ik_0\tau(\boldsymbol{r}_c^{B_{\boldsymbol{r}}};\boldsymbol{s})} f_{qj}(\boldsymbol{s}), \quad (54)$$

   for $1 \leq q \leq 4$ and $1 \leq j \leq 3$, where $\boldsymbol{r}_c^{B_{\boldsymbol{r}}}$ is the center of the receiver box $B_{\boldsymbol{r}}$.

   (2) For ℓ from $L-1$ to L_s, for each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ with $B_{\boldsymbol{r}}$ traversing all boxes at level $L-\ell$ of the receiver tree and $B_{\boldsymbol{s}}$ traversing all boxes at level ℓ of the source tree, interpolate equivalent densities $\{\bar{f}_{qj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}\}$ at equivalent sources $C^{B_s} = \{\boldsymbol{s}_n^{B_s}\}_{n=1}^{p^2}$ from equivalent densities $\{\bar{f}_{qj}^{B_{\boldsymbol{r}}^p,B_{\boldsymbol{s}}^c}\}$ at equivalent sources $C^{B_{\boldsymbol{s}}^c} = \{\boldsymbol{s}_m^{B_{\boldsymbol{s}}^c}\}_{m=1}^{p^2}$ of all children clusters of $B_{\boldsymbol{s}}$ and the parent cluster of $B_{\boldsymbol{r}}$:

   $$\bar{f}_{qj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{s}_n^{B_s}) = \sum_{B_{\boldsymbol{s}}^c}\sum_{m=1}^{p^2} e^{-ik_0\tau(\boldsymbol{r}_c^{B_{\boldsymbol{r}}};\boldsymbol{s}_n^{B_{\boldsymbol{s}}})} L^2_{\boldsymbol{C}^{B_{\boldsymbol{s}}}}(\boldsymbol{s}_m^{B_{\boldsymbol{s}}^c};\boldsymbol{s}_n^{B_{\boldsymbol{s}}}) e^{ik_0\tau(\boldsymbol{r}_c^{B_{\boldsymbol{r}}};\boldsymbol{s}_m^{B_{\boldsymbol{s}}^c})} \bar{f}_{qj}^{B_{\boldsymbol{r}}^p,B_{\boldsymbol{s}}^c}(\boldsymbol{s}_m^{B_{\boldsymbol{s}}^c}), \quad (55)$$

   for $1 \leq q \leq 4$ and $1 \leq j \leq 3$.

3. **Switch** at the level where the **Upward Pass** has ended (level L_s of the source tree and level L_r of the receiver tree). For each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ with $B_{\boldsymbol{r}}$ traversing all boxes at level L_r of the receiver tree and $B_{\boldsymbol{s}}$ traversing all boxes at level L_s of the source tree, compute 9 equivalent fields $\{\bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}\}$ at equivalent points

$C^{B_{\boldsymbol{r}}} = \{\boldsymbol{r}_m^{B_{\boldsymbol{r}}}\}_{m=1}^{p^3}$ from equivalent densities $\{\bar{f}_{qj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}\}$ at equivalent sources $C^{B_{\boldsymbol{s}}} = \{\boldsymbol{s}_n^{B_{\boldsymbol{s}}}\}_{n=1}^{p^2}$:

$$\bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{r}_m^{B_{\boldsymbol{r}}}) = \sum_{q=1}^{4}\sum_{n=1}^{p^2} U_{kq}(\boldsymbol{r}_m^{B_{\boldsymbol{r}}}, \boldsymbol{s}_n^{B_{\boldsymbol{s}}})\bar{f}_{qj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{s}_n^{B_{\boldsymbol{s}}}), \tag{56}$$

for $1 \le k, j \le 3$.

4. The **Downward Pass** starts at the level L_r of the receiver tree T_r where the **Upward Pass** has ended and ends at level L of the receiver tree. Meanwhile, the level of the source tree varies from level L_s to level 0.

   (1). For ℓ from $L_{\boldsymbol{r}}$ to $L-2$, for each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ with $B_{\boldsymbol{r}}$ traversing all boxes at level $\ell+1$ of the receiver tree and $B_{\boldsymbol{s}}$ traversing all boxes at level $L-\ell-1$ of the source tree, interpolate the equivalent fields $\{\bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}\}$ at equivalent points $\{\boldsymbol{r}_m^{B_{\boldsymbol{r}}}\}_{m=1}^{p^3}$ from equivalent fields $\{\bar{g}_{kj}^{B_{\boldsymbol{r}}^p,B_{\boldsymbol{s}}^c}\}$ at equivalent points $C^{B_{\boldsymbol{r}}^p} = \{\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p}\}_{n=1}^{p^3}$ of the parent level ℓ of the receiver tree and the children level $L-l$ of the source tree:

   $$\bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{r}_m^{B_{\boldsymbol{r}}}) = \sum_{B_{\boldsymbol{s}}^c} e^{ik_0\tau(\boldsymbol{r}_m^{B_{\boldsymbol{r}}};\boldsymbol{s}_c^{B_{\boldsymbol{s}}^c})}\sum_{n=1}^{p^3} L^3_{C^{B_{\boldsymbol{r}}^p}}(\boldsymbol{r}_m^{B_{\boldsymbol{r}}};\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p})e^{-ik_0\tau(\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p};\boldsymbol{s}_c^{B_{\boldsymbol{s}}^c})}\bar{g}_{kj}^{B_{\boldsymbol{r}}^p,B_{\boldsymbol{s}}^c}(\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p}), \tag{57}$$

   for $1 \le k, j \le 3$, where $\boldsymbol{s}_c^{B_{\boldsymbol{s}}^c}$ is the center of the source box $B_{\boldsymbol{s}}^c$.

   (2). For each pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ with $B_{\boldsymbol{r}}$ traversing all boxes at the leaf level of the receiver tree and $B_{\boldsymbol{s}}$ traversing all boxes at the root level of the source tree, interpolate the equivalent fields $\{\bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}\}$ at $\boldsymbol{r} \in X_R \cap B_{\boldsymbol{r}}$ from equivalent fields $\{\bar{g}_{kj}^{B_{\boldsymbol{r}}^p,B_{\boldsymbol{s}}^c}\}$ at equivalent points $C^{B_{\boldsymbol{r}}^p} = \{\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p}\}_{n=1}^{p^3}$ of the parent level $L-1$ of the receiver tree and the children level 1 of the source tree:

   $$\bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{r}) = \sum_{B_{\boldsymbol{s}}^c} e^{ik_0\tau(\boldsymbol{r};\boldsymbol{s}_c^{B_{\boldsymbol{s}}^c})}\sum_{n=1}^{p^3} L^3_{C^{B_{\boldsymbol{r}}^p}}(\boldsymbol{r};\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p})e^{-ik_0\tau(\boldsymbol{r};\boldsymbol{s}_c^{B_{\boldsymbol{s}}^c})}\bar{g}_{kj}^{B_{\boldsymbol{r}}^p,B_{\boldsymbol{s}}^c}(\boldsymbol{r}_n^{B_{\boldsymbol{r}}^p}), \tag{58}$$

   for $1 \le k, j \le 3$.

5. **Termination.** At the leaf level of the receiver tree, for each box $B_{\boldsymbol{r}}$, sum up the equivalent fields over all the boxes of the source tree at the root level, and according to equation (51) compute the representative function g_{kj} of $\boldsymbol{g}_{kj}$ at $\boldsymbol{r} \in X_R \cap B_{\boldsymbol{r}}$:

   $$g_{kj}(\boldsymbol{r}) = \sum_{B_{\boldsymbol{s}}} \bar{g}_{kj}^{B_{\boldsymbol{r}},B_{\boldsymbol{s}}}(\boldsymbol{r}),$$

   for $1 \le k, j \le 3$.

Assume that the tree level L is even and that there are $\mathcal{O}(n) = \mathcal{O}(2^L)$ points in each direction. According to the complexity analysis in [28], we can obtain that the total complexity of **Algorithm 2** is

$$\mathcal{O}\left(12p^4n^{5/2} + 36p^5n^{5/2} + 9p^4n^3 + n^3\log n\right), \tag{59}$$

which is much less than the complexity of performing the butterfly algorithm 36 times, i.e.,

$$\mathcal{O}\left(36p^4n^{5/2}+36p^5n^{5/2}+36p^4n^3+36n^3\log n\right).$$

3.6.1 MATLAB-based parallelized butterfly algorithm

In **Algorithm 2**, Ω_R and Ω_S are assumed to be cubed and squared, respectively. Nevertheless, even if this assumption is not satisfied, we can construct a required output receiver-point set X_R and a required input source-point set X_S so that **Algorithm 2** can be parallelized.

At first, in the construction of the two cluster trees, if the domain Ω_R (Ω_S) is not a cube (square), we will divide it into cuboids (rectangles) of the same dimensions that are approximately cubic (squared), and we further subdivide those cuboids (rectangles) until the leaf level is reached. Secondly, the output receiver points X_R (the input source points X_S) should contain all vertices of boxes at leaf level of the receiver (source) tree and are evenly spaced in the receiver (source) domain so that at the leaf level of the receiver (source) tree, the receiver (source) points in each box $B_{\boldsymbol{r}}$ ($B_{\boldsymbol{s}}$) have the same layout and are of the same number. Therefore, as ℓ varies from L to 0, at the level ℓ of the source tree and at the level $L-\ell$ of the receiver tree, matrices and vectors encountered in each of the five summations (54)-(58) have dimensions that do not depend on the box pair $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$, indicating that we can execute the summation for all box pairs $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ in parallel; in our MATLAB implementation, we use the "built-in" parfor (parallelized for) loop in the level of iterations through box pairs $\{B_{\boldsymbol{r}}, B_{\boldsymbol{s}}\}$ so that if the number of specified MATLAB workers is M, then the involved summations (54)-(58) for every M box-pairs are executed simultaneously. Consequently, in comparison with the sequential butterfly algorithm, the parallelized butterfly algorithm reduces the total complexity from the estimate (59) to

$$\mathcal{O}\left(\frac{12p^4n^{5/2}+36p^5n^{5/2}+9p^4n^3+n^3\log n}{M}\right)=\mathcal{O}\left(\frac{n^3\log n}{M}\right).$$

However, the input source-point set $\{\boldsymbol{s}_m\}_{m=1}^{M_S}$ (the output receiver-point set $\{\boldsymbol{r}_n\}_{n=1}^{N_R}$) may not satisfy the above requirements. To resolve this issue, for $1\le j\le 3$ and $1\le q\le 4$ each input data $\boldsymbol{f}_{qj}$ defined on the input source-point set $\{\boldsymbol{s}_m\}_{m=1}^{M_S}$ can be used to obtain the required input data by interpolation, resulting in 12 representative functions f_{qj} for $1\le j\le 3$ and $1\le q\le 4$ on the required input source-point set X_S. Next, by inputting the required data to the parallelized butterfly algorithm, we obtain the required output data, resulting in 9 representative functions g_{kj} for $1\le k,j\le 3$ and $1\le j\le 3$ on the required output receiver-point set X_R. Finally, from the required output data on X_R, we can interpolate the output data back on the output receiver-point set $\{\boldsymbol{r}_n\}_{n=1}^{M_R}$, resulting in the 9 components $\boldsymbol{g}_{kj}$ in equation (51).

3.7 Complexity analysis

The overall algorithm consists of two stages. The first stage, the offline stage, is preprocessing in which the four asymptotic ingredients: traveltime, amplitude, take-off direction, and polarization dyad, are computed and further encoded into a set of data tables of Chebyshev coefficients. The second stage, the online stage, is postprocessing in which we construct the global Green's function for a given primary source $\boldsymbol{r}_0$ and a given wavenumber k_0. Since the two stages are independent of each other and they can be done on different meshes, we will analyze the two stages separately.

In the following analysis, suppose that the computational domain is partitioned into $P+1$ planar layers and we set up P secondary-source planes. Among the $P+1$ layers, we only need to construct the primary-source Green's function in the P layers away from the primary source $\boldsymbol{r}_0$.

3.7.1 Offline preprocessing: computing asymptotic ingredients

At first, since the asymptotic ingredients are independent of wavenumber k_0, they can be computed on a very coarse mesh. Secondly, these ingredients are not only continuous functions of observation points away from the source but also continuous functions of the source itself. Therefore, on each secondary source plane, the asymptotic ingredients can be computed at coarsely sampled secondary sources as well. Interpolation can be used later to generate asymptotic ingredients for densely sampled secondary sources if necessary.

Suppose that the computational domain is uniformly and coarsely discretized by m^3 grid points, amounting to m points in each direction. Since the domain is divided into $P+1$ planar layers, each layer has roughly $\mathcal{O}(\frac{m^3}{P+1})$ points. On each of the P secondary source planes, we set up m^2 secondary sources and compute for each secondary source the asymptotic ingredients in a certain layer so that the computational domain is restricted to that layer as well. According to [28], by using the high-order Lax-Friedrichs sweeping method to solve equations (12), (18), (29) and (30) for the asymptotic ingredients at each secondary source, the computational complexity is $\mathcal{O}(\frac{m^3}{P+1}\log m)$. Next, those asymptotic ingredients are compressed into data tables of Chebyshev coefficients, and the corresponding computational cost is $\mathcal{O}(\frac{m^3}{P+1}\log m)$. Consequently, the overall complexity for generating data tables at the whole m^2P secondary sources is

$$\mathcal{O}\left(\frac{2m^3}{P+1}\log m\cdot Pm^2\right)=\mathcal{O}\left(m^5\log m\right).$$

Although the complexity seems to be high, we can store those compressed data tables in a hard drive and can re-use them for different frequencies and for different primary sources. This feature makes our method appealing to many applications.

On the other hand, to construct the global Green's function, the whole computational domain is uniformly discretized by n point in each direction. Thus, in each layer, we need to recover, from the compressed tables of coefficients, 14 asymptotic ingredients for those specified secondary sources on roughly $\mathcal{O}(\frac{n^3}{P+1})$ points, requiring complexity of $\mathcal{O}(\frac{n^3}{P+1})$ for each ingredient and for each secondary source; see [28] for details.

3.7.2 Online postprocessing: constructing global Green's functions

Given a source point $\boldsymbol{r}_0$ and a free-space wavenumber k_0, a fine mesh is required to capture the highly-oscillatory Green's function in the computational domain. From the given refractive index $m(\boldsymbol{r})$ and k_0, we can estimate the smallest wavelength in the computational domain so that the total number of waves along each direction can be estimated. In principle, along each direction, sampling roughly 4 to 6 grid points per wave is considered to be enough to capture the oscillations; certainly, it does not hurt if more points are taken. Therefore, the number of discretization points in the computational domain can be chosen to satisfy the above consideration, and it is assumed to be $N = n^3$, where n is the number of points in each direction.

Once the four asymptotic ingredients are available on the specified mesh inside each layer, we can construct the Green's function by the butterfly-algorithm-based Huygens-Kirchhoff summation (51). Given accuracy $\varepsilon > 0$, according to [10], we may choose $p = p_\varepsilon \leq \mathcal{O}(\log^2(\frac{1}{\varepsilon}))$ for the order of Chebyshev nodes in each direction in the butterfly algorithm so that the algorithm for computing the summation achieves the accuracy ε in $\mathcal{O}(\frac{N}{P+1}\log N)$, where the prefactor depends only on ε and does not depend on k_0. Therefore, the overall complexity for constructing the Green's function by the butterfly algorithm in the P layers is $\mathcal{O}(\frac{P}{P+1}N\log N) = \mathcal{O}(N\log N)$ for a given primary source point $\boldsymbol{r}_0$ and a given frequency. If computed in parallel, the complexity is further reduced to $\mathcal{O}(N/M\log N)$, where M denotes the number of workers in the parallelization.

4 Numerical examples

In the following examples, we first compute the four geometric-optics (GO) ingredients: traveltime, amplitude, take-off direction, and polarization dyad, at the primary source on a coarse mesh, and this mesh is referred to as the GO coarse mesh. When the corresponding mesh is used for computing the four GO ingredients at secondary sources, the GO coarse mesh will be restricted to a neighborhood of each secondary source. At those secondary sources selected in a secondary source domain, we obtain data tables of 14 functions: traveltime, amplitude, three elements in take-off direction, nine elements in polarization dyad. As all 14 functions are smooth in a narrow layer away from the corresponding secondary source planes, we compress the 14 data tables into 14 tables of Chebyshev coefficients. Therefore, for all the numerical experiments, the GO ingredients are given as the compressed data, and they are recovered onto the finer mesh by the Chebyshev partial summation method when needed.

Given the refractive index $m(\boldsymbol{r})$ and the free space wavenumber $k_0 = \omega/c_0$, the smallest wavelength $\lambda_{\min}$ can be estimated; accordingly we may estimate the number of waves in each direction in the bounded computational domain. In principle, 4 to 6 points per wavelength in each direction are sufficient for the Huygens sweeping method to construct the Green's function $\boldsymbol{G}$ accurately. In our numerical settings, the computational domain is discretized into mesh points with roughly 4 points per wavelength in each direction.

Unless otherwise stated, all computations were executed in a ten-core 2.5 GHz Intel Xeon processor with 256 Gbytes of RAM, associated with a bi-processor Intel Xeon E5-2670V2 node, at the High Performance Computing Center (HPCC) of MSU. All GO ingredients were computed with C codes using a single core while the Green's function was constructed with MATLAB codes and the butterfly algorithm was carried out in parallel in 10 cores via the parallel computing toolbox of MATLAB.

When the refractive index $m(\boldsymbol{r})$ is not constant, an exact solution of the Green's function $\boldsymbol{G}$ is not available. To validate the accuracy of the Huygens' sweeping method, we obtain reference solutions by applying the finite-difference time-domain (FDTD) method [50] directly on the time-domain Maxwell's equations. However, due to limited computing resources, we are only able to compute the FDTD-based solutions at low frequencies, so that the comparison will be only carried out for low frequencies. In the FDTD method, to absorb the outgoing rays, two uniaxial perfectly matched layers (UPMLs) of a fixed width are added to surround the original computational domain in each direction; in the following, we take the UPMLs with width 0.2 so that the whole computational domain is then discretized by using roughly 16 points per wavelength to reduce dispersion error.

Since in the Huygens sweeping method, Green's functions at different sources are approximated by the asymptotic GO form (8) with truncation error $\mathcal{O}(1/k_0)$, we expect that our Huygens sweeping solutions will agree with the FDTD solutions or exact solutions, if available, within the expected accuracy $\mathcal{O}(1/k_0)$ as well.

Example 1: Constant refractive index model. In this example, the Green's function is constructed with the following setup.

- The refractive index function is $m(\boldsymbol{r}) \equiv 1$.
- The computational domain is $\Omega = [0,2] \times [0,2] \times [0,2]$, and the GO coarse mesh is $51 \times 51 \times 51$.
- The primary source point is $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$.
- One secondary source plane is placed at $z = 1.2$ and we coarsely sample 51×51 equally spaced secondary sources on the truncated source domain $\Omega_S = [0,2] \times [0,2] \times \{z = 1.2\}$. The corresponding receiver domain is $\Omega_R = [0,2] \times [0,2] \times [1.4, 2.0]$, which is $d_f = 0.2$ wide away from the source domain Ω_S.
- At each secondary source, numbers of retained Chebyshev coefficients in each of the 14 tables in x-, y-, z- directions are 31, 31, and 13, respectively.

In this example, the running time for computing the 14 tables of Chebyshev coefficients is around 13 hours. Those data tables can be reused for different free-space wavenumbers. For different wavenumbers and different values of p, where p is the order of Chebyshev nodes in each direction used in **Algorithm 2**, we record running times for computing all 9 elements of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ in the computational domain Ω, as shown in Table 1.

Mesh in Ω	$49\times 49\times 49$	$97\times 97\times 97$	$193\times 193\times 193$	$257\times 257\times 257$
$k_0/(2\pi)$	6	12	24	32
$\lambda_{\min}$	1/6	1/12	1/24	1/32
NW	$12\times 12\times 12$	$24\times 24\times 24$	$48\times 48\times 48$	$64\times 64\times 64$
NPW	4	4	4	4
$T_{all}(p=7)$	302.59	928.57	4281.9	4375.2
$T_{all}(p=9)$	336.72	1043.1	4532.2	5141.6
$T_{all}(p=11)$	388.92	1070.7	5613.0	6708.8

Table 1: Constant refractive index model with $\boldsymbol{r}_0=[1.0,1.0,0.2]^T$. T_{all} (unit: s): Total CPU time for computing all 9 elements of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ in Ω; NPW: the number of points per wavelength; $p=7,9,11$ Chebyshev nodes are used in each dimension; NW: approximate number of waves in Ω.

For the constant refractive index model, the exact GO form of Green's functions excited by any source $\boldsymbol{r}_*\in\Omega$ is available in Ω and given by equation (7) with $\boldsymbol{r}_0$ replaced by $\boldsymbol{r}_*$ so that exact forms of the four GO ingredients at the source $\boldsymbol{r}_*$ are available in Ω as well. Therefore, in equation (51), all 12 matrices $\boldsymbol{U}_{kq}$ for $1\le k\le 3$ and $1\le q\le 4$ can be computed directly via the exact GO form of the Green's function and the exact form of take-off direction at the selected secondary sources so that we can use direct matrix-vector products to obtain the 9 components $\boldsymbol{g}_{kj}$ in the receiver domain Ω_R for $1\le k,j\le 3$, which are regarded as reference solutions. Thus, by comparing our butterfly-algorithm based results with those reference solutions, the butterfly algorithm can be justified.

In Table 2, the fourth row shows the running time T_D for computing reference solutions for all 9 elements of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ in Ω_R, denoted by $\boldsymbol{G}_{ref}$, by direct matrix-vector multiplications at different wavenumbers and by using the exact form of matrices $\boldsymbol{U}_{kq}$ in equation (51). In comparison, at the same set of wavenumbers and at different values of p, we record running times of the butterfly-algorithm based Huygens-Kirchhoff summation for computing numerical solutions for all 9 elements of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ in Ω_R, denoted by $\boldsymbol{G}_{num}$, and compute the following L_∞ error

$$E_\infty=||\boldsymbol{G}_{ref}-\boldsymbol{G}_{num}||_\infty,$$

to check the accuracy of the butterfly-algorithm-based Huygens-Kirchhoff summation. We can see from Table 2 that the two different approaches are consistent with each

Mesh X_R in Ω_R	$49 \times 49 \times 17$	$97 \times 97 \times 33$	$193 \times 193 \times 65$	$265 \times 265 \times 81$
$k_0/(2\pi)$	6	12	24	32
NPW	4	4	4	4
T_D	48	3619	6.75E4	2.39E5
$T_M(p = 7)$	276.52	907.87	4169.0	4341.3
E_∞	6.43E-2	7.92E-2	1.16E-1	1.82E-1
T_D/T_M	0.17	3.98	16.19	55.05
$T_M(p = 9)$	313.53	1030.1	4470.0	5109.3
E_∞	3.13E-2	2.93E-2	5.58E-2	9.97E-2
T_D/T_M	0.15	3.51	15.10	46.77
$T_M(p = 11)$	366.03	1057.7	5588.8	6673.3
E_∞	6.80E-3	1.01E-2	2.38E-2	6.6E-2
T_D/T_M	0.13	3.42	12.07	35.81

Table 2: Constant refractive index model. T_D (unit: s): Direct matrix-vector products based on exact forms of $\boldsymbol{U}_{kq}$ in (51). T_M (unit: s): CPU time for the construction of all 9 elements of the Green's function in the receiver domain Ω_R by the butterfly-algorithm based Huygens-Kirchhoff summation; NPW: the number of points per wavelength; $p = 7, 9, 11$ Chebyshev nodes are used in each dimension; E_∞: the L_∞ error between solutions by two different approaches.

other and the butterfly algorithm speeds up the matrix-vector products dramatically at high frequencies.

We remark that since T_D does not include the running time for numerically computing $\boldsymbol{U}_{kq}$, the true running time of computing the primary-source Green's function $\boldsymbol{G}$ in Ω_R via direct matrix-vector multiplications is much longer than the T_D shown here.

As $m(\boldsymbol{r})$ is constant, the exact solution of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ given by formula (6) can be used to validate our numerical solutions. Figures 3, 4, and 5 show contour plots of numerical solutions of real parts of xx-, zx-, and zz- components of the Green's function, $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$, and $\boldsymbol{G}_{33}$, at $y = 1.0$ computed by two different methods: butterfly-algorithm based Huygens sweeping method with Chebyshev order $p = 11$ and the exact solution of form (6) for the free-space wavenumber $k_0 = 24\pi$. We first use the direct GO-ansatz method to construct the three components $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$ and $\boldsymbol{G}_{33}$ in $\Omega/\Omega_R = [0, 2] \times [0, 2] \times [0, 1.4]$, and then use the Huygens sweeping method to

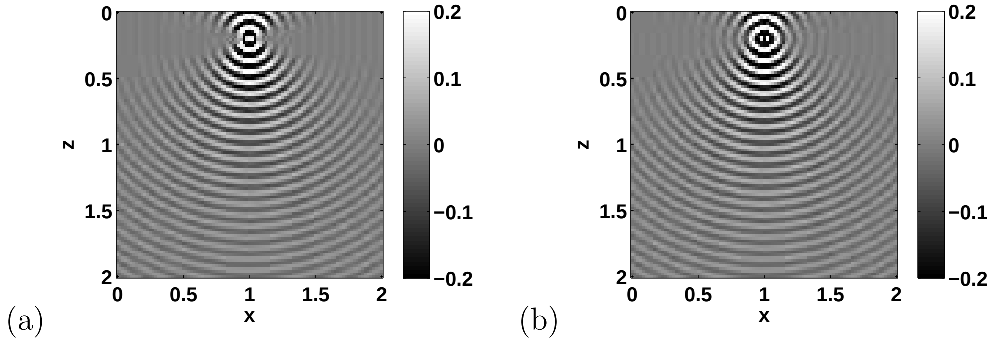


Figure 3: Constant refractive index model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 24\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): butterfly-algorithm-based Huygens sweeping method with $p = 11$; (b): exact solution of form (6). Mesh $97 \times 97 \times 97$.

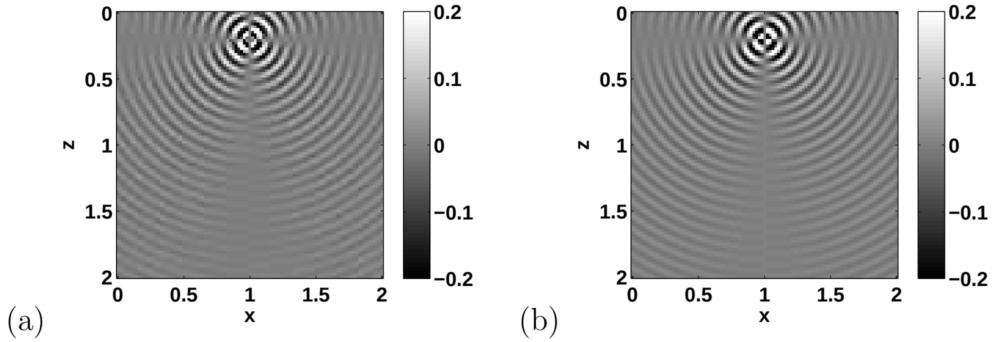


Figure 4: Constant refractive index model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 24\pi$. Real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): butterfly-algorithm-based Huygens sweeping method with $p = 11$; (b): exact solution of form (6). Mesh $97 \times 97 \times 97$.

construct the three components in the receiver domain $\Omega_R = [0, 2] \times [0, 2] \times [1.4, 2.0]$ by the Huygens-Kirchhoff summation (50), as shown in Figures 3(a), 4(a) and 5(a). We can see from Figures 3, 4, and 5 that our Huygens-principle-based numerical solutions are consistent with true solutions except around sources.

Figure 6 shows more detailed comparisons of the three components $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$ and $\boldsymbol{G}_{33}$ at different lines by the two methods when $k_0 = 24\pi$. We can see that our numerical solutions agree with true solutions very well except near the source $\boldsymbol{r}_0$. In particular, in Figure 6(e), the numerical solution for the zz-component $\boldsymbol{G}_{33}$ by our method seems to agree poorly with the true solution. However, the reason is that when $x = 1$ and $y = 1$, the leading term of $\boldsymbol{G}_{33}$ in formula (6) vanishes so that the second term in formula (6) becomes dominant. Nevertheless, our numerical solutions still meet the expected accuracy $\mathcal{O}(1/k_0)$.

Example 2. Gaussian model. This example is set up as the following.

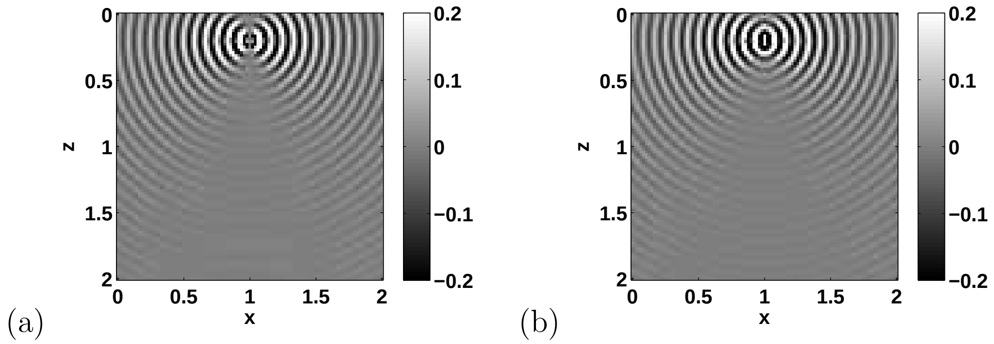


Figure 5: Constant refractive index model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 24\pi$. Real part of the zz-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): Huygens sweeping method with $p = 11$; (b): exact solution of form (6). Mesh $97 \times 97 \times 97$.

- The refractive index function is
$$m(\boldsymbol{r}) = m(r) = \frac{3}{3 - 1.75e^{-r^2/0.64}},$$
where $r = \sqrt{(x-1)^2 + (y-1)^2 + (z-1)^2}$.
- The computational domain is $\Omega = [0, 2] \times [0, 2] \times [0, 2]$, and the GO coarse mesh is $51 \times 51 \times 51$.
- The primary source point is $\boldsymbol{r}_0 = [1.0, 1.0, 1.0]^T$.
- One secondary source plane is placed at $z = 1.2$ and we coarsely sample 51×51 equally spaced secondary sources on the truncated source domain $\Omega_S = [0, 2] \times [0, 2] \times \{z = 1.2\}$. The corresponding receiver domain is $\Omega_R = [0, 2] \times [0, 2] \times [1.4, 2.0]$, which is $d_f = 0.2$ wide away from the source domain Ω_S.
- At each secondary source, the numbers of retained Chebyshev coefficients in each of the 14 tables in x-, y-, z- directions are 31, 31, and 13, respectively.

The running time for computing the 14 tables of Chebyshev coefficients is around 36 hours. For different wavenumbers k_0 and different Chebyshev orders p, we record running times for computing all 9 elements of the Green's function in the computational domain Ω, as shown in Table 3.

Table 4 shows running times of the butterfly-algorithm based Huygens-Kirchhoff summation for constructing all 9 components of the Green's function $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ in Ω_R.

For this model, the geometric-optics approximation of the Green function has the following analytic form

$$\tilde{\boldsymbol{G}}(\boldsymbol{r}; \boldsymbol{r}_0) = \frac{1}{4\pi r}\sqrt{\frac{m(0)}{m(r)}} e^{ik_0 \int_0^r m(r')dr'} \left(\boldsymbol{I} - \tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^T\right), \tag{60}$$

where
$$\tilde{\boldsymbol{r}} = \frac{\boldsymbol{r} - \boldsymbol{r}_0}{|\boldsymbol{r} - \boldsymbol{r}_0|}.$$

Mesh in Ω	$49 \times 49 \times 49$	$97 \times 97 \times 97$	$193 \times 193 \times 193$	$257 \times 257 \times 257$
$k_0/(2\pi)$	5/2	5	10	40/3
$\lambda_{\min}$	1/6	1/12	1/24	1/32
NW	$12 \times 12 \times 12$	$24 \times 24 \times 24$	$48 \times 48 \times 48$	$64 \times 64 \times 64$
NPW	4	4	4	4
$T_{all}(p=7)$	450.37	906.66	3766.9	4403.4
$T_{all}(p=9)$	501.68	1001.6	4812.5	5219.5
$T_{all}(p=11)$	577.69	1180.0	5797.0	6538.0

Table 3: Gaussian model. T_{all} (unit: s): Total CPU time for computing all 9 elements of the Green's function in Ω; NPW: the number of points per wavelength; $p = 7, 9, 11$ Chebyshve nodes are used in each dimension; NW: approximate number of waves in Ω.

The exact solution of the Green's function $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ is not available so that we use the FDTD method to obtain a reference solution.

Figures 7, 8, and 9 show contour plots of numerical solutions of real parts of xx-, zx-, and zz- components of the Green's function, $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$, and $\boldsymbol{G}_{33}$, at $y = 1.0$ computed by two different methods: butterfly-algorithm based Huygens sweeping method with order $p = 11$, and the FDTD method, at the free-space wavenumber $k_0 = 10\pi$. As shown in Figures 7(a), 8(a), and 9(a), we first use the direct GO ansatz to construct the three components $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$, and $\boldsymbol{G}_{33}$ in $\Omega/\Omega_R = [0, 2] \times [0, 2] \times [0, 1.4]$, and then use the Huygens sweeping method to construct the three components in the receiver domain $\Omega_R = [0, 2] \times [0, 2] \times [1.4, 2.0]$. According to the analytic GO form (60), the traveltime for this model has the following exact solution

$$\tau(\boldsymbol{r};\boldsymbol{r}_0) = \int_0^{|\boldsymbol{r}-\boldsymbol{r}_0|} m(r')dr',$$

which is smooth except at the source point $\boldsymbol{r}_0$ so that we can see from Figures 7, 8, and 9 that the direct GO ansatz solution in $\Omega/\Omega_R = [0, 2] \times [0, 2] \times [0, 1.4]$ and the Huygens sweeping solution in $\Omega_R = [0, 2] \times [0, 2] \times [1.4, 2.0]$ together form numerical solutions which are in good agreement with FDTD solutions except around sources.

Figure 10 shows more detailed comparisons of the three components at different lines by two different methods when $k_0 = 10\pi$. We can see that our numerical solutions agree well with the FDTD solution except around the primary source point $[1, 1, 0.2]^T$. In particular, in Figure 10(e), the direct GO-ansatz solution of the zz-component $\boldsymbol{G}_{33}$ in $z \leq 1.4$ agrees with the FDTD solution poorly, since the leading term of $\boldsymbol{G}_{33}$ for $x = 1$ and $y = 1$ vanishes so that the second term becomes dominant and is comparable to the truncation error $\mathcal{O}(1/k_0)$; in contrast, in the region $z > 1.4$ where the Huygens

Mesh X_R in Ω_R	$49 \times 49 \times 17$	$97 \times 97 \times 33$	$193 \times 193 \times 65$	$265 \times 265 \times 81$
$\lambda_{\min}$	1/6	1/12	1/24	1/32
NPW	4	4	4	4
$T_M(p=7)$	321.70	893.49	3742.2	4369.1
$T_M(p=9)$	462.75	988.4	4787.1	5179.2
$T_M(p=11)$	538.76	1163.7	5771.3	6503.3

Table 4: Gaussian model. T_M (unit: s): CPU time for the construction of all 9 elements in the Green's function in Ω_R by the butterfly-algorithm based Huygens-Kirchhoff summation; NPW: the number of points per wavelength; $p = 7, 9, 11$ Chebyshev nodes are used in each dimension.

sweeping method applies, the numerical solution matches with the FDTD solution very well.

Next, we compute the Green's function $\boldsymbol{G}$ at a different primary source $[1.0, 1.0, 0.2]^T$. For $k_0 = 10\pi$, contour plots of numerical solutions for the three components, $\boldsymbol{G}_{xx}$, $\boldsymbol{G}_{zx}$, and $\boldsymbol{G}_{zz}$ at $y = 1$, are shown in Figures 11, 12, and 13, respectively. We can see that caustics appear in $\Omega_R = [0, 2] \times [0, 2] \times [1.4, 2.0]$ and the Huygens sweeping solutions match with FDTD solutions very well.

Figure 14 shows more detailed comparison of the three components at different lines by two different methods when $k_0 = 10\pi$. We can see that our numerical solutions agree well with the FDTD solution within the expected accuracy $\mathcal{O}(1/k_0)$. In particular, in Figure 14(e), we can see that the two numerical solutions differ by expected truncation errors $\mathcal{O}(1/k_0)$ in the region Ω/Ω_R, where the direct GO-ansatz solution is used; nevertheless, in the region $z > 1.4$ where our Huygens sweeping method applies, the numerical solution matches with the FDTD solution very well. Figure 15 show numerical solutions at higher wavenumbers $k_0 = 20\pi$ and $k_0 = 80/3\pi$.

Example 3: Waveguide model. In this example, we have the following setup.

- The refractive index function is

$$m(\boldsymbol{r}) = m(x, y, z) = \frac{1}{1 - 0.5e^{-6((x-1)^2+(y-1)^2)}}.$$

- The computational domain is $\Omega = [0, 2] \times [0, 2] \times [0, 1.6]$, and the GO mesh is $51 \times 51 \times 41$.
- The primary source point is $[1.0, 1.0, 0.2]^T$.
- Two secondary source planes are placed at $z = 0.6$ and $z = 1.0$, respectively, according to the domain-partition rule in Section 3.2. We coarsely sample 51×51 equally spaced secondary sources on each of the two truncated source domain

$\Omega_S^1 = [0,2] \times [0,2] \times \{z = 0.6\}$ and $\Omega_S^2 = [0,2] \times [0,2] \times \{z = 1.0\}$. The corresponding two receiver domains are $\Omega_R^1 = [0,2] \times [0,2] \times [0.8,1.2]$ and $\Omega_R^2 = [0,2] \times [0,2] \times [1.2,1.6]$, which are $d_f = 0.2$ wide away from Ω_S^1 and Ω_S^2, respectively.

- At each secondary source, numbers of retained Chebyshev coefficients in each of the 14 tables in x-, y-, z- directions are 31, 31 and 13, respectively.

As the refractive index m does not depend on z, the 14 tables of Chebyshev coefficients are only computed once for the sampled secondary sources on Ω_S^1 and can be reused for secondary sources on Ω_S^2. The running time of computing the 14 tables is around 32 hours.

For different wavenumbers k_0 and different Chebyshev orders p, we record running times for computing all 9 elements of the Green's function in the computational domain Ω, as shown in Table 5. When $\lambda_{\min} = \frac{1}{32}$ with 4 points per wavelength with $p = 11$, it took around 3 hours for the paralellized fast Huygens sweeping method to construct the 9-component Green's function of $257 \times 257 \times 205 \times 9 = 121.8$ millions unknowns in a 3-D electromagnetic waveguide model, where we have used the Matlab parallel-computing toolbox to manage 10 cores in parallel; in this case, the total λ-volume is $1,887,436.8\lambda_{\min}^3$, a large-scale high-frequency problem.

Mesh in Ω	$49 \times 49 \times 39$	$97 \times 97 \times 77$	$193 \times 193 \times 154$	$257 \times 257 \times 205$
$k_0/(2\pi) = \omega/(2\pi c_0)$	3	6	12	16
$\lambda_{\min}$	1/6	1/12	1/24	1/32
NW	$12 \times 12 \times 10$	$24 \times 24 \times 20$	$48 \times 48 \times 40$	$64 \times 64 \times 52$
NPW	4	4	4	4
$T_{all}(p = 7)$	495.50	1506.3	6401.8	7068.2
$T_{all}(p = 9)$	561.95	1623.3	7596.1	8208.8
$T_{all}(p = 11)$	650.00	1840.4	9224.2	10068.0

Table 5: Waveguide model. T_{all} (unit: s): Total CPU time for computing all 9 elements in the Green's function in Ω; NPW: the number of points per wavelength; $p = 7, 9, 11$ Chebyshev nodes are used in each dimension; NW: approximate number of waves in Ω.

Tables 6 and 7 show the running times for constructing all 9 components of the Green's function $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ in the two receiver domains, Ω_R^1 and Ω_R^2, based on the butterfly-algorithm based summations, respectively. In this example, since caustics occur much closer to the primary source point and since we have to construct the Green's function in two receiver domains Ω_R^1 and Ω_R^2, we execute **Algorithm 2** twice so that the total running times in Table 5 almost double those running times in previous examples.

Mesh X_R^1 in Ω_R^1	$49 \times 49 \times 11$	$97 \times 97 \times 21$	$193 \times 193 \times 41$	$265 \times 265 \times 57$
$\lambda_{\min}$	1/6	1/12	1/24	1/32
NPW	4	4	4	4
$T_M^1(p = 7)$	237.39	765.70	3373.8	3441.1
$T_M^1(p = 9)$	270.19	805.48	3718.0	4101.9
$T_M^1(p = 11)$	314.57	936.63	4938.9	5199.2

Table 6: Waveguide model. T_M^1 (unit: s): CPU time for the reconstruction of all 9 elements in the Green's function in Ω_R^1 by the butterfly-algorithm-based Huygens-Kirchhoff summation; NPW: the number of points per wavelength; $p = 7, 9, 11$ Chebyshev nodes are used in each dimension.

Mesh X_R^2 in Ω_R^2	$49 \times 49 \times 11$	$97 \times 97 \times 21$	$193 \times 193 \times 41$	$265 \times 265 \times 57$
$\lambda_{\min}$	1/6	1/12	1/24	1/32
NPW	4	4	4	4
$T_M^2(p = 7)$	235.65	728.89	3003.8	3596.9
$T_M^2(p = 9)$	269.97	803.42	3493.7	4038.1
$T_M^2(p = 11)$	314.09	892.24	4260.8	4831.8

Table 7: Waveguide model. T_M^2 (unit: s): CPU time for the reconstruction of all 9 elements in the Green's function in Ω_R^2 by the butterfly-algorithm-based Huygens-Kirchhoff summation; NPW: the number of points per wavelength; $p = 7, 9, 11$ Chebyshev nodes are used in each dimension.

When $k_0 = 12\pi$, the total λ-volume is $11,059\lambda^3 \times 9 = 99,531\lambda^3$ with $\lambda = \frac{1}{12}$ being the smallest wavelength. We use the FDTD method to compute a reference solution to validate our numerical solutions. Figures 16, 17, and 18 show contour plots of numerical solutions of real parts of xx-, zx-, and zz- components of the Green's function, $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$, and $\boldsymbol{G}_{33}$ at $y = 1.0$, computed by two different methods: the butterfly-algorithm based Huygens sweeping method with order $p = 11$, and the FDTD method. In Figures 16(a), 17(a) and 18(a), we use the Huygens sweeping method to construct the three components $\boldsymbol{G}_{11}$, $\boldsymbol{G}_{31}$ and $\boldsymbol{G}_{33}$ in the two receiver domains $\Omega_R^1 = [0, 2] \times [0, 2] \times [0.8, 1.2]$ and $\Omega_R^2 = [0, 2] \times [0, 2] \times [1.2, 1.6]$. In this example, caustics appear in $\Omega_R^1 \cup \Omega_R^2$, and we can see that our numerical solutions match with FDTD solutions very well.

Figure 19 shows more detailed comparison of the three components at different lines by the two different methods when $k_0 = 12\pi$. We can see that our numerical solutions agree well with the FDTD solution within the desired accuracy $\mathcal{O}(1/k_0)$ except around the primary source point $[1, 1, 0.2]^T$. Figure 20 show more numerical solutions at higher wavenumbers $k_0 = 24\pi$ and $k_0 = 32\pi$.

Next, we compute the Green's function $\boldsymbol{G}$ at a different primary source $[0.8, 1.0, 0.12]^T$. For $k_0 = 12\pi$, contour plots of numerical solutions for the three components, $\boldsymbol{G}_{xx}$, $\boldsymbol{G}_{zx}$ and $\boldsymbol{G}_{zz}$ at $y = 1$, are shown in Figures 21, 22, and 23, respectively. We can see that the Huygens sweeping solutions match with FDTD solutions very well.

5 Conclusions

We have proposed a new geometrical-optics method, the fast Huygens-sweeping method, for computing the Green's functions of Maxwell's equations in inhomogeneous media in high frequency regime. The new method utilizes the Huygens-Kirchhoff integral to integrate many locally valid asymptotic Green's functions into a globally valid asymptotic Green's functions. To accelerate the Huygens-Kirchhoff integration process, we have used the butterfly algorithm to speed up the Huygens-Kirchhoff summation. The new method has nearly optimal complexity in constructing high-frequency waves for a given source point and a given frequency parameter provided that asymptotic ingredients are precomputed. Numerical examples demonstrate the performance, efficiency, and accuracy of the new method.

We believe that the proposed methodology can be applied to many different problems, such as scattering problems and inverse problems, which constitute ongoing projects.

Acknowledgment

Qian is partially supported by NSF. Yuan is partially supported by the National Natural Science Foundation of China (project No. 11201508). Luo is partially supported by NSF DMS-1418908.

References

[1] T. Alkhalifah. Efficient traveltime compression for 3d prestack Kirchhoff migration. *Geophys. Prospect.*, 69(1):1–9, 2011.

[2] G. S. Avila and J. B. Keller. The high-frequency asymptotic field of a point source in an inhomogeneous medium. *Comm. Pure Appl. Math.*, 16:363–381, 1963.

[3] I. M. Babuška and S. A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Review*, 42:451–484, 2000.

[4] A. Bayliss, C. I. Goldstein, and E. Turkel. On accuracy conditions for the numerical computation of waves. *J. Comput. Phys.*, 59:396–404, 1985.

[5] J. D. Benamou. An introduction to Eulerian geometrical optics (1992 - 2002). *J. Sci. Comp.*, 19:63–93, 2003.

[6] J.-D. Benamou, O. Lafitte, R. Sentis, and I. Solliec. A geometrical optics-based numerical method for high frequency electromagnetic fields computation near fold caustics– Part I. *J. Comput. Appl. Math.*, 156:93–125, 2003.

[7] J.-D. Benamou, O. Lafitte, R. Sentis, and I. Solliec. A geometrical optics-based numerical method for high frequency electromagnetic fields computation near fold caustics– Part II. The energy. *J. Comput. Appl. Math.*, 167:91–134, 2004.

[8] J. P. Boyd. *Chebyshev and Fourier Spectral Methods.* Second edition, Dover, New York, 2001.

[9] R. Burridge. Asymptotic evaluation of integrals related to time-dependent fields near caustics. *SIAM J. Appl. Math.*, 55:390–409, 1995.

[10] E. Candes, L. Demanet, and L. Ying. A fast butterfly algorithm for the computation of Fourier integral operators. *SIAM Multiscale Model. Simul.*, 7:1727–1750, 2009.

[11] V. Cerveny, M. Popov, and I. Psencik. Computation of wave fields in inhomogeneous media-Gaussian beam approach. *Geophys. J. R. Astr. Soc.*, 70:109–128, 1982.

[12] W. C. Chew. *Waves and fields in inhomogeneous media.* IEEE PRESS, New York, 1995.

[13] L. Demanet, M. Ferrara, N. Maxwell, J. Poulson, and L. Ying. A butterfly algorithm for synthetic aperture radar imaging. *SIAM J. Imaging Sci.*, 5:203–243, 2012.

[14] B. Engquist and O. Runborg. Computational high frequency wave propagation. *Acta Numerica*, 12:181–266, 2003.

[15] B. Engquist and H.-K. Zhao. An approximate separability of Green's functions for high frequency Helmholtz equations. *submitted*, xx:xxx–xxx, 2014.

[16] S. Fomel, S. Luo, and H.-K. Zhao. Fast sweeping method for the factored eikonal equation. *J. Comput. Phys.*, 228(17):6440–6455, 2009.

[17] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.

[18] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[19] C. Y. Kao, S. Osher, and J. Qian. Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations. *J. Comput. Phys.*, 196:367–391, 2004.

[20] S. Leung and J. Qian. Eulerian Gaussian beam methods for Schrödinger equations in the semi-classical regime. *J. Comput. Phys.*, 228:2951–2977, 2009.

[21] S. Leung and J. Qian. The backward phase flow and FBI-transform-based Eulerian Gaussian beams for the Schrödinger equation. *J. Comput. Phys.*, 229:8888–8917, 2010.

[22] S. Leung, J. Qian, and R. Burridge. Eulerian Gaussian beams for high-frequency wave propagation. *Geophysics*, 72(5):SM61–SM76, 2007.

[23] S. Leung, J. Qian, and S. Serna. Fast Huygens Sweeping methods for Schrodinger equations in the semiclassical regime. *Methods Appl. Analy.*, 21:031–066, 2014.

[24] X. D. Liu, S. J. Osher, and T. Chan. Weighted Essentially NonOscillatory schemes. *J. Comput. Phys.*, 115:200–212, 1994.

[25] D. Ludwig. Uniform asymptotic expansions at a caustic. *Comm. Pure Appl. Math.*, 19:215–250, 1966.

[26] S. Luo and J. Qian. Factored singularities and high-order Lax-Friedrichs sweeping schemes for point-source traveltimes and amplitudes. *J. Comput. Phys.*, 230:4742–4755, 2011.

[27] S. Luo and J. Qian. Fast sweeping methods for factored anisotropic eikonal equations: multiplicative and additive factors. *J. Sci. Comput.*, 52:360–382, 2012.

[28] S. Luo, J. Qian, and R. Burridge. Fast Huygens sweeping methods for Helmholtz equations in inhomogeneous media in the high frequency regime. *J. Comput. Phys.*, 270:378–401, 2014.

[29] S. Luo, J. Qian, and R. Burridge. High-order factorization based high-order hybrid fast sweeping methods for point-source eikonal equations. *SIAM J. Numer. Analy.*, 52:23–44, 2014.

[30] S. Luo, J. Qian, and H.-K. Zhao. Higher-order schemes for 3-D traveltimes and amplitudes. *Geophysics*, 77:T47–T56, 2012.

[31] V. P. Maslov and M. V. Fedoriuk. *Semi-classical approximation in quantum mechanics.* D. Reidel Publishing Company, 1981.

[32] E. Michielssen and A. Boag. A multilevel matrix decomposition algorithm for analysing scattering from large structures. *IEEE Trans. Antennas Propagat.*, 44:1086–1093, 1996.

[33] J. Milnor. *Morse Theory, Annals of Math., No. 51.* Princeton University Press, 1973.

[34] M. Motamed and O. Runborg. Taylor expansion and discretization errors in Gaussian beam superposition. *Wave Motion*, 47:421–439, 2010.

[35] L. Novotny and B. Hecht. *Principles of nano-optics.* Cambridge, 2006.

[36] M. O'Neil and V. Rokhlin. A new class of analysis-based fast transforms. 2007.

[37] S. J. Osher and C. W. Shu. High-order Essentially NonOscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Analy.*, 28:907–922, 1991.

[38] A. Pica. Fast and accurate finite-difference solutions of the 3D eikonal equation parametrized in celerity. In *67th Ann. Internat. Mtg*, pages 1774–1777. Soc. of Expl. Geophys., 1997.

[39] M. M. Popov. A new method of computation of wave fields using Gaussian beams. *Wave Motion*, 4:85–97, 1982.

[40] M. M. Popov and I. Psencik. Computation of ray amplitudes in inhomogeneous media with curved interfaces. *Studia geop. et geod.*, 22:248–258, 1978.

[41] J. Qian, S. Luo, and R. Burridge. Fast Huygens sweeping methods for multiarrival Green's functions of Helmholtz equations in the high frequency regime. *Geophysics*, xxx:to appear, 2015.

[42] J. Qian and W. W. Symes. An adaptive finite difference method for traveltime and amplitude. *Geophysics*, 67:167–176, 2002.

[43] J. Qian and L. Ying. Fast Gaussian wavepacket transforms and Gaussian beams for the Schrödinger equation. *J. Comput. Phys.*, 229:7848–7873, 2010.

[44] J. Qian and L. Ying. Fast multiscale Gaussian wavepacket transforms and multiscale Gaussian beams for the wave equation. *SIAM J. Multiscale Modeling and Simulation*, 8:1803–1837, 2010.

[45] J. Ralston. Gaussian beams and the propagation of singularities. Studies in partial differential equations. MAA Studies in Mathematics, 23. Edited by W. Littman. pp.206-248., 1983.

[46] S. Serna and J. Qian. A stopping criterion for higher-order sweeping schemes for static Hamilton-Jacobi equations. *J. Comput. Math.*, 28:552–568, 2010.

[47] W. W. Symes and J. Qian. A slowness matching Eulerian method for multivalued solutions of eikonal equations. *J. Sci. Comp.*, 19:501–526, 2003.

[48] N. Tanushev, B. Engquist, and R. Tsai. Gaussian beam decomposition of high frequency wave fields. *J. Comput. Phys.*, 228:8856–8871, 2009.

[49] N. Tanushev, J. Qian, and J. Ralston. Mountain waves and Gaussian beams. *SIAM J. Multiscale Modeling and Simulation*, 6:688–709, 2007.

[50] A. Toflove and S. C. Huganess. *Computational Electrodynamics: The Finite Difference Time Domain Method, Second Editions.* Artech House, Norwood, MA, 2000.

[51] R. Tsai, L.-T. Cheng, S. J. Osher, and H. K. Zhao. Fast sweeping method for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Analy.*, 41:673–694, 2003.

[52] L. Tsang, J. A. Kong, and K-H Ding. *Scattering of Electromagnetic Waves: Theories and Applications.* John Wiley & Sons, Inc., 2000.

[53] B. S. White. The stochastic caustic. *SIAM J. Appl. Math.*, 44:127–149, 1984.

[54] B. S. White, A. Norris, A. Bayliss, and R. Burridge. Some remarks on the Gaussian beam summation method. *Geophys. J. R. Astr. Soc.*, 89:579–636, 1987.

[55] L. Ying. Sparse Fourier transform via butterfly algorithm. *SIAM J. Sci. Comput.*, 31:1678–1694, 2009.

[56] L. Zhang, J. W. Rector, and G. M. Hoversten. Eikonal solver in the celerity domain. *Geophysical Journal International*, 162:1–8, 2005.

[57] Y.-T. Zhang, H.-K. Zhao, and J. Qian. High order fast sweeping methods for static Hamilton-Jacobi equations. *J. Sci. Comput.*, 29:25–56, 2006.

[58] H. K. Zhao. Fast sweeping method for eikonal equations. *Math. Comp.*, 74:603–627, 2005.

[59] M. Zhdanov. *Geophysical inverse theory and regularization problems*, volume 36. Elsevier, 2002.

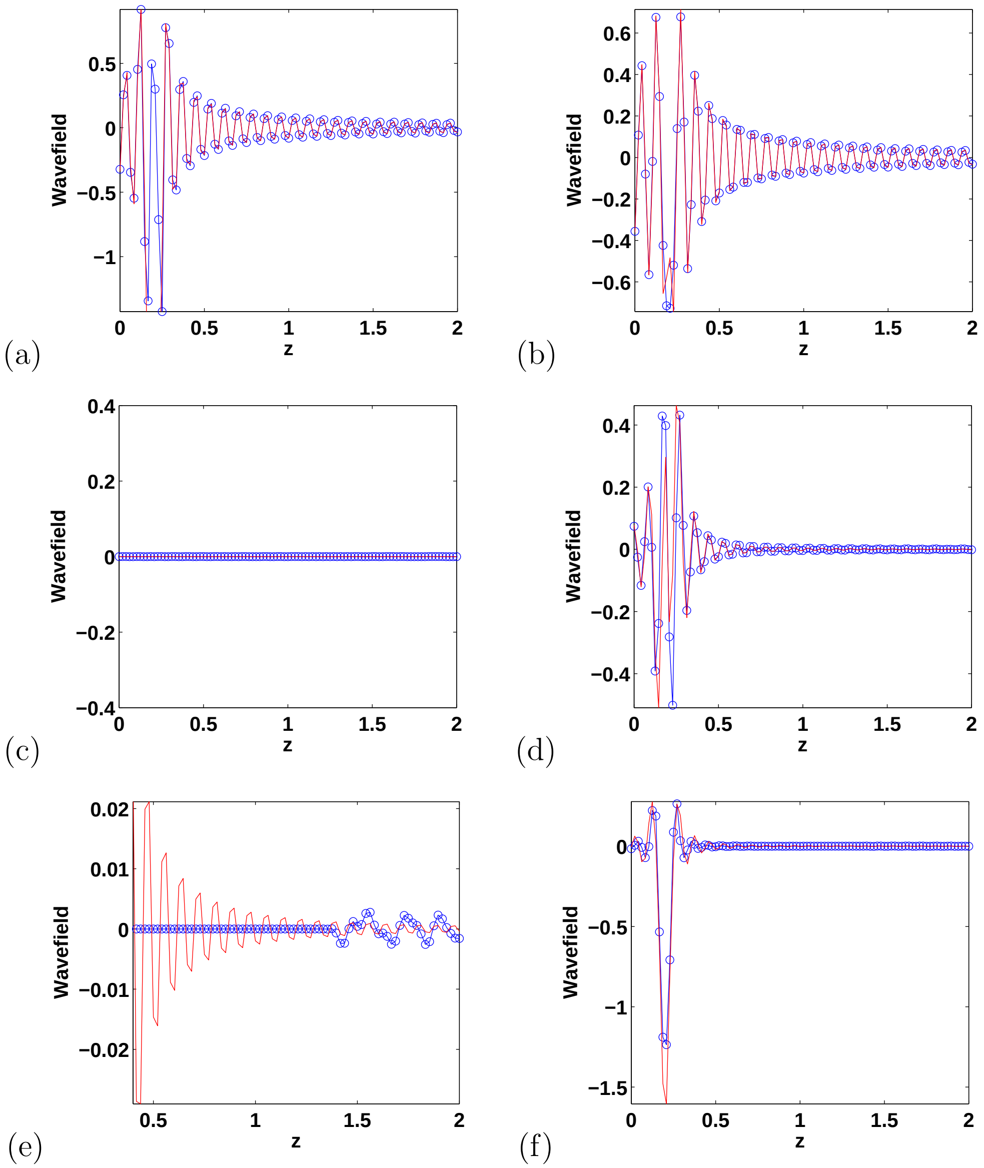


Figure 6: Constant refractive index model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 24\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at (a): line $x = 1, y = 1$ and (b): line $x = \frac{23}{24}$ and $y = 1$; real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at (c): line $x = 1, y = 1$ and (d): line $x = \frac{23}{24}$ and $y = 1$; real part of the zz-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at (e): line $x = 1$ and $y = 1$ and (f): line $x = \frac{23}{24}$ and $y = 1$. $k_0 = 24\pi$ and Chebyshev order $p = 11$. Solid line: exact solution; Circle line: Huygens sweeping solution.

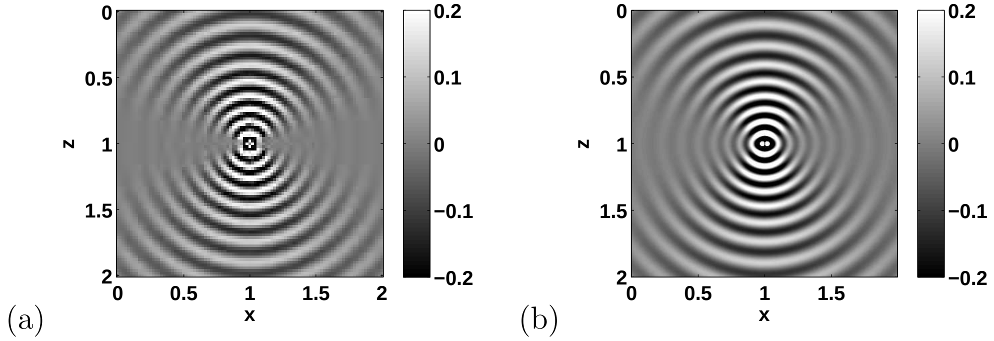


Figure 7: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 1.0]^T$ and $k_0 = 10\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 97$; mesh in (b): $401 \times 401 \times 401$.

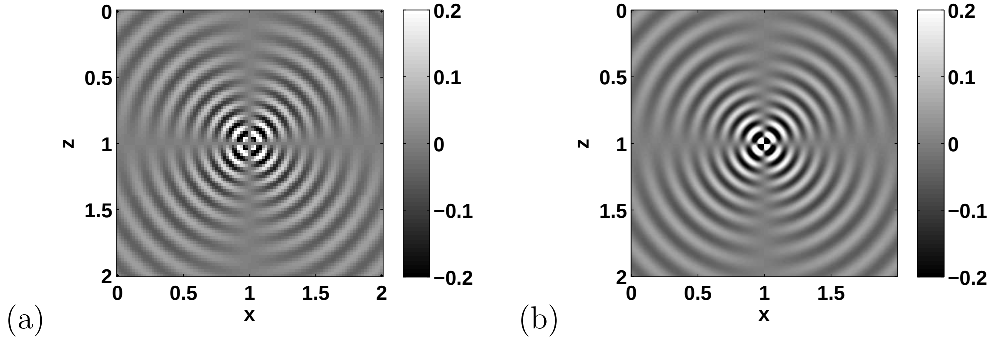


Figure 8: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 1.0]^T$ and $k_0 = 10\pi$. Real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 97$; mesh in (b): $401 \times 401 \times 401$.

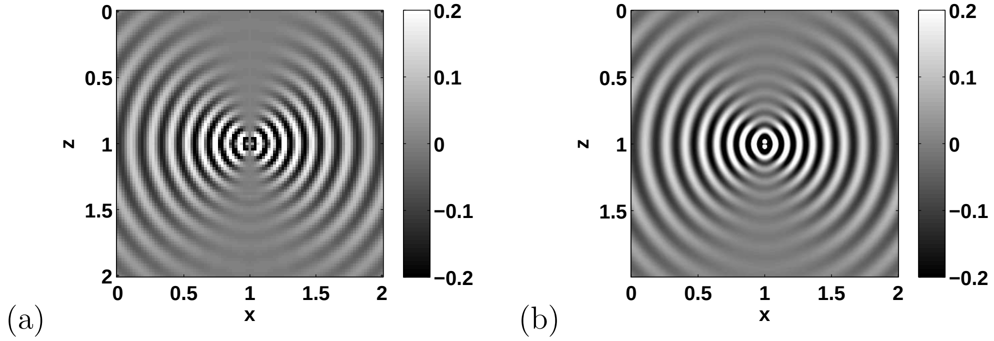


Figure 9: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 1.0]^T$ and $k_0 = 10\pi$. Real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 97$; mesh in (b): $401 \times 401 \times 401$.

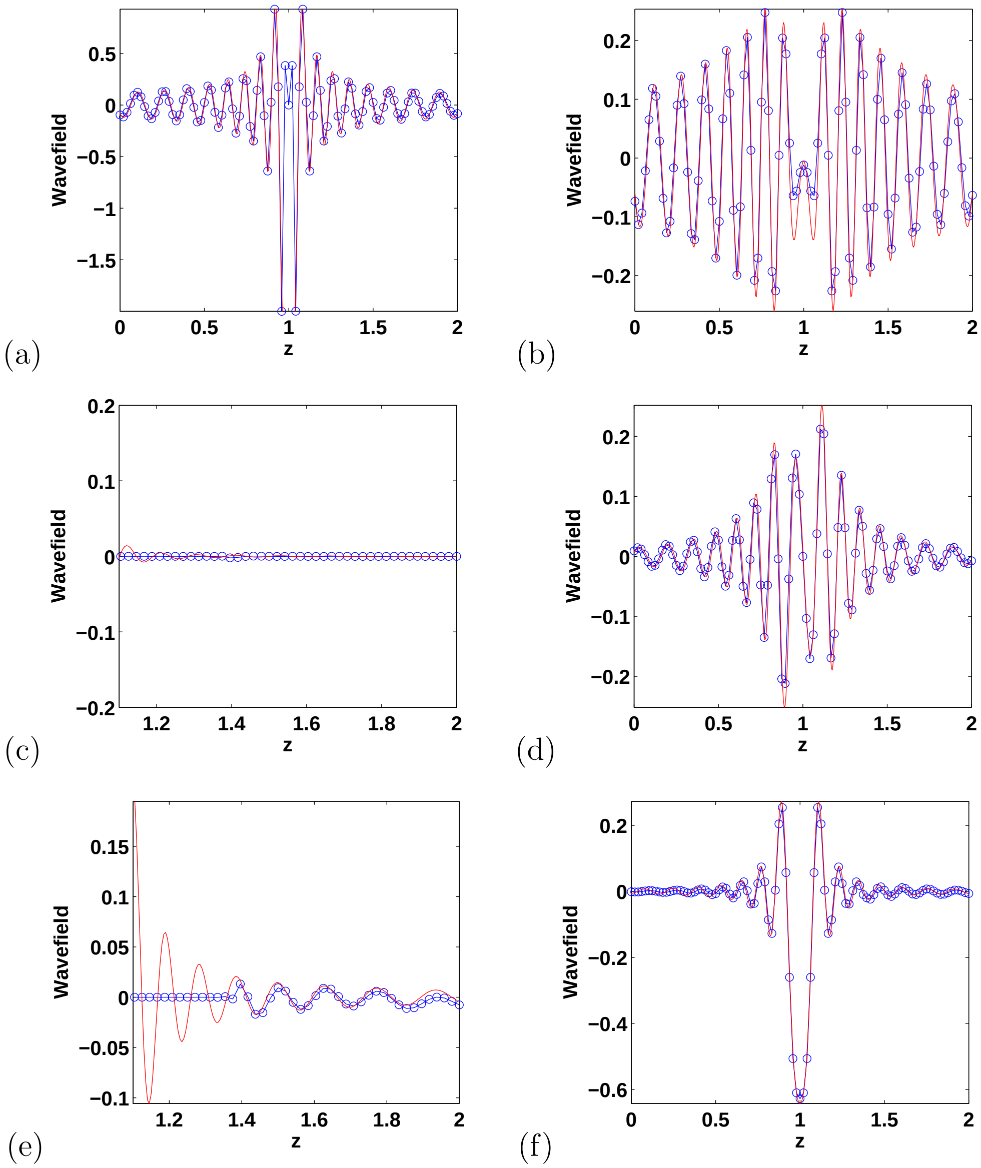


Figure 10: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 1.0]^T$ and $k_0 = 10\pi$. The xx-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at (a): line $x = 1, y = 1$ and (b): line $x = \frac{7}{8}$ and $y = 1$; the zx-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at (c): line $x = 1, y = 1$ and (d): line $x = \frac{7}{8}$ and $y = 1$; the zz-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at (e): line $x = 1, y = 1$ and (f): line $x = \frac{7}{8}$ and $y = 1$. Solid line: FDTD-based solution; circle line: Huygens sweeping solution.

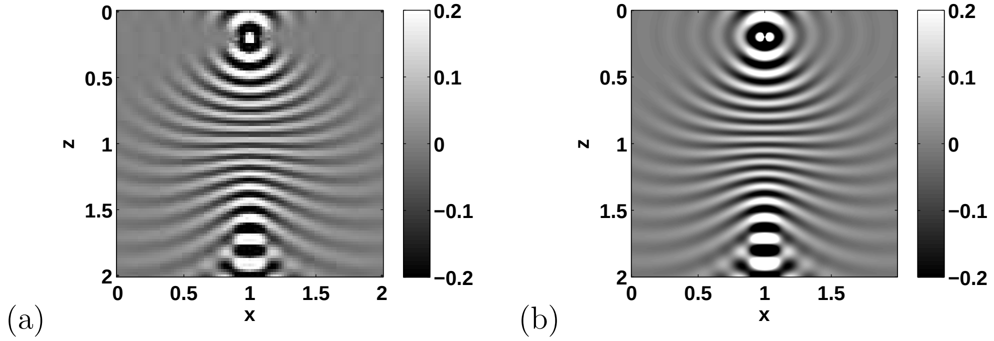


Figure 11: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 10\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 97$; mesh in (b): $401 \times 401 \times 401$.

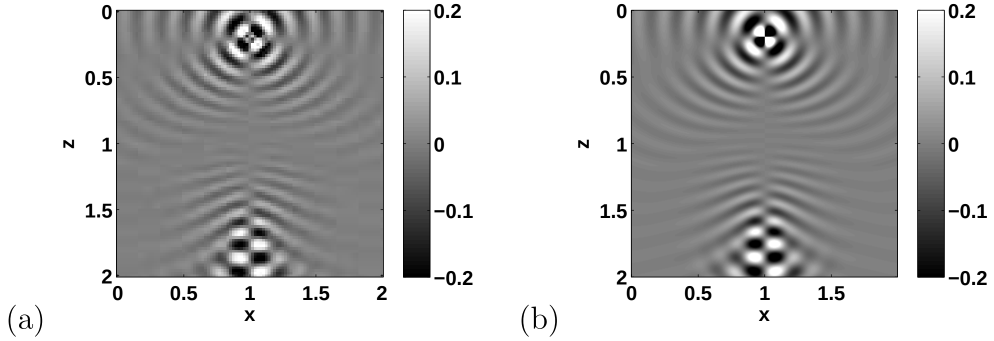


Figure 12: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 10\pi$. Real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 97$; mesh in (b): $401 \times 401 \times 401$.

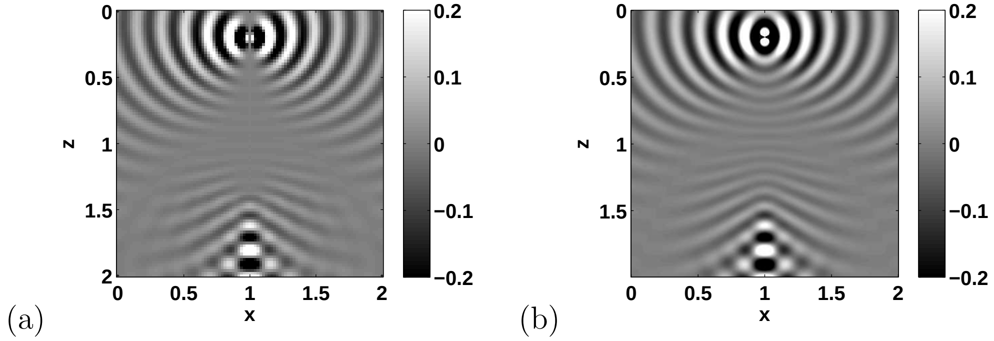


Figure 13: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 10\pi$. Real part of the zz-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 97$; mesh in (b): $401 \times 401 \times 401$.

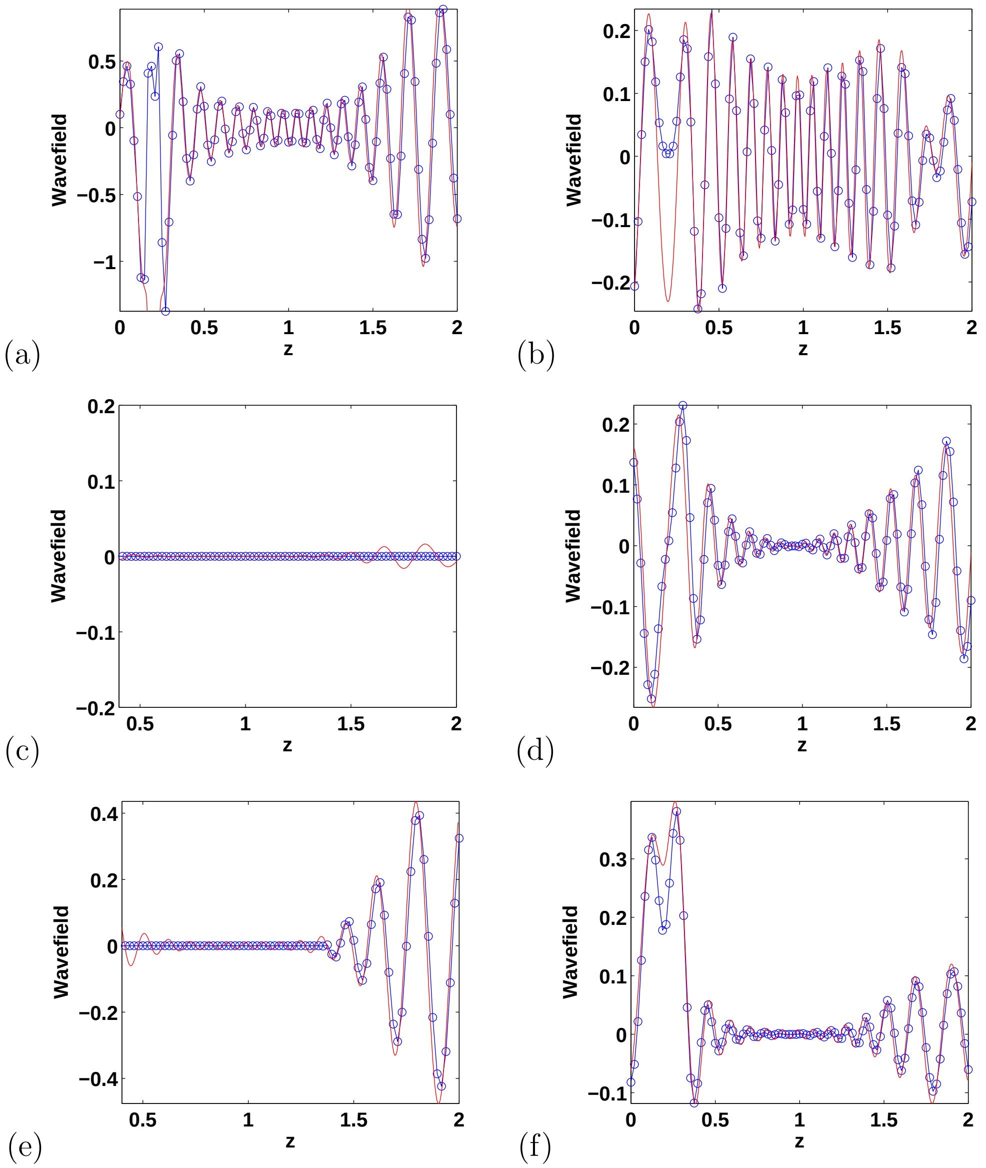


Figure 14: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 10\pi$. Real part of xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at (a): line $x = 1, y = 1$ and (b): line $x = \frac{7}{8}$ and $y = 1$; real part of zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at (c): line $x = 1, y = 1$ and (d): line $x = \frac{7}{8}$ and $y = 1$; real part of zz-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at (e): line $x = 1, y = 1$ and (f): line $x = \frac{7}{8}$ and $y = 1$. Solid line: FDTD-based solution; circle line: Huygens sweeping solution.

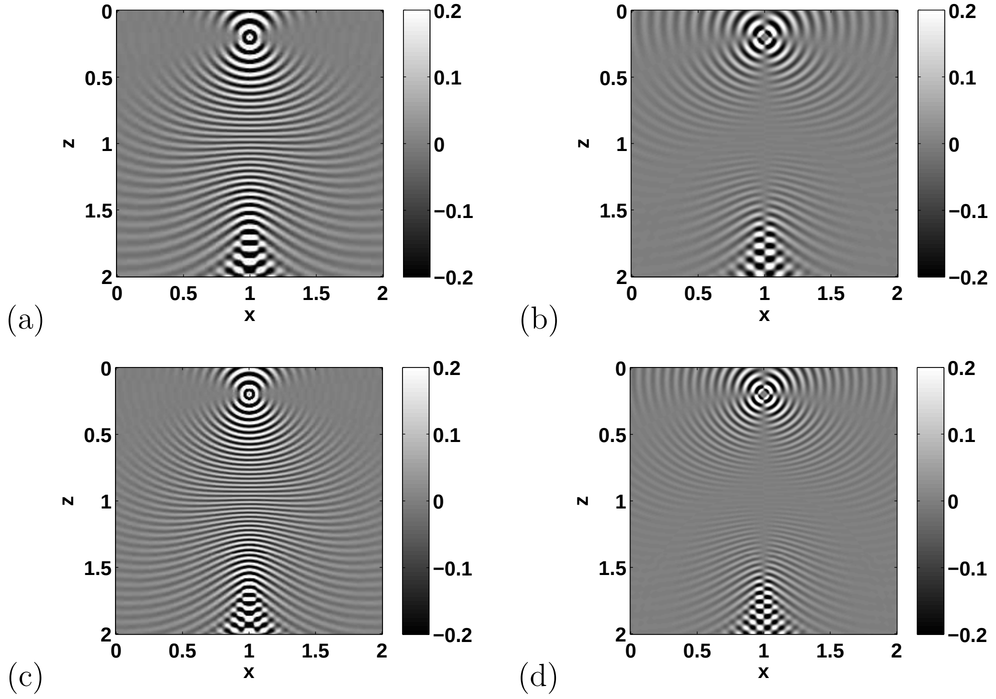


Figure 15: Gaussian model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$. Real parts of (a): xx- and (b): zx-components of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by the Huygens sweeping method with $p = 11$ at free-space wavenumber $k_0 = 20\pi$ with mesh $193 \times 193 \times 193$. Real parts of (c): xx- and (d): zx- components of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by the Huygens sweeping method with $p = 11$ at free-space wavenumber $k_0 = 80/3\pi$ with mesh $257 \times 257 \times 257$.

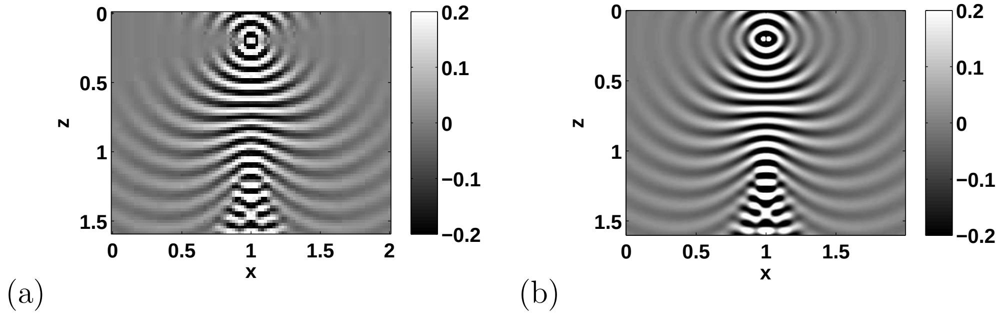
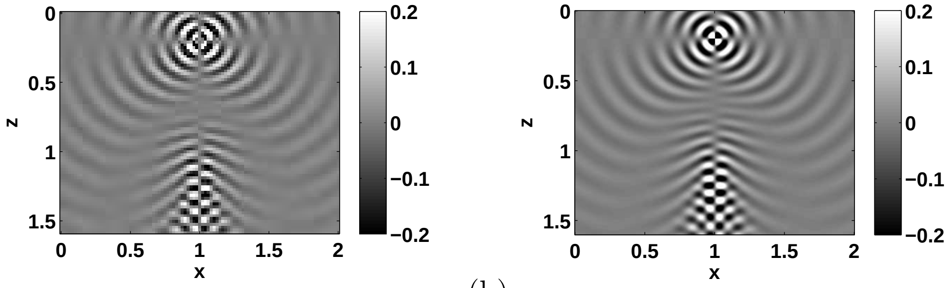
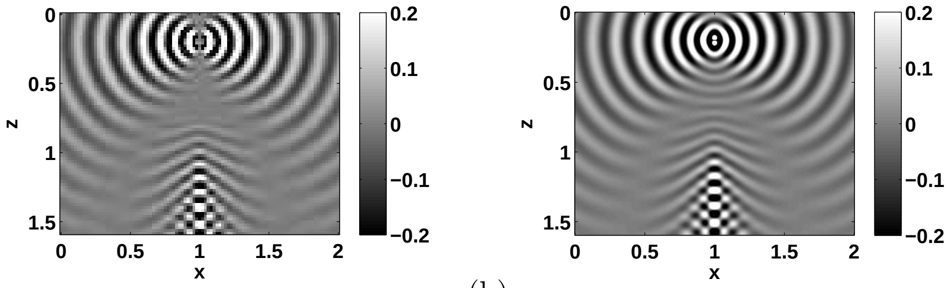


Figure 16: Waveguide model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 12\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 77$; mesh in (b): $401 \times 401 \times 321$.



(a) (b)

Figure 17: Waveguide model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 12\pi$. Real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 77$; mesh in (b): $401 \times 401 \times 321$.



(a) (b)

Figure 18: Waveguide model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 12\pi$. Real part of the zz-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD, at the free space wavenumber $k_0 = 12\pi$. Mesh in (a): $97 \times 97 \times 77$; mesh in (b): $401 \times 401 \times 321$.

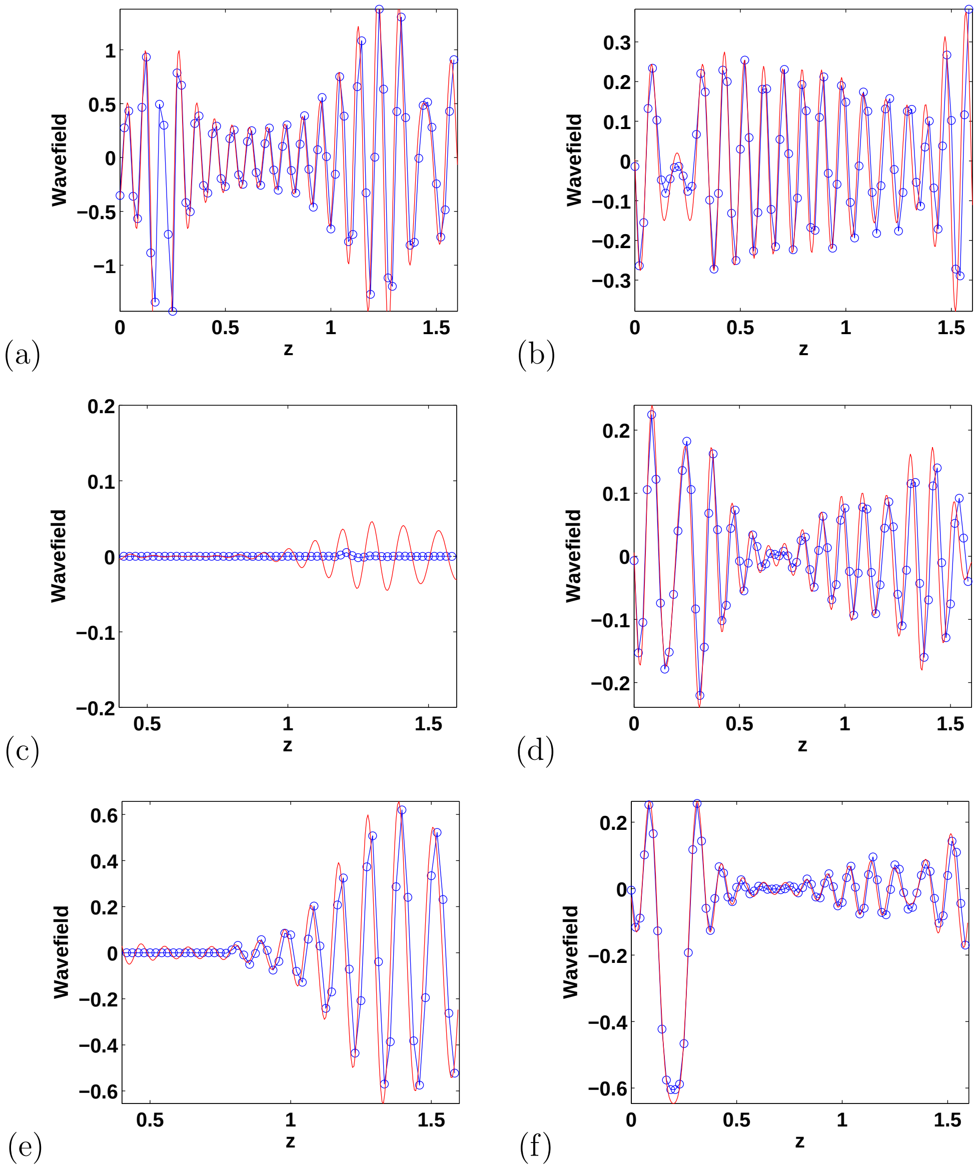
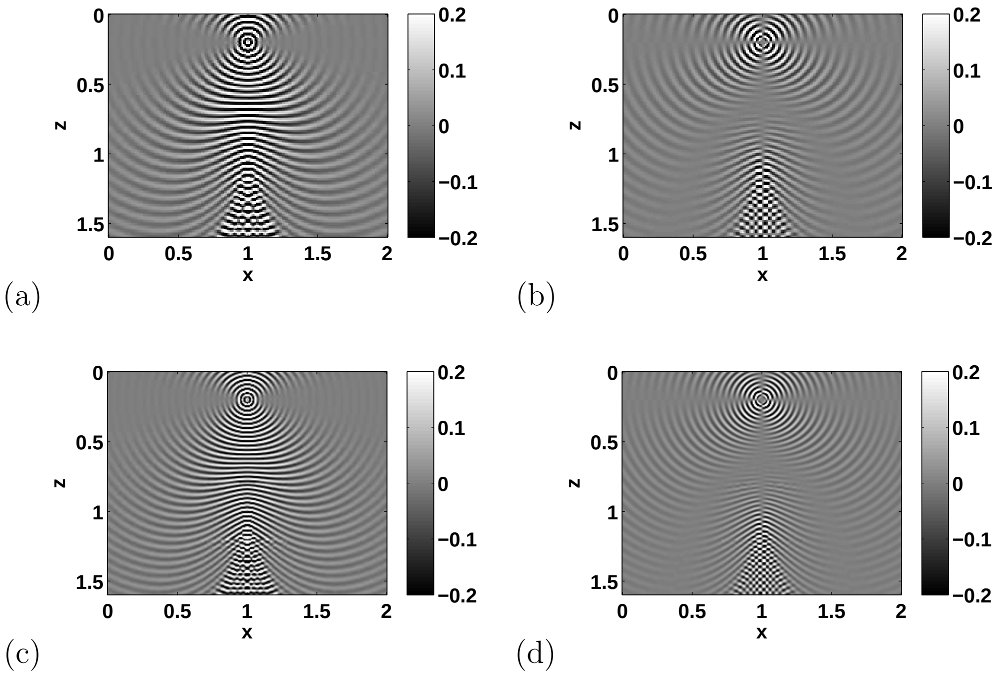


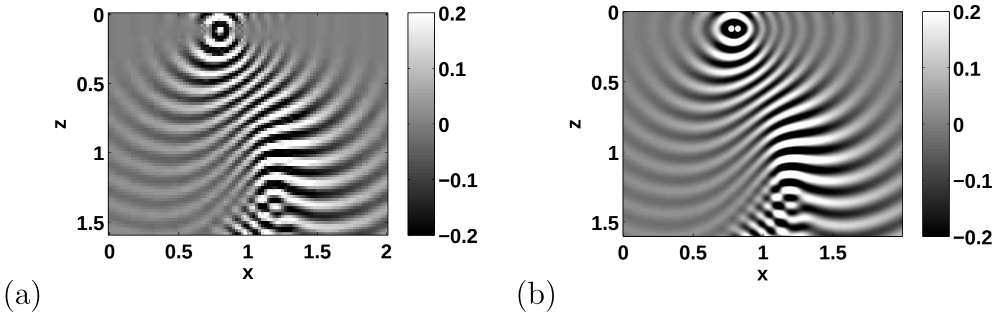
Figure 19: Waveguide model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$ and $k_0 = 12\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at (a): line $x = 1, y = 1$ and (b): line $x = \frac{9}{8}$ and $y = 1$; real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at (c): line $x = 1, y = 1$ and (d): line $x = \frac{9}{8}$ and $y = 1$; real part of the zz-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at (e): line $x = 1, y = 1$ and (f): line $x = \frac{9}{8}$ and $y = 1$. Solid line: FDTD-based solution; circle line: Huygens sweeping solution.



(a) (b)

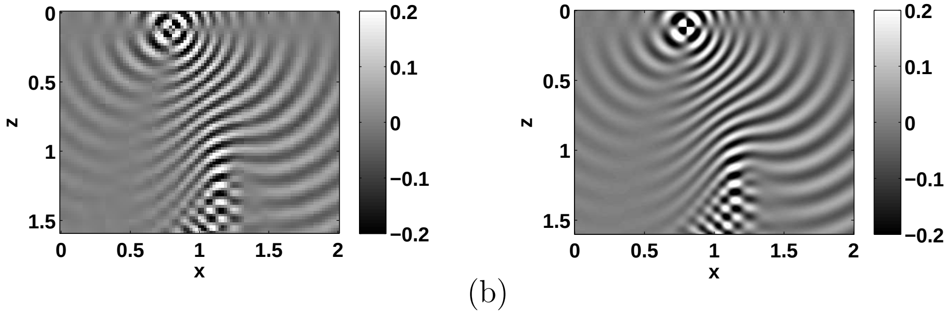
(c) (d)

Figure 20: Waveguide model with $\boldsymbol{r}_0 = [1.0, 1.0, 0.2]^T$. Real parts of (a): xx- and (b): zx-components of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by the Huygens sweeping method with $p = 11$ at free-space wavenumber $k_0 = 24\pi$ with mesh $193 \times 193 \times 154$. Real parts of (c): xx- and (d): zx- components of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by the Huygens sweeping method with $p = 11$ at free-space wavenumber $k_0 = 32\pi$ with mesh $257 \times 257 \times 205$.



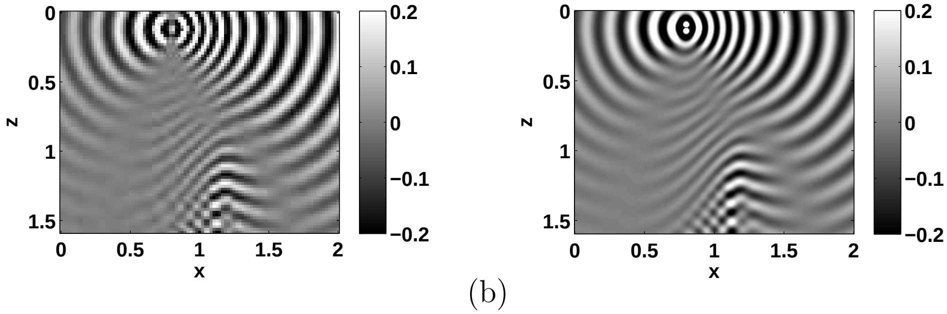
(a) (b)

Figure 21: Waveguide model with $\boldsymbol{r}_0 = [0.8, 1.0, 0.12]^T$ and $k_0 = 12\pi$. Real part of the xx-component of $\boldsymbol{G}(\boldsymbol{r}; \boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 77$; mesh in (b): $401 \times 401 \times 321$.



(a) (b)

Figure 22: Waveguide model with $\boldsymbol{r}_0 = [0.8, 1.0, 0.12]^T$ and $k_0 = 12\pi$. Real part of the zx-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 77$; mesh in (b): $401 \times 401 \times 321$.



(a) (b)

Figure 23: Waveguide model with $\boldsymbol{r}_0 = [0.8, 1.0, 0.12]^T$ and $k_0 = 12\pi$. Real part of the zz-component of $\boldsymbol{G}(\boldsymbol{r};\boldsymbol{r}_0)$ at $y = 1$ computed by (a): fast Huygens sweeping method with $p = 11$; (b): FDTD. Mesh in (a): $97 \times 97 \times 77$; mesh in (b): $401 \times 401 \times 321$.