

Babich's Expansion and the Fast Huygens Sweeping Method for the Helmholtz Wave Equation at High Frequencies

Wangtao Lu ^{*} Jianliang Qian [†] Robert Burridge [‡]

May 14, 2015

Abstract

In some applications, it is reasonable to assume that geodesics (rays) have a consistent orientation so that the Helmholtz equation can be viewed as an evolution equation in one of the spatial directions. With such applications in mind, starting from Babich's expansion, we develop a new high-order asymptotic method, which we dub *the fast Huygens sweeping method*, for solving point-source Helmholtz equations in inhomogeneous media in the high-frequency regime and in the presence of caustics. The first novelty of this method is that we develop a new Eulerian approach to compute *the asymptotics*, i.e. the traveltime function and amplitude coefficients that arise in Babich's expansion, yielding a locally valid solution, which is accurate close enough to the source. The second novelty is that we utilize the Huygens-Kirchhoff integral to integrate many locally valid wavefields to construct globally valid wavefields. This automatically treats caustics and yields uniformly accurate solutions both near the source and remote from it. The third novelty is that the butterfly algorithm is adapted to accelerate the Huygens-Kirchhoff summation, achieving nearly optimal complexity $\mathcal{O}(N \log N)$, where N is the number of mesh points; the complexity prefactor depends on the desired accuracy and is independent of the frequency. To reduce the storage of the resulting tables of asymptotics in Babich's expansion, we use the multivariable Chebyshev series expansion to compress each table by encoding the information into a small number of coefficients.

The new method enjoys the following desired features. First, it precomputes the asymptotics in Babich's expansion, such as traveltime and amplitudes. Second, it

^{*}Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA. Email: wangtaol@math.msu.edu

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

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

takes care of caustics automatically. Third, it can compute the point-source Helmholtz solution for many different sources at many frequencies simultaneously. Fourth, for a specified number of points per wavelength, it can construct the wavefield 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 frequency. Both two-dimensional and three-dimensional numerical experiments have been carried out to illustrate the performance, efficiency, and accuracy of the method.

1 Introduction

We consider the general point-source Helmholtz equation in the m -dimensional space \mathbb{R}^m :

$$\nabla \cdot (\mu \nabla u) + \omega^2 \rho u = -\delta(\mathbf{r} - \mathbf{r}_0), \quad (1)$$

with the Sommerfeld radiation condition imposed at infinity, where \mathbf{r}_0 is the source location, position $\mathbf{r} = [x_1, \dots, x_m]^T$, the gradient operator $\nabla = [\partial_{x_1}, \dots, \partial_{x_m}]^T$, ω is the angular frequency, and both variables ρ and μ are analytic and positive functions of position \mathbf{r} , characterizing certain physical parameters of the medium. The point-source solution, also called the Green's function excited by \mathbf{r}_0 , is used in a variety of applications, such as medical imaging, seismic imaging, underwater acoustics, and synthetic aperture radar. Since in the high frequency regime, the popular finite-difference or finite-element methods require extremely refined grids of mesh points to prevent so-called pollution or dispersion error [5, 4], we seek more effective methods to compute this point-source Helmholtz solution. In this paper, we develop a high-order, high-frequency, asymptotic method, which we call *the fast Huygens sweeping method*, for solving the Helmholtz equation (1) in the high frequency regime and in the presence of caustics, based on Babich's rarely used asymptotic ansatz described in [3]. This ansatz is the Fourier transform in time of the one used by Hadamard [18] as outlined by Courant and Hilbert [13], Chapter VI, Section 15.6. Underlying this new method there are three critical elements: a novel Babich-ansatz based formulation for computing locally valid solutions, the Huygens secondary-source principle for constructing globally valid solutions, and an adapted butterfly algorithm to expedite the whole process.

The usual asymptotic expansion of the point-source solution to (1) is sought as

$$u(\mathbf{r}; \mathbf{r}_0) = e^{i\omega\tau} \sum_{s=0}^{\infty} A_s(\mathbf{r}; \mathbf{r}_0) \frac{1}{(i\omega)^{s-(m-1)/2}}, \quad (2)$$

where τ is the travel time from \mathbf{r}_0 to \mathbf{r} satisfying the eikonal equation

$$|\nabla\tau(\mathbf{r}; \mathbf{r}_0)| = n(\mathbf{r}), \quad (3)$$

with a point source boundary condition $\tau(\mathbf{r}_0; \mathbf{r}_0) = 0$, the refractive index $n = \sqrt{\rho/\mu}$, and the $\{A_s\}$ satisfy a recursive system of ODEs along the ray,

$$2\mu\nabla\tau \cdot \nabla A_s + A_s \nabla \cdot (\mu \nabla\tau) = -\nabla \cdot (\mu \nabla A_{s-1}), \quad (4)$$

for $s = 0, 1, \dots$, with $A_{-1} \equiv 0$. However, one difficulty arises immediately: how to choose the initial data for those $\{A_i\}$. Moreover, when $m = 2$, this usual ray series does not yield an accurate solution close to the source. For constant μ , the problem of initializing starting values of $\{A_i\}$ when $m = 3$ have been solved in [2] and is left incomplete when $m = 2$; in practice, such difficulties were handled by initializing $\{A_i\}$ slightly away from the point source by using the solution for a medium with a constant refractive index equal to that at the source point [53, 42, 27, 31, 30]; the resulting numerical asymptotic solution is not uniform near the source. To resolve the difficulties, Babich [3] proposed a ray series, essentially the Fourier transform in time of Hadamard's method for the wave equation, defined in terms of Hankel functions of the first kind, where the initial data for the frequency-independent coefficients, or asymptotics, are easily specified by comparison with the exact solution for a uniform medium near the source. By computing these, the new expansion yields uniformly accurate solutions in a region containing the source and up to but not including the first focal point (first contact with a caustic) on each ray.

The usefulness of this ansatz for numerical simulation has been justified in our recent work [45] when $\mu = 1$. In this paper, we extend the application of Babich's ansatz to the more general self-adjoint point-source Helmholtz equation (1) with variable μ .

The one-term truncation of the usual geometric-optics series (2), as illustrated in many contexts [6, 15, 17, 30, 41], is not able to characterize the caustics, which will occur with high probability for wave propagation in inhomogeneous media. Babich's ansatz cannot overcome this issue either since it requires the traveltime τ to be smooth [3], which fails at caustics. To treat caustics, we will use the Huygens secondary source principle. It has been shown that in isotropic media, as considered here, which are smooth, the squared eikonal solution of equation (3) is also smooth in a neighborhood of the source \mathbf{r}_0 [2, 34, 48], implying that there is a neighborhood of the source which is free of caustics, and hence Babich's ansatz constructs a locally valid, short-wave asymptotic solution. Based on such an observation and the Huygens secondary source principle, we are able to develop a Huygens sweeping algorithm to construct a globally valid, or far-wave solution by using the Huygens-Kirchhoff integral to patch together many short-wave solutions, so that caustics are automatically taken care of.

To go beyond caustics, we will make some assumptions for this point-source Helmholtz equation under consideration. For some seismic applications, it is natural to assume that a set of geodesics (rays) have a consistent direction so that the Helmholtz equation (1) may be viewed as an evolution equation. Thus along this direction, we partition the computational domain into layered subdomains such that a layer-by-layer sweeping solver can be developed using Huygens' principle to construct a globally valid solution. Specifically, the first layer is identified as a local neighborhood of \mathbf{r}_0 , where Babich's expansion is valid. Next, we identify a neighborhood outside the first layer as the second layer so that Babich's expansion for the Green's function excited by any secondary source on the first layer boundary is caustic-free in the second layer. By using the Huygens-Kirchhoff integral to integrate these locally valid Green's functions over the first layer boundary, we construct the wavefield in the second layer. Repeating this sweeping process, we construct a globally valid solution in the whole computational

domain. In this way caustics are treated automatically.

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 in the Babich-expansion ansatz in that the traveltime function satisfying the eikonal equation has an upwind singularity at the primary source so that it is difficult to compute this function with high-order accuracy; moreover, the occurrence of the Laplacian of the traveltime in the transport equations makes the task more challenging. To deal with this obstacle, we use the newly developed high-order schemes [27, 30, 31, 45] for computing first-arrival traveltimes.

The second obstacle is how to store the many tables of asymptotics that we will generate for the many secondary sources. This storage issue is critical as we are aiming at solving Helmholtz equations in both 2-D and 3-D cases. We reduce data storage by expressing each asymptotic in terms of a multivariable Chebyshev series expansion. Thus we compress each table into a small number of Chebyshev coefficients. Computationally, such compression leads to a significant storage reduction and efficient access to memory.

The third obstacle is how to carry out efficiently the dense matrix-vector products required by the Huygens-Kirchhoff integration. Let J be the number of mesh points along each coordinate direction of the computational domain, so that the total number of mesh points is $N = J^m$ in the m -dimensional case. Because we are interested in the asymptotic solution everywhere in the computational domain, the solution at observation points (receivers) in the m -D case corresponds to the result of some matrix-vector products. In 2-D cases straightforward implementation of the above matrix-vector products requires $\mathcal{O}(N)$ operations for each 1-D straight line of receivers, and the total computational cost will be $\mathcal{O}(N^{3/2})$ as we need to carry out such matrix-vector products for roughly $N^{1/2}$ planes of receivers; in 3-D cases straightforward implementation of the above matrix-vector products requires $\mathcal{O}(N^{4/3})$ operations for each plane of receivers, and the total computational cost will be $\mathcal{O}(N^{5/3})$ as we need to carry out such matrix-vector products for roughly $N^{1/3}$ planes of receivers. Such computational cost is far too high to make our strategy practical. To tackle this difficulty, we adapt to our application the multilevel matrix decomposition based butterfly algorithm [33, 36, 11, 56, 14, 29, 41, 40]. The resulting butterfly algorithm allows us to carry out the required matrix-vector products with the total computational cost of $\mathcal{O}(N \log N)$ complexity, where the proportionality constant depends only on the specified accuracy and is independent of the frequency parameter ω . 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 advantages which may be attributed to the precomputed tables of traveltimes and amplitudes. First, because the traveltime and amplitudes are independent of the frequency, the precomputed tables can be used to construct the wavefield for a given primary source at arbitrary frequencies. Second, those tables can be used to construct the wavefield at many other primary sources for arbitrary frequencies as well. These two merits are much desired in many applications, such as seismic imaging and inversion.

1.1 Related work

The high-order schemes for the eikonal and transport equations that we are using here were developed in [27, 30, 31], which in turn are based on Lax-Friedrichs sweeping [21, 58, 59, 52, 47, 45], weighted essentially non-oscillatory (WENO) finite-difference approximation [37, 25, 20, 19], and factorization of the upwind source singularities [38, 57, 17, 27, 31, 28, 30]. 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.

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, amplitudes and related ingredients involved in the sweeping process, as inspired by the work in [1].

To construct a globally valid asymptotic Helmholtz solution 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 [26, 10], which require that the caustic structure be given. The second approach is based on the Maslov canonical operator theory [32]. 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 [54]. The third approach is based on Gaussian beam methods [12, 39, 46, 55, 24, 50, 49]. 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 [24, 43, 44, 22, 23, 35, 49], and such shortcomings sometimes have hindered applications of Gaussian beam methods to complicated inhomogeneous media. In addition, Benamou et al. [7, 8] 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 [33], and it has been further developed in [36, 56, 11, 14]. In this work, we are using the version of the fast butterfly algorithm first developed in [11] and further analyzed in [14]. This fast butterfly algorithm was then adapted for Helmholtz equations [29, 41] and for Maxwell's equations [40]. The significance of the fast butterfly algorithm for high-frequency wave computation was illustrated in a recent work [16]. Engquist and Zhao [16] showed that the ϵ -approximate separability of $G(\mathbf{r}; \mathbf{r}_0) = A(\mathbf{r}; \mathbf{r}_0)e^{i\omega\tau(\mathbf{r}; \mathbf{r}_0)}$ has ω -dependent lower and upper bounds when $\mathbf{r} \in X$ and $\mathbf{r}_0 \in Y$ with $X \subset \mathbb{R}^3$ and $Y \subset \mathbb{R}^3$ disjoint and compact. This result has two implications. The first implication is that when ω 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 ω 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 an ω -dependent manner which is exactly the departure point for fast butterfly algorithms in [11, 29, 41, 40] and in the current paper.

1.2 Plan of the paper

In section 2, by introducing Babich's expansion, we propose a novel formulation to construct a locally valid solution of equation (1). To construct the globally valid solution, we develop a Huygens-principle based sweeping algorithm in section 3. In sections 4 and 5, we present details of numerical implementations, along with a complexity analysis of our sweeping algorithm. Extensive numerical experiments are carried out in section 6 to illustrate the performance, efficiency and accuracy of our new method.

2 Babich's expansion based local solution

2.1 Babich's expansion

Using Babich's ansatz [3] we seek an asymptotic solution to equation (1) as $\omega \rightarrow \infty$ in a deleted neighborhood of \mathbf{r}_0 in the following form

$$u(\mathbf{r}; \omega) = \sum_{s=0}^{\infty} v_s(\mathbf{r}) f_{s+1-\frac{m}{2}}(\tau(\mathbf{r}); \omega), \quad (5)$$

where τ is the traveltime from \mathbf{r}_0 to \mathbf{r} ,

$$f_p(\tau, \omega) = i \frac{\sqrt{\pi}}{2} e^{ip\pi} \left(\frac{2\tau}{\omega} \right)^p H_p^{(1)}(\omega\tau) = i \frac{\sqrt{\pi}}{2} \left(\frac{2\tau}{\omega} \right)^p H_{-p}^{(1)}(\omega\tau),$$

$H_p^{(1)}$ is the p -th Hankel function of the first kind, and the $\{v_s\}$, for $s \geq 0$, are assumed to be smooth functions in the source neighborhood.

Substituting equation (5) into equation (1), and using the formulas [3]

$$\frac{1}{\tau} \frac{\partial f_p(\tau; \omega)}{\partial \tau} = -2f_{p-1}(\tau; \omega), \quad (6)$$

$$\omega^2 f_p(\tau; \omega) = 4(1-p)f_{p-1}(\tau; \omega) - 4\tau^2 f_{p-2}(\tau; \omega), \quad (7)$$

we obtain

$$\begin{aligned} \sum_{s=-\infty}^{\infty} f_{s+1-m/2} [4v_{s+2}\mu\tau^2 |\nabla\tau|^2 - 4v_{s+2}\rho\tau^2 - 2\nabla \cdot (v_{s+1}\mu\tau\nabla\tau) \\ - 2\mu\tau\nabla\tau \cdot \nabla v_{s+1} - 4(s+1-m/2)\rho v_{s+1} + \nabla \cdot (\mu\nabla v_s)] = 0, \end{aligned} \quad (8)$$

where we have made the convention that $v_s \equiv 0$ when $s \leq -1$.

Meanwhile, since for large ω [3],

$$\frac{f_{p+1}(\tau; \omega)}{f_p(\tau; \omega)} = \mathcal{O}\left(\frac{1}{\omega}\right), \quad (9)$$

the coefficient of every $f_{s+1-m/2}$ in equation (8) must be zero so that we get, for $s = -2, -1, \dots$,

$$4v_{s+2}\mu\tau^2|\nabla\tau|^2 - 4v_{s+2}\rho\tau^2 - 2\nabla \cdot (v_{s+1}\mu\tau\nabla\tau) - 2\mu\tau\nabla\tau \cdot \nabla v_{s+1} - 4(s+1-m/2)\rho v_{s+1} + \nabla \cdot (\mu\nabla v_s) = 0. \quad (10)$$

In particular, when $s = -2$, remembering that $v_{-2} \equiv v_{-1} \equiv 0$, we obtain

$$4v_0\tau^2(\mu|\nabla\tau|^2 - \rho) = 0, \quad (11)$$

which leads to the same eikonal equation (3), when we define

$$n = \sqrt{\frac{\rho}{\mu}}. \quad (12)$$

Then for general $s \geq -1$, equation (10) reduces to a recursive system of transport equations

$$4\mu\tau\nabla\tau \cdot \nabla v_{s+1} + v_{s+1}[2(2s+2-m)\rho + 2\nabla \cdot (\mu\tau\nabla\tau)] = \nabla \cdot (\mu\nabla v_s). \quad (13)$$

2.2 Interpretation as energy conservation when $s = -1$

We now study equation (13) for the leading term, i.e., for $s = -1$,

$$4\mu\tau\nabla\tau \cdot \nabla v_0 + v_0[-2m\rho + 2\nabla \cdot (\mu\tau\nabla\tau)] = 0. \quad (14)$$

Notice that the amplitude of the leading term in equation (5) is not v_0 but v_0 multiplied by the leading amplitude for ω large of $f_{1-m/2}$. But this amplitude is

$$\mathcal{O}\left(\tau^{1-m/2}\tau^{-1/2}\right) = \mathcal{O}\left(\tau^{\frac{1-m}{2}}\right), \quad (15)$$

where the extra $\tau^{-1/2}$ is attributed to

$$|H_{1-m/2}^{(1)}(\omega\tau)| \simeq \sqrt{\frac{2}{\pi\omega}}\tau^{-1/2},$$

for large ω . Therefore, the true leading amplitude is (proportional to) $u_0 = v_0\tau^{\frac{1-m}{2}}$. We set $v_0 = \tau^{\frac{m-1}{2}}u_0$ in equation (13) and get

$$4\mu\tau\nabla\tau \cdot \nabla u_0 + 2u_0\tau\nabla\mu \cdot \nabla\tau + 2u_0\mu\tau\Delta\tau = 0, \quad (16)$$

which can be further simplified to

$$2\mu\nabla\tau \cdot \nabla u_0 + u_0\nabla \cdot (\mu\nabla\tau) = 0. \quad (17)$$

Finally, on multiplying by u_0 , we get

$$\nabla \cdot (\mu u_0^2 \nabla \tau) = 0, \quad (18)$$

or

$$\nabla \cdot (\rho u_0^2 c^2 \nabla \tau) = 0, \quad (19)$$

where $c = \frac{1}{n} = \sqrt{\frac{\mu}{\rho}}$ is the magnitude of ray velocity.

As $c^2 \nabla \tau$ is the ray velocity vector, if we think ρu_0^2 as the energy density, then $\rho u_0^2 c^2 \nabla \tau$ becomes the energy flux vector. Thus, the divergence of the energy flux is zero so that energy is conserved to leading order, and energy flux is conserved along tubes of rays as it should be. This also verifies that equation (13) leads to the conventional transport equation (4) with $s = 0$ for the leading amplitude A_0 , which always has this energy-conservation interpretation due to

$$\nabla \cdot (\mu A_0^2 \nabla \tau) = 0. \quad (20)$$

Unlike the true amplitudes $\{A_s\}_{s=0}^\infty$, starting values of which are hard to obtain, those “non-real” amplitudes $\{v_s\}_{s=0}^\infty$ can be well-initialized, as will be discussed below.

2.3 Computing v_s

It has been shown that in an isotropic medium, as considered here, unlike τ non-differentiable at \mathbf{r}_0 , τ^2 is analytic in the source neighborhood [30] so that by the simple relation $\nabla \tau^2 = 2\tau \nabla \tau$, we can transform equation (13) into the following transport equations with analytic coefficients in terms of τ^2 ,

$$2\mu \nabla \tau^2 \cdot \nabla v_{s+1} + v_{s+1} [2(2s + 2 - m)\rho + \nabla \cdot (\mu \nabla \tau^2)] = \nabla \cdot (\mu \nabla v_s), \quad (21)$$

for $s = -1, 0, \dots$, which, from numerical perspectives, is superior to equation (13). To compute $\{v_s\}_{s \geq 0}$, proper initial conditions of v_s at \mathbf{r}_0 must be imposed. In the following, we shall assume that the traveltime τ is found in the source neighborhood by the method of characteristics.

2.3.1 Analytic form of v_0

By ray theory or method of characteristics, along a ray traced out from the source \mathbf{r}_0 to \mathbf{r} , the directional derivative operator along the traveltime τ satisfies

$$n^2 \frac{\partial}{\partial \tau} = \nabla \tau \cdot \nabla. \quad (22)$$

Therefore, equation (21) becomes

$$4\rho\tau \frac{\partial v_{s+1}}{\partial \tau} + v_{s+1} [2(2s + 2 - m)\rho + \nabla \cdot (\mu \nabla \tau^2)] = \nabla \cdot (\mu \nabla v_s). \quad (23)$$

Clearly, when $\mu = 1$, $\rho = n^2$, and so equation (23) exactly coincides with Babich's equation (6) in [3], which is

$$4n^2\tau \frac{\partial v_{s+1}}{\partial \tau} + v_{s+1} [2(2s+2-m)n^2 + \Delta\tau^2] = \Delta v_s. \quad (24)$$

Taking $s = -1$ in equation (21), we get

$$\nabla\tau^2 \cdot \nabla v_0 + v_0 \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} - mn^2 \right] = 0. \quad (25)$$

Correspondingly, equation (23) becomes

$$\frac{\partial v_0}{\partial \tau} + v_0 \left[\frac{\nabla \cdot (\mu \nabla \tau^2) - 2m\rho}{4\tau\rho} \right] = 0, \quad (26)$$

or,

$$\frac{\partial \log v_0}{\partial \tau} = - \frac{\nabla \cdot (\mu \nabla \tau^2) - 2m\rho}{4\tau\rho}. \quad (27)$$

Thus, we obtain a general solution for v_0

$$v_0(\tau) = v_0|_{\tau=0} \exp \left(- \int_0^\tau \frac{\nabla \cdot (\mu \nabla \tau'^2) - 2m\rho}{4\tau'\rho} d\tau' \right), \quad (28)$$

which coincides with Babich's equation (7) in [3] when $\mu = 1$. Moreover, in a source neighborhood where τ^2 is analytic, v_0 is analytic and unique if the starting value $v_0|_{\tau=0}$ is specified.

We now determine $v_0(\mathbf{r}_0) = v_0|_{\tau=0}$. Near the source \mathbf{r}_0 , the wavefield u has the following asymptotic form [3]

$$u(\mathbf{r}) \simeq \begin{cases} -\frac{1}{2\pi\mu_0} \log |\mathbf{r} - \mathbf{r}_0| & m = 2, \\ \frac{\Gamma(m/2)}{(m-2)2\pi^{m/2}\mu_0} |\mathbf{r} - \mathbf{r}_0|^{2-m} & m \geq 3. \end{cases} \quad (29)$$

where $\mu_0 = \mu(\mathbf{r}_0)$. Meanwhile, knowing that $\tau(\mathbf{r}) \simeq n_0 |\mathbf{r} - \mathbf{r}_0|$ with $n_0 = n(\mathbf{r}_0)$, we may obtain from the Hankel based ansatz (5) that

$$\begin{aligned} u(\mathbf{r}) &\simeq v_0(\mathbf{r}_0) \frac{\sqrt{\pi}i}{2} \left(\frac{2\tau}{\omega} \right)^{1-m/2} H_{m/2-1}^{(1)}(\omega\tau) \\ &\simeq \begin{cases} \frac{-v_0(\mathbf{r}_0)}{\sqrt{\pi}} \log |\mathbf{r} - \mathbf{r}_0| & m = 2, \\ \frac{\Gamma(m/2)n_0^{2-m}v_0(\mathbf{r}_0)}{(m-2)\sqrt{\pi}} |\mathbf{r} - \mathbf{r}_0|^{2-m} & m \geq 3, \end{cases} \end{aligned} \quad (30)$$

where we have used the asymptotic form of the Hankel function, as $\mathbf{r} \rightarrow \mathbf{r}_0$,

$$H_{m/2-1}^{(1)} \simeq \begin{cases} \frac{2}{\pi} \log |\mathbf{r} - \mathbf{r}_0| & m = 2, \\ -\frac{\Gamma(m/2)}{(m/2-1)\pi} \left(\frac{2}{\omega\tau} \right)^{m/2-1} & m \geq 3. \end{cases} \quad (31)$$

Consequently, one gets

$$v_0(\mathbf{r}_0) = \frac{n_0^{m-2}}{2\mu_0\pi^{(m-1)/2}}, \quad (32)$$

for $m \geq 2$. Equipped with such an initial condition, equation (25) can be solved for v_0 .

2.3.2 Analytic form of v_s for $s \geq 1$

To solve equation (23) for $s \geq 0$, we first consider its homogeneous solution $v_s^{(0)}$, solving

$$4\rho\tau \frac{\partial v_{s+1}^{(0)}}{\partial \tau} + v_{s+1}^{(0)} [2(2s+2-m)\rho + \nabla \cdot (\mu \nabla \tau^2)] = 0. \quad (33)$$

It can be rewritten as by equation (27),

$$\begin{aligned} \frac{\partial \log v_{s+1}^{(0)}}{\partial \tau} &= -\frac{2(2s+2-m)\rho + \nabla \cdot (\mu \nabla \tau^2)}{4\tau\rho} \\ &= -\frac{s+1}{\tau} - \frac{\partial \log v_0}{\partial \tau} \\ &= \frac{\partial \log \tau^{-(s+1)} v_0}{\partial \tau}. \end{aligned} \quad (34)$$

Therefore, let $v_{s+1}^{(0)} = \tau^{-(s+1)} v_0$, motivating us to seek the inhomogeneous solution in the form

$$v_{s+1} = w v_{s+1}^{(0)} = w v_0 \tau^{-(s+1)},$$

where the unknown w is a function of τ . Substituting this into equation (23) and taking equation (33) into account, we see that

$$4\rho\tau^{-s} v_0 \frac{\partial w}{\partial \tau} = \nabla \cdot (\mu \nabla v_s), \quad (35)$$

leading to

$$w = \int_0^\tau \frac{\tau'^s \nabla \cdot (\mu \nabla v_s)|_{\tau'}}{4\rho v_0(\tau')} d\tau' + c_{s+1}, \quad (36)$$

and finally

$$v_{s+1}(\tau) = \tau^{-(s+1)} v_0(\tau) \left[\int_0^\tau \frac{\tau'^s \nabla \cdot (\mu \nabla v_s)|_{\tau'}}{4\rho v_0(\tau')} d\tau' + c_{s+1} \right], \quad (37)$$

where $c_{s+1} \equiv \text{const.}$ on each ray.

To determine the value of c_{s+1} , we analyze the asymptotic behavior of v_{s+1} near \mathbf{r}_0 . As $\mathbf{r} \rightarrow \mathbf{r}_0$, or $\tau \rightarrow 0$, one can easily see that

$$v_{s+1}(\mathbf{r}) \simeq \begin{cases} c_{s+1} v_0(\mathbf{r}_0) \tau^{-(s+1)} & \text{if } c_{s+1} \neq 0, \\ \frac{\nabla \cdot (\mu \nabla v_s)|_{\mathbf{r}=\mathbf{r}_0}}{4(s+1)\rho v_0} & \text{if } c_{s+1} = 0. \end{cases} \quad (38)$$

Clearly, to ensure the continuity of v_{s+1} at the source \mathbf{r}_0 , we have to choose $c_{s+1} = 0$ so that we see from equation (37) that v_{s+1} is totally determined by v_s and τ for $s \geq 0$. In other words, once v_0 and τ are known, v_s as well as its starting value $v_s(\mathbf{r}_0)$ is uniquely determined. In the source neighborhood where τ^2 is analytic, v_s is also analytic according to [3]. We point out that there appears to be an error in Babich's equation (8) in [3], to which equation (37) should reduce when $\mu = 1$ and $\rho = n^2$.

In the rest of this paper, we will mainly use the Babich ansatz truncated after the second term:

$$u(\mathbf{r}; \omega) \approx \frac{\sqrt{\pi}i}{2} \left(\frac{2\tau}{\omega}\right)^{1-m/2} \left(v_0(\mathbf{r}) H_{m/2-1}^{(1)}(\omega\tau) + v_1(\mathbf{r}) \left(\frac{2\tau}{\omega}\right) H_{m/2-2}^{(1)}(\omega\tau) \right). \quad (39)$$

Consequently, one first needs to solve the point-source eikonal equation (3) for τ , then the homogeneous transport equation (25) for v_0 , and finally the inhomogeneous transport equation (21) with $s = 0$, i.e.,

$$\nabla\tau^2 \cdot \nabla v_1 + v_1 \left[(2-m)n^2 + \frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} \right] = \frac{\nabla \cdot (\mu \nabla v_0)}{2\mu}, \quad (40)$$

for v_1 . Numerical methods for computing τ , v_0 and v_1 will be discussed later.

2.4 Local validity of the Babich ansatz

In an inhomogeneous medium with variable refractive index n , the solution τ to equation (3) is non-smooth in general, due to ray focusing away from the source \mathbf{r}_0 . As a result, caustics develop and τ becomes multi-valued. Caustics are lower dimensional manifolds enveloped by the rays. When caustics form, we usually consider the viscosity solution of equation (3) instead, which is singled out at each point as the minimum among the multiple values of τ , if they exist. However, in the vicinity of caustics, the viscosity solution is non-differentiable and the coefficients in equation (21) relating to τ become discontinuous so that the ray ansatz of Babich does not give a good approximation to the wavefield. The $\{v_s\}_{s \geq 0}$ lose analyticity at caustics. The Hankel-based series (5) is then not smooth and hence fails to characterize the wavefield at caustics, which is independently known to be smooth everywhere.

Fortunately, in isotropic media such as considered here there is a neighborhood of the source in which no caustics occur except the source itself. There is therefore a certain neighborhood of the source \mathbf{r}_0 within which the ansatz (5) yields a valid, short-wave, solution to the point-source Helmholtz equation (1). A natural question arises: can we extend the short-wave asymptotic solution beyond this region of local validity? The Huygens principle states that the wavefield away from a domain enclosing the source \mathbf{r}_0 is determined by wavefields on the boundary of the domain. Therefore, the far-wave field can be generated by accumulating the short-wave field layer by layer when we use Huygens-Kirchhoff integrals to integrate many locally valid Green's functions to construct the primary wavefield in each layer. In this way, we construct a globally valid solution, uniformly accurate both near the source and remote from it, even in the presence of caustics. Even though the primary wavefield we are constructing may contain caustics, our layer-by-layer method is successful because the contributing Green's functions do not contain caustics.

3 Huygens-principle based global solution

3.1 Huygens-Kirchhoff formula

Supposing that u is known on the boundary S of a bounded domain enclosing \mathbf{r}_0 , as shown in Figure 1(a), we develop the Huygens-Kirchhoff formula below to construct u in the exterior domain Ω_{ext} of S .

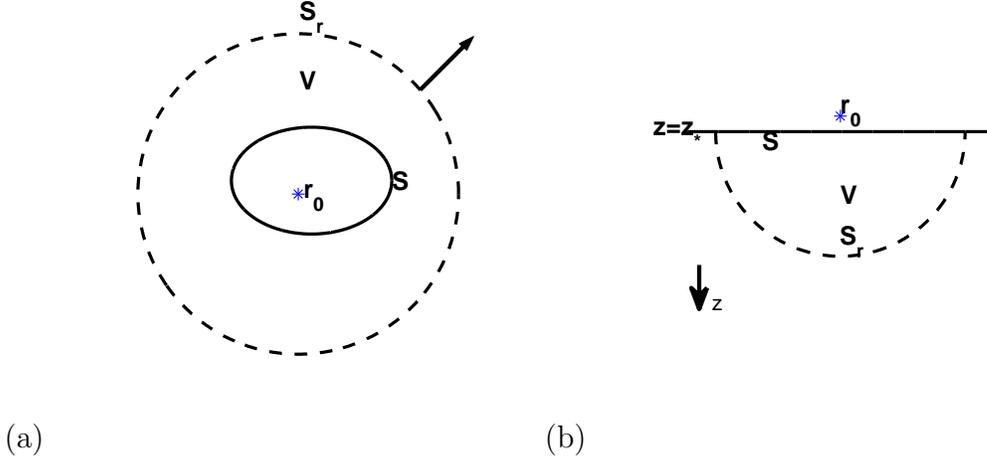


Figure 1: A 2-D view of the Huygens principle. (a) Bounded secondary source surface S ; (b) Unbounded secondary-source planar surface S .

Let $\mathbf{r}' = [x'_1, \dots, x'_m]^T$ and L be the differential operator

$$L = \nabla' \cdot (\mu(\mathbf{r}') \nabla') + \omega^2 \mu(\mathbf{r}') n^2(\mathbf{r}'),$$

where $\nabla' = [\partial_{x'_1}, \dots, \partial_{x'_m}]^T$. Then the Green's function excited by \mathbf{r} , $\mathbf{G}(\mathbf{r}'; \mathbf{r})$ satisfies

$$L\mathbf{G}(\mathbf{r}'; \mathbf{r}) = -\delta(\mathbf{r}' - \mathbf{r}), \quad (41)$$

and we have for sufficiently large $R > 0$,

$$\begin{aligned} -u(\mathbf{r}; \mathbf{r}_0) &= \int_V -\delta(\mathbf{r}' - \mathbf{r}) u(\mathbf{r}'; \mathbf{r}_0) d\mathbf{r}' \\ &= \int_V [\nabla' \cdot (\mu(\mathbf{r}') \nabla' \mathbf{G}(\mathbf{r}'; \mathbf{r})) + \omega^2 n^2(\mathbf{r}') \mu(\mathbf{r}') \mathbf{G}(\mathbf{r}'; \mathbf{r})] u(\mathbf{r}'; \mathbf{r}_0) d\mathbf{r}' \\ &= \int_V [\nabla' \cdot (\mu(\mathbf{r}') \nabla' \mathbf{G}(\mathbf{r}'; \mathbf{r})) u(\mathbf{r}'; \mathbf{r}_0) - \nabla' \cdot (\mu(\mathbf{r}') \nabla u(\mathbf{r}')) \mathbf{G}(\mathbf{r}'; \mathbf{r})] d\mathbf{r}' \\ &= \int_{S \cup \partial B(\mathbf{r}_0, R)} \mu(\mathbf{r}') [\mathbf{G}_{,\nu'}(\mathbf{r}'; \mathbf{r}) u(\mathbf{r}') - \mathbf{G}(\mathbf{r}'; \mathbf{r}) u_{,\nu'}(\mathbf{r}')] dS(\mathbf{r}'), \end{aligned} \quad (42)$$

where $V = \Omega_{ext} \cap B(\mathbf{r}_0, R)$ and ν' is the outer unit normal vector to the surface $S \cup \partial B(\mathbf{r}_0, R)$. Here and in what follows we use the subscript $_{,\nu'}$ for the directional

derivative $\boldsymbol{\nu}' \cdot \nabla'$ evaluated at \mathbf{r}' . We also define $\boldsymbol{\nu}' = \boldsymbol{\nu}(\mathbf{r}')$. Letting $R \rightarrow \infty$ and remembering the Sommerfeld radiation condition, the integral on $\partial B(\mathbf{r}_0, R)$ vanishes so that we get the Huygens-Kirchhoff formula: for $\mathbf{r} \in \Omega$,

$$u(\mathbf{r}; \mathbf{r}_0) = \int_S \mu(\mathbf{r}') [\mathbf{G}(\mathbf{r}'; \mathbf{r})u_{,\boldsymbol{\nu}'}(\mathbf{r}') - \mathbf{G}_{,\boldsymbol{\nu}'}(\mathbf{r}'; \mathbf{r})u(\mathbf{r}')] dS(\mathbf{r}'). \quad (43)$$

Similarly, one can arrive at this formula when S is an unbounded surface, e.g., a plane $z = z_*$ as shown in Figure 1(b), where we assume that u is already given in the upper half plane.

Since the operator L is self-adjoint, the Green's function satisfies the natural reciprocity:

$$\mathbf{G}(\mathbf{r}'; \mathbf{r}) = \mathbf{G}(\mathbf{r}; \mathbf{r}'). \quad (44)$$

This shows that the two arguments in \mathbf{G} and $\mathbf{G}_{,\boldsymbol{\nu}'}$ in equation (43) can be exchanged so that numerically computing the Green's function becomes more efficient since, as a manifold of source locations, the surface S has one dimension less than the volume Ω . Consequently, we have for any $\mathbf{r} \in \Omega$,

$$u(\mathbf{r}; \mathbf{r}_0) = \int_S \mu(\mathbf{r}') [\mathbf{G}(\mathbf{r}; \mathbf{r}')u_{,\boldsymbol{\nu}'}(\mathbf{r}') - \mathbf{G}_{,\boldsymbol{\nu}'}(\mathbf{r}; \mathbf{r}')u(\mathbf{r}')] dS(\mathbf{r}'). \quad (45)$$

3.2 Computing the normal derivative $\mathbf{G}_{,\boldsymbol{\nu}'}$

Using the two-term Babich approximation to approximate $\mathbf{G}(\mathbf{r}; \mathbf{r}')$, we get the approximation of its normal derivative as follows,

$$\begin{aligned} \mathbf{G}_{,\boldsymbol{\nu}'}(\mathbf{r}; \mathbf{r}') &\approx \frac{\sqrt{\pi}i}{2} (2/\omega)^{1-m/2} \left[\left(\tau^{1-m/2} v_0 H_{m/2-1}^{(1)}(\omega\tau) \right)_{,\boldsymbol{\nu}'} \right. \\ &\quad \left. - 2i/\omega \left(\tau^{2-m/2} v_1 H_{m/2-2}^{(1)}(\omega\tau) \right)_{,\boldsymbol{\nu}'} \right] \\ &= \frac{\sqrt{\pi}i}{2} (2/\omega)^{1-m/2} \left\{ \left[\omega \tau^{1-m/2} v_0 \left(H_{m/2}^{(1)} + H_{m/2-2}^{(1)} \right) / 2 \right. \right. \\ &\quad \left. \left. + (1-m/2) \tau^{-m/2} v_0 H_{1-m/2}^{(1)} - i \tau^{2-m/2} v_1 \left(H_{m/2-1}^{(1)} + H_{m/2-3}^{(1)} \right) \right] \tau_{,\boldsymbol{\nu}'} \right. \\ &\quad \left. + \tau^{1-m/2} H_{m/2-1}^{(1)} v_{0,\boldsymbol{\nu}'} + \mathcal{O}(1/\omega) \right\}, \end{aligned} \quad (46)$$

where the $\mathcal{O}(1/\omega)$ -term can be discarded at high frequencies. Clearly, computing the normal derivative requires two more derivative terms

$$\begin{aligned} \tau_{,\boldsymbol{\nu}'}(\mathbf{r}; \mathbf{r}') &:= \nabla' \tau(\mathbf{r}; \mathbf{r}') \cdot \boldsymbol{\nu}(\mathbf{r}'), \\ v_{0,\boldsymbol{\nu}'}(\mathbf{r}; \mathbf{r}') &:= \nabla' v_0(\mathbf{r}; \mathbf{r}') \cdot \boldsymbol{\nu}(\mathbf{r}'). \end{aligned}$$

3.2.1 Computing τ, ν'

We start by computing τ, ν' . It satisfies

$$\begin{aligned}\nabla\tau(\mathbf{r}; \mathbf{r}') \cdot \nabla\tau_{,\nu'}(\mathbf{r}; \mathbf{r}') &= \frac{1}{2}\nabla'(\nabla\tau(\mathbf{r}; \mathbf{r}') \cdot \nabla\tau(\mathbf{r}; \mathbf{r}')) \cdot \boldsymbol{\nu}(\mathbf{r}') \\ &= \frac{1}{2}\nabla'(n^2(\mathbf{r})) \cdot \boldsymbol{\nu}(\mathbf{r}') \\ &= 0,\end{aligned}\tag{47}$$

with a point source condition

$$\lim_{\mathbf{r} \rightarrow \mathbf{r}'} \left[\tau_{,\nu'}(\mathbf{r}; \mathbf{r}') - \frac{n(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|} (\mathbf{r}' - \mathbf{r}) \cdot \boldsymbol{\nu}(\mathbf{r}') \right] = 0,$$

obtained by taking into account that $\tau(\mathbf{r}; \mathbf{r}') \simeq n(\mathbf{r}')|\mathbf{r} - \mathbf{r}'|$ as $\mathbf{r} \rightarrow \mathbf{r}'$. Since equation (47) shows that the derivative of τ, ν' along a ray is zero, τ, ν' remains constant along the ray and is equal to its initial value

$$\tau_{,\nu'}(\mathbf{r}; \mathbf{r}') = -n(\mathbf{r}')\mathbf{t}' \cdot \boldsymbol{\nu}' = -n(\mathbf{r}') \cos \theta',\tag{48}$$

where \mathbf{t}' is the takeoff direction of the ray from \mathbf{r}' to \mathbf{r} , and θ' is the angle between \mathbf{t}' and $\boldsymbol{\nu}'$.

To resolve the singularity of τ, ν' at $\mathbf{r} = \mathbf{r}'$, we compute $\Phi = \tau\tau_{,\nu'}$ instead, solving

$$\nabla\tau^2 \cdot \nabla\Phi = 2(\nabla\tau \cdot \nabla\tau)\tau\tau_{,\nu'} + \tau^2\nabla\tau \cdot \nabla\tau_{,\nu'} = 2n^2\Phi, \quad \text{where } \Phi = \tau\tau_{,\nu'},\tag{49}$$

with a point source condition

$$\lim_{\mathbf{r} \rightarrow \mathbf{r}'} [\Phi(\mathbf{r}; \mathbf{r}') - n^2(\mathbf{r}')(\mathbf{r}' - \mathbf{r}) \cdot \boldsymbol{\nu}(\mathbf{r}')] = 0.\tag{50}$$

3.2.2 Computing $v_{0,\nu'}$

Let us now turn to the consideration of $v_{0,\nu'}$. By equation (25), we have

$$\begin{aligned}\nabla\tau^2 \cdot \nabla v_{0,\nu'}(\mathbf{r}; \mathbf{r}') &= \nabla\tau^2(\mathbf{r}; \mathbf{r}') \cdot \nabla(\nabla'v_0(\mathbf{r}; \mathbf{r}') \cdot \boldsymbol{\nu}'(\mathbf{r}')) \\ &= (\tau^2 \cdot \nabla v_0)_{,\nu'} - \nabla((\tau^2)_{,\nu'}) \cdot \nabla v_0 \\ &= - \left(v_0 \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} - mn^2 \right] \right)_{,\nu'} - 2\nabla\Phi \cdot \nabla v_0 \\ &= - \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} - mn^2 \right] v_{0,\nu'} - v_0 \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} \right]_{,\nu'} - 2\nabla\Phi \cdot \nabla v_0 \\ &= - \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} - mn^2 \right] v_{0,\nu'} - v_0 \left[\frac{\nabla \cdot (\mu \nabla \tau_{,\nu'}^2)}{2\mu} \right] - 2\nabla\Phi \cdot \nabla v_0 \\ &= - \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} - mn^2 \right] v_{0,\nu'} - v_0 \left[\frac{\nabla \cdot (\mu \nabla \Phi)}{\mu} \right] - 2\nabla\Phi \cdot \nabla v_0,\end{aligned}\tag{51}$$

thereby leading to the governing equation for $v_{0,\nu'}$:

$$\nabla\tau^2 \cdot \nabla v_{0,\nu'} + v_{0,\nu'} \left[\frac{\nabla \cdot (\mu \nabla \tau^2)}{2\mu} - mn^2 \right] = -v_0 \frac{\nabla \cdot (\mu \nabla \Phi)}{2\mu} - 2\nabla\Phi \cdot \nabla v_0, \quad (52)$$

with a point source condition

$$\lim_{\mathbf{r} \rightarrow \mathbf{r}'} v_{0,\nu'}(\mathbf{r}; \mathbf{r}') = v_{0,\nu'}(\mathbf{r}; \mathbf{r}')|_{\mathbf{r}=\mathbf{r}'}, \quad (53)$$

to be determined by v_0 .

To use the Huygens-Kirchhoff formula (45) for computing u in Ω_{ext} , we need to construct $\mathbf{G}(\mathbf{r}; \mathbf{r}')$ and $\mathbf{G}_{,\nu'}(\mathbf{r}; \mathbf{r}')$ by computing the following five ingredients sequentially: $\tau^2(\mathbf{r}; \mathbf{r}')$ (not τ as it is not smooth at the source), $v_0(\mathbf{r}; \mathbf{r}')$, $v_1(\mathbf{r}; \mathbf{r}')$, $\Phi(\mathbf{r}; \mathbf{r}')$ and $v_{0,\nu'}(\mathbf{r}; \mathbf{r}')$ at any $\mathbf{r} \in \Omega_{ext}$ and $\mathbf{r}' \in S$, which will be referred in what follows as the **Babich ingredients**.

3.3 Layer-by-layer sweeping: continuous case

As discussed before, the Babich ansatz (5) is valid locally so that computed \mathbf{G} and $\mathbf{G}_{,\nu'}$ by equations (39) and (46), respectively, are only accurate in a short-wave distance away from the source surface S so that we may only construct the wavefield in a layer within the short-wave distance away from S by the Huygens-Kirchhoff formula (45). Nevertheless, by repeating such process, one still can construct a globally valid solution layer by layer. We now present the layer-by-layer sweeping algorithm.

Algorithm 1. *Layer-by-layer Sweeping Algorithm.*

1. For the primary source \mathbf{r}_0 , identify the first layer Ω_0 as a region enclosing \mathbf{r}_0 , where Babich approximation (39) is valid, and compute the wavefield u by equation (39) in Ω_0 .
2. In the exterior of Ω_0 , identify the second layer Ω_1 as a common neighborhood where the Babich approximation (39) remains valid for the Green's function excited by any secondary source point on the boundary of Ω_0 , $S_0 = \partial\Omega_0$, named by the first secondary source surface. Consequently, the Green's function as well as its normal derivative become available by the Babich approximations (39) and (46).
3. Compute u in Ω_1 by the Huygens-Kirchhoff formula (45) with $S = S_0$ and $\Omega_{ext} = \Omega_1$.
4. Repeating steps 2 and 3, we get a sequence of layers $\{\Omega_j\}_{j \geq 1}$ so that the Green's function and its normal derivative at the secondary source surface $S_{j-1} = \partial\Omega_{j-1}$ are available in Ω_j . Repeatedly using the Huygens-Kirchhoff formula in all layers $\{\Omega_j\}$, we sweep the wavefield u layer-by-layer so that a globally valid solution is constructed.

In the following, we will present details on implementing **Algorithm 1** and will only focus on the two- or three- dimensional space, i.e., $m = 2$ or 3 . For the sake of clarity, we will name the Cartesian axes in \mathbb{R}^2 by x and z , in \mathbb{R}^3 by x , y and z .

4 Lax-Fredrichs WENO schemes for the Babich ingredients

4.1 Numerical schemes

To compute the wavefield u by the primary source \mathbf{r}_0 or the Green's function \mathbf{G} by a secondary source, we have to solve the eikonal equation (3) first as τ appears as coefficients in the governing equations (25) and (40). In general, if we desire a 1st-order accurate v_1 , we need a 3rd-order accurate v_0 due to the term $\frac{\nabla \cdot (\mu \nabla v_0)}{\mu}$ in equation (40). This in turn indicates that we have to compute a 5th-order accurate τ .

Since τ itself is non-differentiable at \mathbf{r}_0 , all high-order upwind numerical schemes on discretizing the point-source eikonal equation (3) itself yields only 1st-order accuracy near the source, which further propagates over the whole computational domain due to the upwinding nature of the underlying schemes. To resolve this issue, we follow the factorization approach in [38, 57, 17, 27, 31, 28, 30], rewriting τ as the product of a known non-differentiable term (e.g. $n_0|\mathbf{r} - \mathbf{r}_0|$) and an unknown which is analytic in a source neighborhood. Next, by employing the existing high-order Lax-Fredrichs Weighted Essentially Non-Oscillatory (LxF-WENO) schemes [21, 58, 27, 30, 31, 45] to compute the unknown, we obtain a solution converging to the viscosity solution of equation (3) with high accuracy in the smooth region of τ^2 . Such high-order LxF-WENO schemes can be easily adapted to solve the other first-order hyperbolic equations, such as equations (25), (40), (49) and (52).

In this paper, we use the 5th-, 3rd- and 1st-order LxF-WENO schemes developed in [45] to solve equations (3), (25) and (40), respectively, yielding a 5th-order accurate τ , a 3rd-order accurate v_0 and a 1st-order accurate v_1 . Consequently, the primary wavefield or the Green's function excited by a secondary source can be constructed. In addition, we use the 3rd-order and 1st-order LxF-WENO schemes in [45] to compute a 3rd-order accurate Φ and a 1st-order accurate $v_{0,\nu'}$, respectively, so that the normal-derivative term $\mathbf{G}_{,\nu'}$ can be constructed.

In the implementation, to use a P -th order LxF-WENO scheme to solve any of the five point-source equations (3), (25), (40), (49), and (52), one needs to initialize a P -th order accurate solution in a neighborhood of size $(P+1)h$ centered at the source where h is the grid size used in LxF-WENO schemes. During each iteration, values of the targeted variable in this neighborhood are fixed while those elsewhere are updated by the related scheme until convergence is achieved. In the next section, following closely [45], we present details on initializing the five Babich ingredients near the source.

4.2 Initialization of the Babich ingredients

Without loss of generality, we consider initializing the five Babich ingredients near the primary source \mathbf{r}_0 and taking $\mathbf{r}_0 = \mathbf{0}$. Define $\Psi(\mathbf{r}) = n^2(\mathbf{r})$ and $U(\mathbf{r}) = \log \mu(\mathbf{r})$. As they are analytic near the source, we can rewrite them by their Taylor series about the origin as:

$$\Psi(\mathbf{r}) = \sum_{k=0}^{\infty} \Psi_k(\mathbf{r}), \quad (54)$$

$$U(\mathbf{r}) = \sum_{k=0}^{\infty} U_k(\mathbf{r}), \quad (55)$$

where and hereafter in this section the term with subscript k denotes a homogeneous polynomial of degree k .

As τ^2 is analytic in the source neighborhood, we denote its Taylor series by

$$\tau^2(\mathbf{r}) = \sum_{k=0}^{\infty} T_k(\mathbf{r}). \quad (56)$$

According to [30], by substituting equations (54) and (56) into the squared eikonal equation, we obtain recursive formulas for $\{T_k\}$ when $k \geq 3$:

$$(P-1)\Psi_0 T_P = \sum_{k=1}^{P-2} \Psi_k T_{P-k} - \frac{1}{4} \sum_{k=2}^{P-2} \nabla T_{k+1} \cdot \nabla T_{P-k+1}, \quad (57)$$

together with initial setups: $T_0 = T_1 = 0$ and $T_2 = \Psi_0 \mathbf{r}^2$. For example, by equation (57), one gets

$$T_3(\mathbf{r}) = \frac{1}{2} \Psi_1(\mathbf{r}) \mathbf{r}^2. \quad (58)$$

Therefore, if we use the truncated series $\tilde{T}_{\tilde{P}}(\mathbf{r}) = \sum_{k=2}^{\tilde{P}} T_k(\mathbf{r})$ to approximate τ^2 near the source, we obtain

$$|\tau(\mathbf{r}) - \sqrt{|\tilde{T}_{\tilde{P}}(\mathbf{r})|}| = \mathcal{O}(|\mathbf{r}|^{\tilde{P}+1}), \quad (59)$$

as $|\mathbf{r}| \rightarrow 0^+$. In our implementation, we take $\tilde{P} = 6$ to initialize τ .

Next, to initialize v_0 , we first rewrite it as a Taylor series at the origin:

$$v_0(\mathbf{r}) = \sum_{k=0}^{\infty} B_k(\mathbf{r}), \quad (60)$$

with

$$B_0 = \frac{n_0^{m-2}}{2\pi^{(m-1)/2} \mu_0},$$

according to the initial condition (32) for v_0 . Thus, equation (25) can be rewritten as

$$\begin{aligned} & \sum_{k=2}^{\infty} \nabla T_k(\mathbf{r}) \cdot \sum_{k=1}^{\infty} \nabla B_k(\mathbf{r}) + \sum_{k=0}^{\infty} B_k(\mathbf{r}) \left[\frac{1}{2} \sum_{k=2}^{\infty} \Delta T_k(\mathbf{r}) \right. \\ & \left. + \frac{1}{2} \sum_{k=1}^{\infty} \nabla U_k(\mathbf{r}) \sum_{k=2}^{\infty} \nabla T_k(\mathbf{r}) - m \sum_{k=0}^{\infty} \Psi_k(\mathbf{r}) \right] = 0. \end{aligned} \quad (61)$$

By comparing the linear terms of both sides, we obtain

$$2\Psi_0 B_1 + B_0 \left[\frac{\Delta T_3}{2} + \frac{1}{2} \nabla U_1 \cdot \nabla T_2 - m \Psi_1 \right] = 0, \quad (62)$$

solved by

$$\begin{aligned} B_1 &= -\frac{B_0}{2\Psi_0} \left(\frac{\Delta T_3}{2} + \frac{1}{2} \nabla U_1 \cdot \nabla T_2 - m \Psi_1 \right) \\ &= -\frac{B_0}{2\Psi_0} \left[\left(1 - \frac{m}{2}\right) \Psi_1 + \Psi_0 U_1 \right]. \end{aligned} \quad (63)$$

By comparing the P -th degree terms of both sides of equation (25), we derive the following recursive formulas for $\{B_k\}_{k \geq 2}$:

$$\begin{aligned} 2\Psi_0 P B_P &= -\sum_{k=1}^{P-1} \nabla B_k \cdot \nabla T_{P+2-k} - \frac{1}{2} \sum_{k=0}^{P-1} B_k \Delta T_{P+2-k} \\ &+ m \sum_{k=0}^{P-1} B_k \Psi_{P-k} - \frac{1}{2} \sum_{k=0}^{P-1} B_k \left(\sum_{j=1}^{P-k} \nabla U_j \cdot \nabla T_{P+2-j-k} \right). \end{aligned} \quad (64)$$

Therefore, we may use the truncated series $\tilde{B}_{\tilde{P}}(\mathbf{r}) = \sum_{k=0}^{\tilde{P}} B_k(\mathbf{r})$ to approximate v_0 near the source, and we get

$$|v_0(\mathbf{r}) - \tilde{B}_{\tilde{P}}(\mathbf{r})| = \mathcal{O}(|\mathbf{r}|^{\tilde{P}+1}), \quad (65)$$

as $|\mathbf{r}| \rightarrow 0^+$. Here, we take $\tilde{P} = 3$ to initialize v_0 .

Similarly, to initialize v_1 , we first rewrite it as its Taylor series at \mathbf{r}_0 :

$$v_1(\mathbf{r}) = \sum_{k=0}^{\infty} C_k(\mathbf{r}). \quad (66)$$

Then, by comparing the P -th degree terms in equation (40), we obtain the recursive formulas for $\{C_k\}_{k \geq 0}$:

$$\begin{aligned} 2\Psi_0(P+1)C_P &= \frac{1}{2} \sum_{k=1}^{P+1} \nabla B_k \cdot \nabla U_{P+2-k} + \frac{1}{2} \Delta B_{P+2} - \sum_{k=1}^{P-1} \nabla C_k \cdot \nabla T_{P+2-k} \\ &- \sum_{k=0}^{P-1} C_k \left(\frac{\Delta T_{P+2-k}}{2} + \frac{1}{2} \sum_{j=1}^{P-k} \nabla U_j \cdot \nabla T_{P+2-k-j} + (2-m)\Psi_{P-k} \right). \end{aligned} \quad (67)$$

Particularly, when $P = 0$, we have

$$2\Psi_0 C_0 = \frac{1}{2} (\nabla U_1 \cdot \nabla B_1 + \Delta B_2), \quad (68)$$

which further confirms that the starting value of v_1 is determined by v_0 .

Therefore, we may use the truncated series $\tilde{C}_{\tilde{P}}(\mathbf{r}) = \sum_{k=0}^{\tilde{P}} C_k(\mathbf{r})$ to approximate v_1 near the source, and we get

$$|v_1(\mathbf{r}) - \tilde{C}_{\tilde{P}}(\mathbf{r})| = \mathcal{O}(|\mathbf{r}|^{\tilde{P}+1}), \quad (69)$$

as $|\mathbf{r}| \rightarrow 0^+$. Here, we take $\tilde{P} = 2$ to initialize v_1 .

Finally, we deal with initializing $\Phi(\mathbf{r}; \mathbf{r}')$ and $v_{0,\nu'}(\mathbf{r}; \mathbf{r}')$ near the source $\mathbf{r}' = \mathbf{0}$. Let the power series of Φ at the origin be:

$$\Phi(\mathbf{r}) = \sum_{k=0}^{\infty} \Phi_k(\mathbf{r}). \quad (70)$$

Then, by the point-source condition (50), we get the initial setup for Φ :

$$\Phi_0 = 0, \quad \Phi_1(\mathbf{r}) = (n^2(\mathbf{r}')(\mathbf{r}' - \mathbf{r}) \cdot \boldsymbol{\nu}(\mathbf{r}'))|_{\mathbf{r}'=\mathbf{0}} = -\Psi_0 \mathbf{r} \cdot \boldsymbol{\nu}_0, \quad (71)$$

where $\boldsymbol{\nu}_0$ is the unit normal vector specified at the source. By comparing the P -th degree terms at both sides of equation (49), we obtain the following recursive formulas for $\{\Phi_k\}_{k \geq 2}$:

$$2\Psi_0(P-1)\Phi_P = 2 \sum_{k=1}^{P-1} \Phi_k \Psi_{P-k} - \sum_{k=1}^{P-1} \nabla T_{P-k+2} \cdot \nabla \Phi_k.$$

Therefore, we use the truncated series $\tilde{\Phi}_{\tilde{P}}(\mathbf{r}) = \sum_{k=0}^{\tilde{P}} \Phi_k(\mathbf{r})$ to approximate Φ near the source, and we get

$$|\Phi(\mathbf{r}) - \tilde{\Phi}_{\tilde{P}}(\mathbf{r})| = \mathcal{O}(|\mathbf{r}|^{\tilde{P}+1}) \quad (72)$$

as $|\mathbf{r}| \rightarrow 0^+$. Here, we take $\tilde{P} = 4$ to initialize Φ .

On the other hand, assuming the power series of $v_{0,\nu'}$ at the origin to be

$$v_{0,\nu'}(\mathbf{r}) = \sum_{k=0}^{\infty} W_k(\mathbf{r}), \quad (73)$$

by the point-source conditions (53) and (63), we obtain the initial setup for the sequences $\{W_k\}$:

$$\begin{aligned} W_0 &= (\nabla' v_0(\mathbf{r}; \mathbf{r}') \cdot \boldsymbol{\nu}(\mathbf{r}'))|_{\mathbf{r}=\mathbf{r}'=\mathbf{0}} \\ &= (\nabla' B_1(\mathbf{r}; \mathbf{r}') \cdot \boldsymbol{\nu}(\mathbf{r}'))|_{\mathbf{r}=\mathbf{r}'=\mathbf{0}} \\ &= (-\nabla B_1(\mathbf{r}; \mathbf{r}') \cdot \boldsymbol{\nu}(\mathbf{r}'))|_{\mathbf{r}=\mathbf{r}'=\mathbf{0}} \\ &= \frac{B_0}{2\Psi_0} [(1 - m/2)\nabla \Psi_1 + \Psi_0 \nabla U_1] \cdot \boldsymbol{\nu}_0. \end{aligned} \quad (74)$$

By comparing the P -th degree terms at both sides of equation (52), we obtain the following recursive formulas for $\{W_k\}_{k \geq 1}$:

$$\begin{aligned}
2\Psi_0 P W_P = & - \sum_{k=1}^{P-1} \nabla W_k \cdot \nabla T_{P+2-k} - \frac{1}{2} \sum_{k=0}^{P-1} W_k \Delta T_{P+2-k} \\
& + m \sum_{k=0}^{P-1} W_k \Psi_{P-k} - \frac{1}{2} \sum_{k=0}^{P-1} W_k \left(\sum_{j=1}^{P-k} \nabla U_j \cdot \nabla T_{P+2-j-k} \right) \\
& - \sum_{k=0}^P B_k \left[\Delta \Phi_{P+2-k} + \sum_{j=1}^{P+1-k} \nabla U_j \cdot \nabla \Phi_{P+2-k-j} \right] \\
& - 2 \sum_{k=1}^{P+1} \nabla B_k \cdot \nabla \Phi_{P+2-k}.
\end{aligned} \tag{75}$$

Therefore, we may use the truncated series $\tilde{W}_{\tilde{P}}(\mathbf{r}) = \sum_{k=0}^{\tilde{P}} W_k(\mathbf{r})$ to approximate $v_{0,\nu'}$ near the source, and we get

$$|v_{0,\nu'}(\mathbf{r}) - \tilde{W}_{\tilde{P}}(\mathbf{r})| = \mathcal{O}(|\mathbf{r}|^{\tilde{P}+1}), \tag{76}$$

as $|\mathbf{r}| \rightarrow 0^+$. Here, we take $\tilde{P} = 2$ to initialize $v_{0,\nu'}$.

5 Layer-by-layer sweeping: discretized case

We now develop the sweeping algorithm in the discretized case. In general, the sweeping algorithm can be implemented in two stages: the offline stage and the online stage. The offline stage mainly deals with decomposing the computational domain into layers and computing the Babich ingredients at primary and secondary sources, while the online stage focuses on updating the wavefield in each layer by the Huygens-Kirchhoff formula.

Suppose now the wavefield propagates mainly along the positive z -direction so that we can choose those layers to be m -dimensional boxes perpendicular to the z -axis and extended to infinity along the other axes. The secondary-source surfaces in turn become $(m-1)$ -dimensional planar surfaces separating those layers. If u is computed in a bounded m -dimensional box Ω enclosing \mathbf{r}_0 , those layers and secondary-source surfaces are automatically truncated to bounded boxes and planar surfaces, respectively.

5.1 Offline stage: domain decomposition and computing the Babich ingredients

To compute the wavefield in Ω enclosing the source \mathbf{r}_0 , according to **Algorithm 1**, we first need to decompose Ω into layers $\{\Omega_j\}_{j=0}^{N_{lay}}$ separated by the secondary-source surfaces $\{S_j\}_{j=0}^{N_{lay}-1}$, where N_{lay} is the number of layers in Ω .

At first, we numerically compute the viscosity solution of τ at the primary source \mathbf{r}_0 in Ω , and then identify a box neighborhood enclosing \mathbf{r}_0 , where τ^2 is smooth, as the first layer Ω_0 . Next, rather than taking the boundary of Ω_0 to be the first secondary-source surface S_0 , we move this $(m - 1)$ -dimensional planar surface slightly close to \mathbf{r}_0 so that the new secondary source surface S_0 is well separated from the next layer Ω_1 in the exterior of Ω_0 . This guarantees that the Green's function in equation (45) is never singular. Now, we sample secondary-source points on S_0 , and identify a box neighborhood exterior of Ω_0 as the next layer Ω_1 so that τ^2 at any sampling point of S_0 in Ω_1 is smooth. Repeating such process, we obtain a sequence of layers $\{\Omega_j\}_{j=0}^{N_{lay}}$ and secondary-source surfaces $\{S_j\}_{j=0}^{N_{lay}-1}$. Finally, we compute the Babich ingredients at the primary source \mathbf{r}_0 in Ω_0 (of course, only τ^2 , v_0 and v_1 are enough for \mathbf{r}_0) and at all sampling secondary sources of each S_{j-1} in Ω_j for $j = 1, \dots, N_{lay}$ by the aforementioned high-order LxF-WENO schemes. Those tables of ingredients enable us to construct the Green's function \mathbf{G} and its normal derivative $\mathbf{G}_{,\nu'}$ in each layer, and hence the wavefield u can be constructed by the Huygens-Kirchhoff formula (45), as will be detailed at the online stage in **section 5.2**.

In practice, we let the distance between each pair of Ω_j and S_{j-1} , denoted by $d_f > 0$, be a constant and independent of ω . In addition, since all the Babich ingredients are independent of the frequency ω , the computational domain Ω can be discretized by a very coarse mesh, independent of ω . Correspondingly, the layer Ω_j is discretized by mesh points restricted in Ω_j , and sampling secondary sources on S_{j-1} are directly chosen as mesh points restricted on S_{j-1} . Therefore, the Babich ingredients can be computed on the restricted coarse mesh points in each Ω_j at the restricted coarse source points on S_{j-1} .

5.1.1 Data compression

At the offline stage, once tables of all Babich ingredients are obtained, we store them on a local hard disk, which can be reloaded afterwards. However, directly storing them requires a great deal of storage, which is more pronounced in the 3-D case. To reduce data storage, utilizing the fact that all Babich ingredients in each layer are smooth, we follow the approach in [29] to compress each data table into a linear combination of tensor-product based multivariate Chebyshev polynomials so that the information in each table can be encoded into a small number of coefficients. In our setting, there are five scalar tables (except at the primary source \mathbf{r}_0) in total to be compressed, including τ^2 , v_0 , v_1 , Φ and $v_{0,\nu'}$.

To expedite reconstruction of information from those compressed tables, we will follow the low-rank matrices based approach in [29] which is equivalent to the Orszag partial summation method [9].

5.2 Online stage: wavefields by Huygens-Kirchhoff summation

Computationally, to apply the Huygens-Kirchhoff formula (45) when S is an unbounded plane, one needs to truncate the unbounded integration plane S into a bounded domain \tilde{S} . Since Babich's expansion (5) automatically exhibits outgoing wave behavior and satisfies the Sommerfeld radiation condition at infinity, such a truncation only affects the accuracy of the wavefield near the boundary of \tilde{S} . Therefore, the Huygens-Kirchhoff formula (45) with S replaced by any bounded secondary planar surface S_j can be directly applied to update the wavefield in the layer Ω_{j+1} for $j = 0, \dots, N_{lay} - 1$. The integral over S_j is further discretized so that a quadrature rule can be applied to approximate the integral.

In the high-frequency regime, to capture the highly oscillatory behavior of the wavefield, one has to specify sufficient mesh points to sample the overall solution. In principle, the optimal number of mesh points is arguably 4 to 6 points per wavelength in all m dimensions. While in the popular finite difference time domain (FDTD) method or the finite element method (FEM), it is hard to obtain accurate solutions by using such a "small" number of mesh points due to the dispersion or pollution error [5, 4], our method based on the Babich formula explicitly splits out the frequency independent terms from the wavefield, i.e., the Babich ingredients, so that only 4 to 6 points per wavelength can still yield accurate solutions, which has been verified numerically in [29, 41, 40] and will later be shown in our examples.

For a given refractive index $n(\mathbf{r})$, the minimum wavelength in the computational domain Ω is $\lambda_{\min} = 2\pi/(\omega n_{\max})$, where n_{\max} is the maximum refractive index in Ω , so that Ω can be uniformly discretized by an optimal number of points with 4 to 6 points per minimum wavelength. However, since all Babich ingredients are independent of the frequency, one can compute them on much coarser meshes with the number of points independent of frequency rather than on this very fine mesh. Only when we construct the wavefield in Ω do we need to interpolate those ingredients onto the fine mesh.

With the above statements under consideration, we are ready to discretize the Huygens-Kirchhoff formula (45) with $S = S_j$ and $\Omega_{ext} = \Omega_{j+1}$ for $j = 0, \dots, N_{lay} - 1$. Without loss of generality, we only consider the case when $j = 0$. Suppose S_0 is uniformly discretized by M points $\{\mathbf{s}_j\}_{j=1}^M$ with a grid size h in each dimension while Ω_1 is uniformly discretized by $\{\mathbf{r}_i\}_{i=1}^N$ with the same grid size h . Then, by the trapezoidal rule, equation (45) at $\mathbf{r} = \mathbf{r}_i$ becomes

$$u(\mathbf{r}_i) \approx h^{m-1} \sum_{j=1}^M [\mathbf{G}(\mathbf{r}_i, \mathbf{s}_j) u_{,\nu'}(\mathbf{s}_j) \mu(\mathbf{s}_j) - \mathbf{G}_{,\nu'}(\mathbf{r}_i; \mathbf{s}_j) u(\mathbf{s}_j) \mu(\mathbf{s}_j)], \quad (77)$$

for $i = 1, \dots, M$. In matrix form, it becomes

$$\mathbf{u} = \mathbf{U}_1 \mathbf{f}_1 + \mathbf{U}_2 \mathbf{f}_2, \quad (78)$$

where the $N \times 1$ vector \mathbf{u} , the $M \times 1$ vectors \mathbf{f}_1 and \mathbf{f}_2 , and the $N \times M$ matrices \mathbf{U}_1

and \mathbf{U}_2 are defined in the following,

$$\mathbf{u} = [u(\mathbf{r}_1), \dots, u(\mathbf{r}_N)]^T, \quad (79)$$

$$\mathbf{f}_1 = h^{m-1} [u_{,\nu'}(\mathbf{s}_1)\mu(\mathbf{s}_1), \dots, u_{,\nu'}(\mathbf{s}_M)\mu(\mathbf{s}_M)]^T, \quad (80)$$

$$\mathbf{f}_2 = -h^{m-1} [u(\mathbf{s}_1)\mu(\mathbf{s}_1), \dots, u(\mathbf{s}_M)\mu(\mathbf{s}_M)]^T, \quad (81)$$

$$\mathbf{U}_1 = [(\mathbf{U}_1)_{ij}] = [\mathbf{G}(\mathbf{r}_i, \mathbf{s}_j)]_{1 \leq i \leq N, 1 \leq j \leq M}, \quad (82)$$

$$\mathbf{U}_2 = [(\mathbf{U}_2)_{ij}] = [\mathbf{G}_{,\nu'}(\mathbf{r}_i; \mathbf{s}_j)]_{1 \leq i \leq N, 1 \leq j \leq M}. \quad (83)$$

To ensure that equation (78) attains a specified accuracy and captures the highly oscillatory behavior of the wavefield in Ω_1 , we directly take the mesh points in Ω restricted to S_0 as the source points $\{\mathbf{s}_j\}_{j=1}^M$ and to Ω_1 as the receiver points $\{\mathbf{r}_i\}_{i=1}^N$. As the Babich ingredients at \mathbf{r}_0 are obtained in Ω_0 , one first interpolates them onto the restricted dense mesh points in Ω_0 so that by equation (39), the wavefield u at mesh points in Ω_0 , including $\{\mathbf{s}_j\}_{j=1}^M$, can be constructed. In turn, its normal derivative $u_{,\nu'}$ at $\{\mathbf{s}_j\}_{j=1}^M$ can be obtained by numerical differentiation. On the other hand, as the tables of the Babich ingredients at coarsely discretized mesh points in layer Ω_1 are only computed for coarsely sampled sources on S_0 , to construct \mathbf{U}_1 and \mathbf{U}_2 , we first interpolate each table onto the dense mesh points $\{\mathbf{r}_i\}_{i=1}^N$ in Ω_1 , and then interpolate those refined tables onto new refined tables at densely sampled sources $\{\mathbf{s}_j\}_{j=1}^M$ on S_0 since Babich ingredients are also continuous functions of source locations. Consequently, by equation (78), one obtains the wavefield u at the restricted mesh points $\{\mathbf{r}_i\}_{i=1}^N$ in Ω_1 so that u and $u_{,\nu'}$ on the next secondary source surface S_1 become available. Repeating such process, one constructs u in the whole computational domain Ω .

However, a significant barrier in such a formulation is that at high frequencies, if the number of mesh points in each direction is $J = \mathcal{O}(\omega)$, then $M = \mathcal{O}(J^{m-1})$, $N = \mathcal{O}(J^m)$, and both the complexity and storage requirements in obtaining \mathbf{u} by direct matrix-vector multiplications in equation (78) become $\mathcal{O}(MN) = \mathcal{O}(J^{2m-1})$ which is extremely expensive and impractical at high frequencies. To resolve this issue, we adopt a multilevel matrix decomposition based butterfly algorithm [11, 14, 29, 41, 40] to speed up the multiplications.

5.3 A butterfly algorithm

Equation (78) involves two matrix-vector multiplications that can be reformulated as computing

$$u(\mathbf{r}) = \sum_{k=1}^2 \sum_{\mathbf{s} \in \mathbf{X}_s} U_k(\mathbf{r}; \mathbf{s}) f_k(\mathbf{s}), \quad \mathbf{r} \in \mathbf{X}_r \subset \Omega_R. \quad (84)$$

Here, \mathbf{X}_S and \mathbf{X}_R are input source points in the source domain Ω_S and output receiver points in the receiver domain Ω_R , respectively, where Ω_R and Ω_S are d_f apart from each other, $f_k(\mathbf{s})$ is the representative function of \mathbf{f}_k in the sense that $f_k(\mathbf{s}_j)$ is the

j -th element of \mathbf{f}_k for $1 \leq j \leq M$ and $U_k(\mathbf{r}; \mathbf{s})$, the representative function of \mathbf{U}_k , can be decomposed as

$$U_k(\mathbf{r}; \mathbf{s}) = A_k(\mathbf{r}; \mathbf{s})e^{i\omega\tau(\mathbf{r}; \mathbf{s})}, \quad (85)$$

where according to equations (39) and (46),

$$A_1 = \frac{\sqrt{\pi}i}{2} \left(\frac{2\tau}{\omega}\right)^{1-m/2} \left(v_0 H_{m/2-1}^{(1)}(\omega\tau) + v_1 \left(\frac{2\tau}{\omega}\right) H_{m/2-2}^{(1)}(\omega\tau) \right) e^{-i\omega\tau}, \quad (86)$$

$$\begin{aligned} A_2 = & \frac{\sqrt{\pi}i}{2} (2/\omega)^{1-m/2} \left\{ \left[\omega\tau^{1-m/2} v_0 \left(H_{m/2}^{(1)}(\omega\tau) + H_{m/2-2}^{(1)}(\omega\tau) \right) / 2 \right. \right. \\ & + (1-m/2)\tau^{-m/2} v_0 H_{m/2-1}^{(1)}(\omega\tau) \\ & \left. \left. - i\tau^{2-m/2} v_1 \left(H_{m/2-1}^{(1)}(\omega\tau) + H_{m/2-3}^{(1)}(\omega\tau) \right) \right] \tau, \nu' \right. \\ & \left. + \tau^{1-m/2} H_{m/2-1}^{(1)}(\omega\tau) v_{0, \nu'} \right\} e^{-i\omega\tau}, \end{aligned} \quad (87)$$

and they together with the traveltime τ are available for $\mathbf{r} \in \Omega_R$ and $\mathbf{s} \in \Omega_S$. For large τ , due to the asymptotic form of the Hankel function

$$H_k^{(1)}(\omega\tau) \simeq \sqrt{\frac{2}{\pi\omega\tau}} e^{-i(k\pi/2 - \pi/4)} e^{i\omega\tau}, \quad (88)$$

for $k \in \{m/2 - 3, m/2 - 2, m/2 - 1, m/2\}$, the two amplitudes A_1 and A_2 , though dependent upon ω , are not oscillatory at high frequencies. Based on such decomposition (85), we can adopt the low-rank separation based butterfly algorithm [11, 14, 29, 40] to speed up the matrix-vector products in equation (78). Some modifications can be made so as to make it more adaptive to our applications.

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 with nodes X at $x \in [-1, 1]$ by $L_X(x; x_j)$, taking 1 at x_j and 0 elsewhere in X , for $j = 1, \dots, p$. On the standard m -dimensional box $[-1, 1]^m$, the Chebyshev nodes of order p are m tensor products of X as:

$$X^m = \{x_{j_1}\}_{j_1=1}^p \times \dots \times \{x_{j_m}\}_{j_m=1}^p.$$

Thus, the $\mathbf{j} = (j_1, \dots, j_m)$ -th Lagrange basis function with nodes X^m at \mathbf{x} satisfies:

$$L_{X^m}(\mathbf{x}; \mathbf{x}_j) = L_X(x^1; x_{j_1}) \cdots L_X(x^m; x_{j_m}),$$

where $\mathbf{x} = (x^1, \dots, x^m)^T \in [-1, 1]^m$ and $\mathbf{x}_j = (x_{j_1}, \dots, x_{j_m})^T$.

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

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

with the j -th Lagrange basis function at $y \in [a, b]$ denoted by $L_Y(y; y_j)$. Meanwhile, on a general m -dimensional box $[a_1, b_1] \times \cdots \times [a_m, b_m]$, the Chebyshev nodes of order p become the following tensor products:

$$\mathbf{Y} = Y_1 \times \cdots \times Y_m,$$

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 j -th Lagrange basis function at $\mathbf{y} = (y^1, \cdots, y^m)$ is

$$L_{\mathbf{Y}}(\mathbf{y}; \mathbf{y}_j) = L_{Y_1}(y^1; y_{j_1}) \cdots L_{Y_m}(y^m; y_{j_m}).$$

In the following, we will denote by \mathbf{C}^B the set of p^m m -dimensional Chebyshev nodes in an m -dimensional box B . Following closely the butterfly algorithm in [29, 40], 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 $\Omega_R = [\mathcal{L}_{\min}^r, \mathcal{L}_{\max}^r]^m$, and the domain of sources is $\Omega_S = [\mathcal{L}_{\min}^s, \mathcal{L}_{\max}^s]^{m-1}$. The domains are discretized such that the number of sampling points per wavelength is fixed, such as 4 to 6 points per wavelength. When $m = 3$, the cluster trees for the receivers and sources are an octree and a quadtree, respectively; when $m = 2$, the cluster trees for the receivers and sources are a quadtree and a binary tree, 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, binary tree, respectively), each box is equally divided into 8 (4, 2, respectively) 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, 2, respectively) 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 $\{B_r, B_s\}$ with $B_r \in T_r$ and $B_s \in T_s$, $l(B_s) = \ell$ and $l(B_r) = L - \ell$, 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 $\{B_r, B_s\}$ satisfies

$$w(B_r)w(B_s) = (\mathcal{L}_{\max}^r - \mathcal{L}_{\min}^r)\mathcal{O}(2\lambda_{\min}) = \mathcal{O}\left(\frac{4\pi(\mathcal{L}_{\max}^r - \mathcal{L}_{\min}^r)}{\omega n_{\max}}\right) = \mathcal{O}\left(\frac{1}{\omega}\right),$$

where $w(B)$ is the size of box B and n_{\max} is the maximum value of the 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, $w(B_r)$ is divided by 2 while $w(B_s)$ is multiplied by 2, and so $w(B_r)w(B_s) = \mathcal{O}(1/\omega)$ is automatically satisfied. For the motivation of such condition on all box pairs $\{B_r, B_s\}$, please see [11, 29] for details. In the following, we mean by the equivalent sources the Chebyshev nodes on the source box B_s and by equivalent points the Chebyshev nodes on the receiver box B_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 $w(B_s) \geq \mathcal{O}(\frac{1}{\sqrt{\omega}})$. Correspondingly, the level of the receiver tree T_r varies from the root level (level 0) to the level $L_r \equiv L - L_s$.

- (1) **Initialization.** For each pair $\{B_r, B_s\}$ with B_r traversing all boxes at the root level of the receiver tree and B_s traversing all boxes at the leaf level of the source tree, interpolate two equivalent densities $\{\bar{f}_k^{B_r, B_s}\}$ at the equivalent sources $\mathbf{C}^{B_s} = \{\mathbf{s}_j^{B_s}\}_{j=1}^{p^{m-1}}$ in B_s from the two given input densities $\{f_k\}$ at all sources in $\mathbf{X}_S \cap B_s$:

$$\bar{f}_k^{B_r, B_s}(\mathbf{s}_j^{B_s}) = \sum_{\mathbf{s} \in B_s \cap \mathbf{X}_S} e^{-i\omega\tau(\mathbf{r}_c^{B_r}; \mathbf{s}_j^{B_s})} L_{\mathbf{C}^{B_s}}^{m-1}(\mathbf{s}; \mathbf{s}_j^{B_s}) e^{i\omega\tau(\mathbf{r}_c^{B_r}; \mathbf{s})} f_k(\mathbf{s}), \quad (89)$$

for $k = 1, 2$, where $\mathbf{r}_c^{B_r}$ is the center of the receiver box B_r .

- (2) **Recursion.** For ℓ from $L-1$ to L_s , for each pair $\{B_r, B_s\}$ with B_r traversing all boxes at level $L-\ell$ of the receiver tree and B_s traversing all boxes at level ℓ of the source tree, interpolate equivalent densities $\{\bar{f}_k^{B_r, B_s}\}$ at equivalent sources $\mathbf{C}^{B_s} = \{\mathbf{s}_j^{B_s}\}_{j=1}^{p^{m-1}}$ from equivalent densities $\{\bar{f}_k^{B_r, B_s^c}\}$ at equivalent sources $\mathbf{C}^{B_s^c} = \{\mathbf{s}_j^{B_s^c}\}_{j=1}^{p^{m-1}}$ of all children clusters of B_s and the parent cluster of B_r :

$$\bar{f}_k^{B_r, B_s}(\mathbf{s}_j^{B_s}) = \sum_{B_s^c} \sum_{q=1}^{p^{m-1}} e^{-i\omega\tau(\mathbf{r}_c^{B_r}; \mathbf{s}_j^{B_s})} L_{\mathbf{C}^{B_s}}^{m-1}(\mathbf{s}_q^{B_s^c}; \mathbf{s}_j^{B_s}) e^{i\omega\tau(\mathbf{r}_c^{B_r}; \mathbf{s}_q^{B_s^c})} \bar{f}_k^{B_r, B_s^c}(\mathbf{s}_q^{B_s^c}), \quad (90)$$

for $k = 1, 2$.

3. **Switching** 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 $\{B_r, B_s\}$ with B_r traversing all boxes at level L_r of the receiver tree and B_s traversing all boxes at level L_s of the source tree, compute the equivalent field $\{\bar{u}^{B_r, B_s}\}$ at equivalent points $\mathbf{C}^{B_r} = \{\mathbf{r}_j^{B_r}\}_{j=1}^{p^m}$ from equivalent densities $\{\bar{f}_k^{B_r, B_s}\}$ at equivalent sources $\mathbf{C}^{B_s} = \{\mathbf{s}_q^{B_s}\}_{q=1}^{p^{m-1}}$:

$$\bar{u}^{B_r, B_s}(\mathbf{r}_j^{B_r}) = \sum_{k=1}^2 \sum_{q=1}^{p^{m-1}} U_k(\mathbf{r}_j^{B_r}, \mathbf{s}_q^{B_s}) \bar{f}_k^{B_r, B_s}(\mathbf{s}_q^{B_s}). \quad (91)$$

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_r to $L - 2$, for each pair $\{B_r, B_s\}$ with B_r traversing all boxes at level $\ell + 1$ of the receiver tree and B_s traversing all boxes at level $L - \ell - 1$ of the source tree, interpolate the equivalent fields $\{\bar{u}^{B_r, B_s}\}$ at equivalent points $\{\mathbf{r}_j^{B_r}\}_{j=1}^{p^m}$ from equivalent fields $\{\bar{u}^{B_r^p, B_s^c}\}$ at equivalent points $C^{B_r^p} = \{\mathbf{r}_q^{B_r^p}\}_{q=1}^{p^m}$ of the parent level ℓ of the receiver tree and the children level $L - \ell$ of the source tree:

$$\bar{u}^{B_r, B_s}(\mathbf{r}_j^{B_r}) = \sum_{B_s^c} e^{i\omega\tau(\mathbf{r}_j^{B_r}; \mathbf{s}_c^{B_s^c})} \sum_{q=1}^{p^m} L_{C^{B_r^p}}^m(\mathbf{r}_j^{B_r}; \mathbf{r}_q^{B_r^p}) e^{-i\omega\tau(\mathbf{r}_q^{B_r^p}; \mathbf{s}_c^{B_s^c})} \bar{u}^{B_r^p, B_s^c}(\mathbf{r}_q^{B_r^p}), \quad (92)$$

where $\mathbf{s}_c^{B_s^c}$ is the center of the source box B_s^c .

- (2). For each pair $\{B_r, B_s\}$ with B_r traversing all boxes at the leaf level of the receiver tree and B_s traversing all boxes at the root level of the source tree, interpolate the equivalent fields $\{\bar{u}^{B_r, B_s}\}$ at $\mathbf{r} \in X_R \cap B_r$ from equivalent fields $\{\bar{u}^{B_r^p, B_s^c}\}$ at equivalent points $C^{B_r^p} = \{\mathbf{r}_j^{B_r^p}\}_{j=1}^{p^m}$ of the parent level $L - 1$ of the receiver tree and the children level 1 of the source tree:

$$\bar{u}^{B_r, B_s}(\mathbf{r}) = \sum_{B_s^c} e^{i\omega\tau(\mathbf{r}; \mathbf{s}_c^{B_s^c})} \sum_{j=1}^{p^m} L_{C^{B_r^p}}^m(\mathbf{r}; \mathbf{r}_j^{B_r^p}) e^{-i\omega\tau(\mathbf{r}_j^{B_r^p}; \mathbf{s}_c^{B_s^c})} \bar{u}^{B_r^p, B_s^c}(\mathbf{r}_j^{B_r^p}). \quad (93)$$

5. **Terminating** at the leaf level of the receiver tree, for each box B_r , sum up the equivalent fields over all the boxes of the source tree at the root level, and then according to equation (84) compute the representative function u at $\mathbf{r} \in X_R \cap B_r$:

$$u(\mathbf{r}) = \sum_{B_s} \bar{u}^{B_r, B_s}(\mathbf{r}).$$

Assume that the tree level L is even and that there are $\mathcal{O}(J) = \mathcal{O}(2^L)$ points in each direction. According to the complexity analysis in [29], we can obtain that the total complexity of **Algorithm 2** is

$$\mathcal{O}\left(p^4 J^{5/2} + p^5 J^{5/2} + p^4 J^3 + J^3 \log J\right). \quad (94)$$

5.3.1 MATLAB-based parallelized butterfly algorithm

In **Algorithm 2**, Ω_R and Ω_S are assumed to be cubed and squared, respectively when $m = 3$, while Ω_R is assumed to be a square when $m = 2$. Nevertheless, even if this assumption is not satisfied, following [40] we can artificially 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. Take the case $m = 3$ as an example. 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 then subdivide them until the leaf level is reached; second, the output receiver-points X_R (the input source-points X_S) should contain all vertices of boxes at the 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_r (B_s) have the same layout and amount to the same number of points. 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 (89)-(93) have dimensions that do not depend on the box pair $\{B_r, B_s\}$, indicating that we can execute the summation for all box pairs $\{B_r, B_s\}$ in parallel; in our MATLAB implementation, we use the “built-in” `parfor` (parallelized for) loop in the level of iterations through box pairs $\{B_r, B_s\}$; if the number of specified MATLAB workers is N_w , then the involved summations (89)-(93) for every N_w box pairs are executed simultaneously. Consequently, in comparison with the sequential butterfly algorithm, the parallelized butterfly algorithm reduces the total complexity from (94) to

$$\mathcal{O}\left(\frac{p^4 J^{5/2} + p^5 J^{5/2} + p^4 J^3 + J^3 \log J}{N_w}\right) = \mathcal{O}\left(\frac{J^3 \log J}{N_w}\right).$$

However, the input source-point set $\{\mathbf{s}_i\}_{i=1}^{M_S}$ (the output receiver-point set $\{\mathbf{r}_i\}_{i=1}^{N_R}$) may not meet the above requirements, and thus differs from the required input source-point set X_S (the required output receiver-point set X_R) under construction. To resolve such inconsistency, for $k = 1, 2$, each prepared input data \mathbf{f}_k defined on the input source-point set $\{\mathbf{s}_i\}_{i=1}^{M_S}$ can be used to interpolate for the required input data, namely, two groups of values of representative functions f_k 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, namely, the group of values of the wavefield u on the required output receiver-point set X_R ; finally, by interpolating onto the required output receiver-points, we obtain the wavefield on the targeted receiver-point set $\{\mathbf{r}_i\}_{i=1}^{N_R}$, namely, the vector \mathbf{u} in equation (78).

5.4 Planar layer sweeping algorithm

Considering that the wavefield under construction directs along the z direction so that layers and secondary-source surfaces become boxes and planar surfaces, respectively, we have developed the planar-layer based sweeping algorithm, which can be summarized as the following.

Algorithm 3. *Planar layer sweeping algorithm—discretized case.*

- **Offline stage.** *Partitioning the computational domain and computing the Babich ingredients.*
 - *Partition the computational domain into layers $\{\Omega_j\}_{j=0}^{N_{lay}}$ and secondary-source planes $\{S_j\}_{j=0}^{N_{lay}-1}$. The number of layers is one more than the number*

of secondary source planes since we need not set up a secondary-source plane in the last layer.

- At the primary source \mathbf{r}_0 , and at each secondary source sampled on S_j , compute the tables of the five Babich ingredients: τ^2 , v_0 , v_1 , Φ and $v_{0,\nu'}$ on a coarse mesh in the layer Ω_{j+1} for $j = 0, \dots, N_{\text{lay}} - 1$. In practice, we only compute those tables for a coarse set of secondary sources since we can generate tables for a dense set of secondary sources from the coarse tables by interpolation.
- The tables for the coarse set of secondary sources are compressed, then stored (on a hard drive), and can be used to construct the wavefield at all high frequencies and at many different primary sources.
- **Online stage.** Given a frequency parameter ω , construct the primary Green's function layer by layer.
 - At all secondary sources on S_j , the tables of the five Babich ingredients are reconstructed from the compressed tables to construct the Green's function \mathbf{G} and its normal derivative $\mathbf{G}_{,\nu'}$ in layer Ω_{j+1} .
 - For each table, first interpolate the data onto a finer mesh to resolve the highly oscillatory behavior of \mathbf{G} and then compute the wavefield u in Ω_{j+1} by the Huygens-Kirchhoff summation (78) with a quadrature rule, which is accelerated by the MATLAB-based parallelized butterfly algorithm.
 - If the sampling of secondary sources on the source plane S_j is not dense enough, then we can interpolate the tables from given source locations onto the region or segment bounded by these source locations. This is feasible because asymptotic ingredients are continuous functions of the source location. For instance, when $m = 3$, if the four source points s_A , s_B , s_C and s_D are vertices of a rectangular segment $ABCD$ 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 in the rectangular region $ABCD$.

We can use Figure 2 to give a brief illustration of 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 at the same row 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 two-term Babich approximation (39) to compute the wavefield excited by the large star (the primary source \mathbf{r}_0) at those circle points (including the smaller stars) so that the wavefield at those smaller stars (secondary sources) is available. Next, in layer 2, we compute Green's functions excited by those smaller stars in layer 1 at the squared points so that the Huygens-Kirchhoff summation (78) can be applied to compute the wavefield at those squared points. Once the wavefield at those smaller stars (secondary sources) in layer 2 is available, a similar process

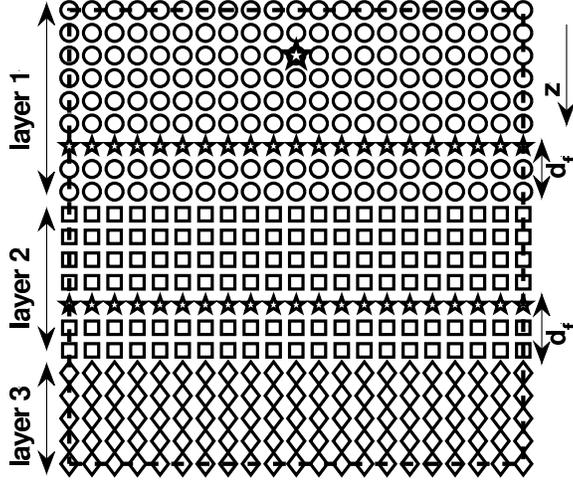


Figure 2: A 2-D sketch of the Huygens sweeping method. “large star”: a primary source; “small star”: secondary sources; d_f : the distance from a secondary source plane in one layer to the next layer; Green’s functions excited by the large star at “circle”, “squared”, and “diamond” points are computed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

can be used to compute the wavefield at those diamond points in layer 3. Consequently, the wavefield in the entire computational domain becomes available.

5.5 Complexity analysis

Since the offline and online 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 the computational domain is partitioned into $N_{lay} + 1$ planar layers and we set up N_{lay} secondary source planes. Among the $N_{lay} + 1$ layers, we only need to use the Huygens-Kirchhoff summation (78) to construct the wavefield in the N_{lay} layers away from the primary source \mathbf{r}_0 .

5.5.1 Offline stage: computing the Babich ingredients

At first, since the asymptotic ingredients are independent of frequency ω , they can be computed on a very coarse mesh. Second, 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 the Babich ingredients for densely sampled sources if necessary.

Suppose the computational domain is uniformly and coarsely discretized by n_0^m grid points, amounting to n_0 points in each direction. Since the domain is divided into $N_{lay} + 1$ planar layers, each layer has roughly $\mathcal{O}(\frac{n_0^m}{N_{lay}+1})$ points. On each of the N_{lay} secondary source planes, we set up n_0^{m-1} secondary sources, and we compute

for each secondary source the Babich ingredients in a layer so that the computational domain is restricted to that layer. According to [29], by using the high-order LxF-WENO schemes to compute the Babich ingredients, the computational complexity is $\mathcal{O}(\frac{n_0^m}{N_{lay}+1} \log n_0)$; next, those asymptotic ingredients are compressed into data tables of Chebyshev coefficients, and the complexity is $\mathcal{O}(\frac{n_0^m}{N_{lay}+1} \log n_0)$ as well. Consequently, the overall complexity for generating data tables at all $n_0^{m-1}N_{lay}$ secondary sources is

$$\mathcal{O}\left(\frac{2n_0^m}{N_{lay}+1} \log n_0 \cdot N_{lay}n_0^{m-1}\right) = \mathcal{O}(n_0^{2m-1} \log n_0).$$

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

On the other hand, to construct the wavefield, the whole computational domain is uniformly re-discretized by n_1 point in each direction. Thus, in each layer, we need to reconstruct, from the compressed tables of coefficients, five Babich ingredients for those specified secondary sources on roughly $\mathcal{O}(\frac{n_1^m}{N_{lay}+1})$ points, requiring complexity of $\mathcal{O}(\frac{n_1^m}{N_{lay}+1})$ for each ingredient and for each secondary source; see [29] for details.

5.5.2 Online stage: constructing global wavefield

Given a source point \mathbf{r}_0 and a frequency parameter ω , a fine mesh is required to capture the highly oscillatory wavefield in the computational domain. From the given refractive index $n(\mathbf{r})$ and ω , 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, taking roughly 4 to 6 grid points in each 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 is assumed to be $N = n_2^m$ where n_2 is the number of points in each direction.

Once the five Babich 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 (78). Given accuracy $\varepsilon > 0$, according to [11], 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 $\mathcal{O}(\frac{N}{N_{lay}+1} \log N)$, where the prefactor depends only on ε and does not depend on ω . Therefore, the overall complexity for constructing the Green's function by the butterfly algorithm in the N_{lay} layers is $\mathcal{O}(\frac{N_{lay}}{N_{lay}+1} N \log N) = \mathcal{O}(N \log N)$ for a given primary source point \mathbf{r}_0 and a given frequency. If computed in parallel, the complexity is further reduced to $\mathcal{O}(N/N_w \log N)$, where N_w denotes the number of workers in the parallelization.

6 Numerical examples

Unless otherwise stated, all computations were executed in a 16-core 2.0GHz Intel E5-2620 processor with 64 GBytes of RAM at the High Performance Computing Center (HPCC) of MSU. The offline stage was implemented in C codes and executed by using a single core while the online stage was constructed in MATLAB codes, where the butterfly algorithm was carried out in parallel in 10 cores via the parallel computing toolbox of MATLAB.

To validate the accuracy of our sweeping method, we obtain reference solutions by applying the FDTD method [51] directly on the associated time-domain equation. However, due to limited computing resources, we are only able to compute the FDTD-based solution at low frequencies, so that all comparisons will be carried out at low frequencies only. Unless otherwise stated, the grid size in the FDTD method is $h_{FDTD}=0.002$ in all 2-D examples, and $h_{FDTD}=0.01$ in all 3-D examples.

6.1 Two-dimensional examples

Example 1. A medium with constant refractive index is set up as the following:

- $\mu = (1 + 0.2 \sin(3\pi(x + 0.05)) \sin(0.5\pi z))^2$ and $\rho = 4\mu$ so that $n = \sqrt{\rho/\mu} = 2$.
- The computational domain is $\Omega = [0, 1] \times [0, 1]$.
- The distance between a layer and a secondary source line is $d_f = 0.1$.
- The mesh at the offline stage is 101×101 .

In this example, since n is constant, there is no caustic in the wavefield so that we expect that Babich's ansatz (5) is valid in the whole computational domain Ω .

At first, we check the accuracy of the Babich-ansatz based methods by testing both one-term and two-term truncations of the series (5). Since μ is not constant, we use the FDTD method to compute a reference solution for comparison. To reduce dispersion error in the FDTD method, we compute the FDTD-based solution in a small squared neighborhood of the primary source $\mathbf{r}_0 = [0.5, 0.2]^T$ of size 0.1, i.e., $[0.45, 0.55] \times [0.15, 0.25]$, which is uniformly discretized by 1001×1001 points with grid size 0.0001. Values of the wavefield at the set of points $\{[0.45 + 0.01m_1, 0.15 + 0.01m_2]^T | 0 \leq m_1, m_2 \leq 10\}$ except at the source \mathbf{r}_0 are computed at different frequencies, by using the FDTD method, and the one-term and two-term Babich approximations. Then, we measure the L^∞ error or the maximum absolute error between the FDTD solution and the two solutions with one-term and two-term truncations, respectively, as recorded in Table 1. Clearly, both the one-term and two-term Babich approximations yield accurate solutions uniformly at all frequencies, while the two-term approximation is more accurate at high frequencies.

Next, we compute the wavefield excited by the same source \mathbf{r}_0 in the whole domain Ω at different frequencies by **Algorithm 3**. To do so, we artificially set up one secondary-source line at $S_0 := \{z = 0.6\}$, and we update the wavefield in the layer $\Omega_1 := [0, 1] \times [0.7, 1]$ by using the Huygens-Kirchhoff summation (78) accelerated by **Algorithm 2**.

$\omega/(2\pi)$	1	2	4	8	16
One-term L^∞ error	8.9E-3	4.5E-3	1.3E-3	5.0E-4	1.4E-4
Two-term L^∞ error	1.0E-2	3.5E-3	1.1E-3	4.1E-4	9.3E-5

Table 1: Example 1. Comparison of FDTD solution and Babich-formula based solution in the small domain: $[0.45, 0.55] \times [0.15, 0.25]$.

The total running times of constructing the wavefield in Ω at different frequencies are recorded in Table 2. Assume that at mesh points in Ω_1 **Algorithm 2** produces the wavefield vector \mathbf{u}_B while straightforward matrix-vector multiplications give rise to the wavefield vector \mathbf{u}_D . To justify validity and efficiency of the butterfly algorithm, we compute the L^∞ errors between \mathbf{u}_D and \mathbf{u}_B , i.e. $\|\mathbf{u}_B - \mathbf{u}_D\|$ at different values of p and at different frequencies. These errors along with the running times of computing \mathbf{u}_D and \mathbf{u}_B are listed in Table 3. Evidently, the acceleration is dramatic, especially at high frequencies.

Mesh in Ω	161×161	321×321	641×641	1281×1281	2561×2561
$\omega/(2\pi)$	16	32	64	128	256
NPW	5	5	5	5	5
$T_{all}(p = 9)$	1.8	3.8	9.4	20.0	55.8
$T_{all}(p = 11)$	1.9	4.2	11.1	23.5	69.1
$T_{all}(p = 13)$	2.4	5.1	14.0	28.4	88.3

Table 2: Example 1. $\mathbf{r}_0 = [0.5, 0.2]^T$. T_{all} (unit: s): total CPU time for computing u in Ω ; NPW: the number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

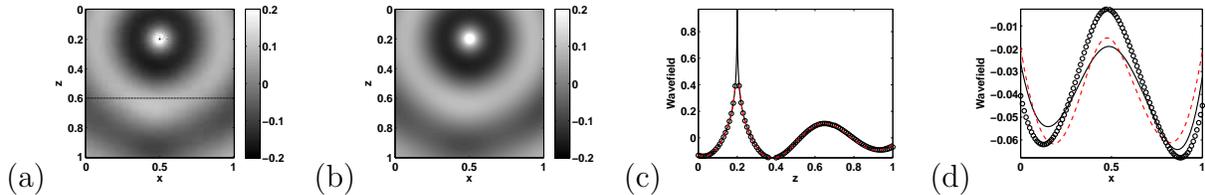


Figure 3: Example 1. $\mathbf{r}_0 = [0.5, 0.2]^T$, $\omega = 2\pi$: (a) Huygens principle based solution: $p = 11$ and NPW = 5; (b) FDTD solution. Real part of wavefields at (c): $x = 0$ and (d): $z = 0.8$. Circle: Huygens principle based solution; dashed line: two-term Babich approximation solution; solid line: FDTD solution.

Moreover, Figures 3, 4, and 5 show the numerical solutions at a low frequency $\omega = 2\pi$, an intermediate frequency $\omega = 8\pi$, and a high frequency $\omega = 32\pi$, respectively, by our approach and by the FDTD method, where a straight line at $z = 0.6$ shows

Mesh in Ω_1	169×53	337×105	673×209	1345×417	2689×833
$\omega/(2\pi)$	16	32	64	128	256
T_D	2.3	13.3	84.5	634.0	5062
NPW	5	5	5	5	5
$T_B(p=9)$	1.7	3.7	9.2	19.4	53.7
T_D/T_B	1.4	3.6	9.2	32.7	94.3
L_∞ -error	1.9E-2	2.5E-2	2.2E-2	1.9E-2	1.9E-2
$T_B(p=11)$	1.9	4.2	10.9	23.0	67.2
T_D/T_B	1.2	3.2	7.8	27.6	75.3
L_∞ -error	8E-3	1.2E-2	1.2E-2	9.1E-3	1.1E-2
$T_B(p=13)$	2.3	4.9	13.9	28.0	86.5
T_D/T_B	1.0	2.7	6.1	22.6	58.5
L_∞ -error	3E-3	6.9E-3	5.5E-3	4.7E-3	6.6E-3

Table 3: Example 1. $\mathbf{r}_0 = [0.5, 0.2]^T$. T_B (unit: s): running time for computing u in Ω_1 by **Algorithm 2**; T_D (unit: s): running time for computing u in Ω_1 by straightforward matrix-vector multiplication. NPW: the number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

the location of the secondary source line S_0 . We can see that the three solutions are consistent with each other, and the asymptotic solutions approach the FDTD solution reasonably well as the frequency becomes higher.

Example 2. Two-dimensional sinusoidal medium model:

- $\mu = (1 + 0.2 \sin(3\pi(x + 0.05)) \sin(0.5\pi z))^2$ and $\rho = 1$.
- The computational domain is $[0, 1] \times [0, 2]$.
- The distance between a layer and a secondary source line is $d_f = 0.1$.
- The mesh at the offline stage is 101×201 .

In this example, even though ρ is constant, the refractive index $n = \frac{1}{\sqrt{\mu}}$ is not a constant so that caustics may develop.

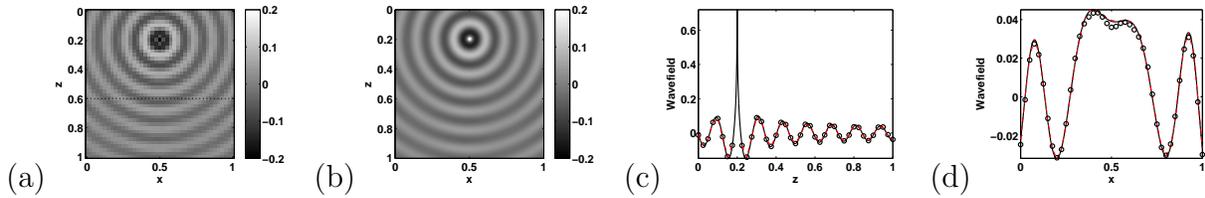


Figure 4: Example 1. $\mathbf{r}_0 = (0.5, 0.2)$, $\omega = 8\pi$: (a) Huygens principle-based solution: $p = 11$ and NPW = 5; (b) FDTD solution. Real part of wavefields at (c): $x = 0.50$; (d): $z = 0.8$. Circle: Huygens principle-based solution; dashed line: two-term Babich approximation solution; solid line: FDTD solution.

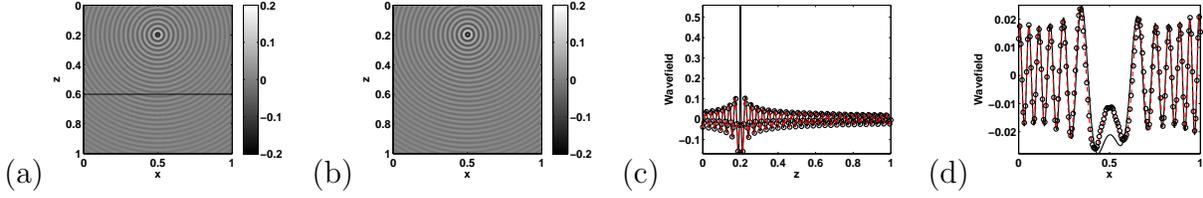


Figure 5: Example 1: $\mathbf{r}_0 = (0.5, 0.2)$, $\omega = 32\pi$: (a) Huygens principle-based solution: $p = 13$ and NPW = 5; (b) FDTD solution. Real part of wavefields at (c): $x = 0.50$; (d): $z = 0.8$. Circle: Huygens principle based solution; dashed line: two-term Babich approximation solution; solid line: FDTD solution.

We compute the wavefield excited by the source $\mathbf{r}_0 = [0.5, 0.2]^T$ in the whole domain Ω at different frequencies. According to **Algorithm 3**, we decompose Ω into six layers $\{\Omega_j\}_{j=0}^5$ distinguished by five secondary-source lines $\{S_j\}_{j=0}^4$ at the offline stage. During the online stage, we update the wavefield u in $\{\Omega_j\}_{j=1}^5$ by **Algorithm 2**. The total running times of constructing the wavefield in Ω at different frequencies are recorded in Table 4. Assume that at mesh points in Ω_1 **Algorithm 2** produces the wavefield vector \mathbf{u}_B while straightforward matrix-vector multiplications give rise to the wavefield vector \mathbf{u}_D . To justify validity and efficiency of the butterfly algorithm, we compute the L^∞ errors between \mathbf{u}_D and \mathbf{u}_B , i.e. $\|\mathbf{u}_B - \mathbf{u}_D\|$ at different values of p and at different frequencies. These errors along with the running times of computing \mathbf{u}_D and \mathbf{u}_B are listed in Table 5.

Mesh in Ω	241×481	481×961	961×1921	1921×3841	3841×7681
$\omega/(2\pi)$	32	64	128	256	512
NPW	6	6	6	6	6
$T_{all}(p = 9)$	18.9	48.7	104.3	289.7	691.0
$T_{all}(p = 11)$	21.8	58.8	122.8	364.0	838.3
$T_{all}(p = 13)$	25.8	73.8	149.1	466.4	1034.4

Table 4: Example 2: $\mathbf{r}_0 = [0.5, 0.2]^T$. T_{all} (unit: s): total CPU time for computing u in Ω ; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

In addition, we compare our numerical solution with the FDTD solution at $\omega = 32\pi$, as shown in Figure 6, where the five straight lines in Figure 6(a) show locations of secondary source lines $\{S_j\}_{j=0}^4$. One can observe that the two different solutions match each other very well in general. It seems that their differences are more pronounced in the vicinity of caustics. We believe that there are at least two reasons causing such phenomena: one is that there are an infinite number of rays propagating through caustics while our method truncates the secondary source lines so that we capture a finite number of rays only, and the other is that the FDTD solution loses accuracy due to the dispersion error at high frequencies.

Mesh in Ω_R	241×73	481×145	961×289	1921×577	3841×1153
$\omega/(2\pi)$	32	64	128	256	512
T_D	3.3	22.6	159.6	1195.8	12518
NPW	6	6	6	6	6
$T_B(p=9)$	3.7	9.5	20.4	54.7	126.4
T_D/T_B	0.9	2.4	7.8	21.9	99.0
L_∞ -error	6.1E-3	4.5E-3	4.1E-3	4.5E-3	4.2E-3
$T_B(p=11)$	4.4	11.4	24.2	69.0	153.5
T_D/T_B	0.8	2.0	6.6	17.3	81.6
L_∞ -error	1.1E-3	8.3E-4	7.0E-4	7.9E-4	8.1E-4
$T_B(p=13)$	5.1	14.4	28.7	89.3	194.2
T_D/T_B	0.6	1.6	5.6	13.4	64.5
L_∞ -error	3.5E-4	8.3E-5	1.0E-4	1.1E-4	6.5E-5

Table 5: Example 2: $\mathbf{r}_0 = [0.5, 0.2]^T$. T_B (unit: s): running time for computing u in Ω_1 by **Algorithm 2**; T_D (unit: s): running time for computing u in Ω_1 by straightforward matrix-vector multiplication. NPW: the number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

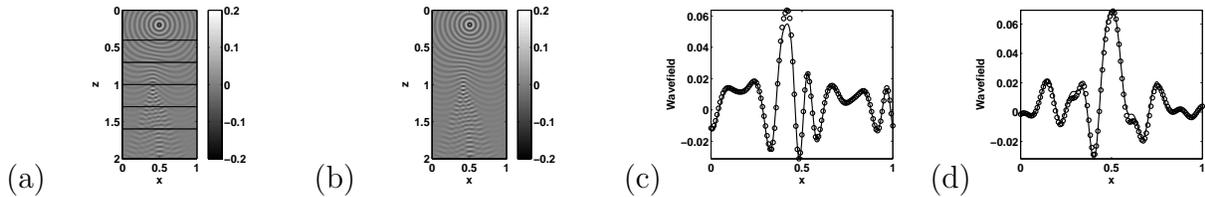


Figure 6: Example 2. $\mathbf{r}_0 = [0.5, 0.2]^T$, $\omega = 32\pi$: (a) Huygens principle based solution: $p = 13$ and NPW=6; (b) FDTD solution. Real part of wavefields at (c): $z = 1$; (d): $z = 1.8$. Circle: Huygens principle based solution; solid line: FDTD solution.

The wavefield u excited by a different primary source $\mathbf{r}_0 = [0.6, 0.35]^T$ at $\omega = 32\pi$ is computed and compared with the FDTD solution, as shown in Figure 7. Since the two-term Babich approximation (39) is accurate close to the source \mathbf{r}_0 , we see that even though the first secondary source line is close to \mathbf{r}_0 , the numerical solution is still consistent with the FDTD solution quite well.

Example 3. Two-dimensional waveguide model:

- $\mu = (1 - 0.5e^{-8(x-1)^2})^2$ and $\rho = 1$.
- The computational domain is $\Omega = [0, 2] \times [0, 2]$.
- The distance between a layer and a secondary source line is $d_f = 0.1$.
- The mesh at the offline stage is 201×201 .

We compute the wavefield excited by the source $\mathbf{r}_0 = [1.0, 0.2]^T$ in the whole domain Ω at different frequencies. According to **Algorithm 3**, we decompose Ω into six

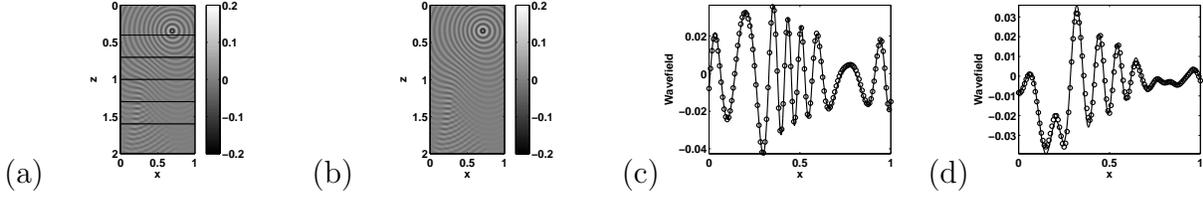


Figure 7: Example 2. $\mathbf{r}_0 = [0.5, 0.2]^T$ and $\omega = 32\pi$. (a) Huygens principle based solution: $p = 13$ and NPW=6; (b) FDTD solution. Real part of wavefields at (c): $z = 1$ and (d): $z = 1.8$. Circle: Huygens principle-based solution; solid line: FDTD solution.

layers $\{\Omega_j\}_{j=0}^5$ distinguished by five secondary source lines $\{S_j\}_{j=0}^4$ at the offline stage. During the online stage, we update the wavefield u in $\{\Omega_j\}_{j=1}^5$ by **Algorithm 2**. The total running times for constructing the wavefield u in Ω are recorded in Table 6. Assume that at mesh points in Ω_1 **Algorithm 2** produces the wavefield vector \mathbf{u}_B while straightforward matrix-vector multiplications give rise to the wavefield vector \mathbf{u}_D . To justify validity and efficiency of the butterfly algorithm, we compute the L^∞ errors between \mathbf{u}_D and \mathbf{u}_B , i.e. $\|\mathbf{u}_B - \mathbf{u}_D\|$ at different values of p and at different frequencies. These errors along with the running times of computing \mathbf{u}_D and \mathbf{u}_B are listed in Table 7.

Mesh in Ω	511×511	1021×1021	2051×2051	4101×4101
$\omega/(2\pi)$	32	64	128	256
NPW	4	4	4	4
$T_{all}(p = 9)$	63.1	178.2	386.0	1194.6
$T_{all}(p = 11)$	77.8	231.2	490.0	1588.1
$T_{all}(p = 13)$	97.5	305.8	630.2	2092.8

Table 6: Example 3: $\mathbf{r}_0 = [1.0, 0.2]^T$. T_{all} (unit: s): total CPU time for computing u in Ω ; NPW: the number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

In addition, we compare our numerical solution with the FDTD solution at $\omega = 32\pi$, as shown in Figure 8, where the five straight lines in Figure 8(a) show locations of secondary-source lines $\{S_j\}_{j=0}^4$.

The wavefield u excited by a different primary source $\mathbf{r}_0 = [1.0, 0.35]^T$ at $\omega = 32\pi$ is computed and compared with the FDTD solution, as shown in Figure 9, where the first secondary-source line S_0 is located close to \mathbf{r}_0 .

Mesh in Ω_R	561×81	1121×161	2241×321	4481×641
$\omega/(2\pi)$	32	64	128	256
T_D	17.6	119.7	892.1	10220
NPW	4	4	4	4
$T_B(p=9)$	12.6	35.4	75.0	229.8
T_D/T_B	1.4	3.4	11.9	44.5
L_∞ -error	1.7E-2	1.9E-2	1.8E-2	1.4E-2
$T_B(p=11)$	15.1	45.8	96.3	309.9
T_D/T_B	1.2	2.6	9.3	33.0
L_∞ -error	3.8E-3	4.9E-3	5.1E-3	5.2E-3
$T_B(p=13)$	19.2	60.2	123.8	412.4
T_D/T_B	0.92	2.0	7.2	24.8
L_∞ -error	1.1E-3	1.5E-3	8.8E-4	9.7E-4

Table 7: Example 3: $\mathbf{r}_0 = [1.0, 0.2]^T$. T_B (unit: s): running time for computing u in Ω_1 by **Algorithm 2**; T_D (unit: s): running time for computing u in Ω_1 by straightforward matrix-vector multiplication. NPW: the number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

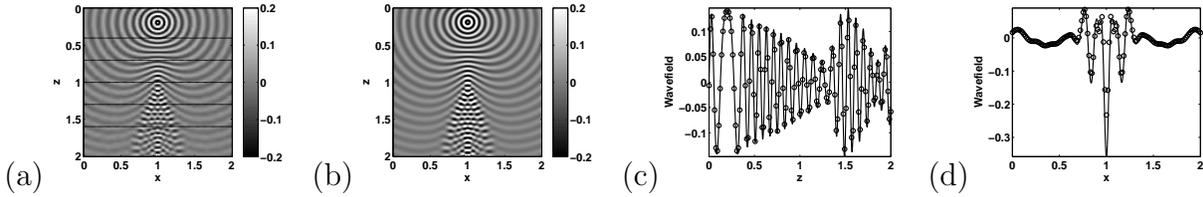


Figure 8: Example 3. $\mathbf{r}_0 = [1.0, 0.2]^T$ and $\omega = 32\pi$. (a) Huygens principle based solution: $p = 11$ and NPW=4; (b) FDTD solution. Real part of wavefields at (c): $x = 0.8$; (d): $z = 1.8$. Circle: Huygens principle based solution; solid line: FDTD solution.

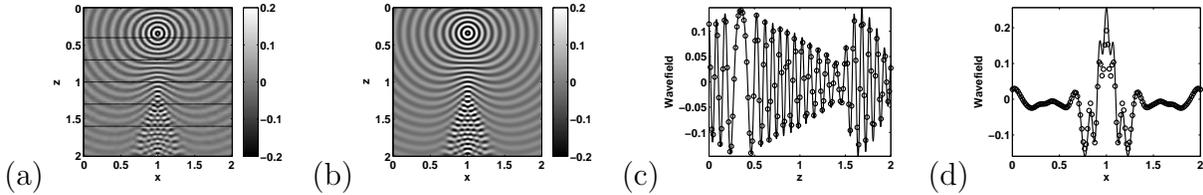


Figure 9: Example 3. $\mathbf{r}_0 = [1.0, 0.35]^T$ and $\omega = 32\pi$. (a) Huygens principle based solution: $p = 11$ and NPW=4; (b) FDTD solution. Real part of wavefields at (c): $x = 0.8$; (d): $z = 1.8$. Circle: Huygens principle based solution; solid line: FDTD solution.

6.2 Three-dimensional examples

Example 4. A constant refractive index model is set up as follows:

- $\mu = (3 - 1.75\exp(-((x - 1)^2 + (y - 1)^2 + (z - 1)^2)/0.64))^2$ and $\rho = 4\mu$ so that $n = 2$.
- The computational domain is $\Omega = [0, 2] \times [0, 2] \times [0, 2]$.
- The mesh at the offline stage is $51 \times 51 \times 51$.

We compute the wavefield excited by the source $\mathbf{r}_0 = [1.0, 1.0, 0.2]^T$ in the whole domain Ω at different frequencies. As $n = 2$ is constant, no caustics develop in this example. To check the performance of **Algorithm 3**, we artificially set up one secondary-source plane at $z = 1.2$, and compute the wavefield u in the layer $\Omega_1 = [0, 2] \times [0, 2] \times [1.3, 2]$. The total running time of constructing the wavefield in Ω_1 are recorded in Table 8.

Mesh in Ω	$41 \times 41 \times 41$	$81 \times 81 \times 81$	$161 \times 161 \times 161$	$261 \times 261 \times 261$
$\omega/(2\pi)$	2	4	8	16
NPW	5	5	5	5
$T_{all}(p = 9)$	83.0	290.4	1085.9	2888.4
$T_{all}(p = 11)$	99.2	357.2	1175.0	3733.9
$T_{all}(p = 13)$	137.1	429.2	1532.5	5241.1

Table 8: Example 4: $\mathbf{r}_0 = [1.0, 1.0, 0.2]^T$. T_{all} (unit: s): total CPU time for computing u in Ω ; NPW: the number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

In addition, we compare our numerical solution with the FDTD solution at three frequencies: $\omega = 2\pi, 4\pi$ and 8π . Corresponding results at $y = 1$ are shown in Figures 10, 11, and 12, respectively.

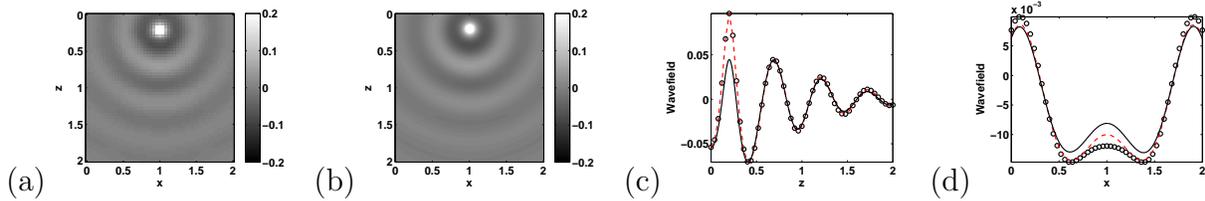


Figure 10: Example 4. Source point $[1.0, 1.0, 0.2]^T$ and $\omega = 2\pi$. (a) Huygens principle based solution at $y = 1$: $p = 11$ and NPW=5; (b) FDTD solution at $y = 1$. Real part of wavefields at (c): $x = 0.9$ and $y = 1$ and at (d): $y = 1$ and $z = 1.4$. Circle: Huygens principle-based solution; dashed line: two-term Babich approximation solution; solid line: FDTD solution.

Example 5. Three-dimensional Gaussian model:

- $\mu = (3 - 1.75\exp(-((x - 1)^2 + (y - 1)^2 + (z - 1)^2)/0.64))^2$ and $\rho = 1$.

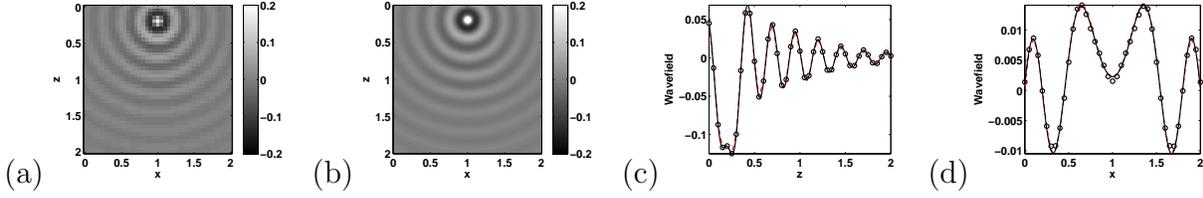


Figure 11: Example 4. Source point $[1.0, 1.0, 0.2]^T$ and $\omega = 4\pi$. (a) Huygens principle based solution at $y = 1$: $p = 11$ and NPW=5; (b) FDTD solution at $y = 1$. Real part of wavefields at (c): $x = 0.9$ and $y = 1$ and at (d): $y = 1$ and $z = 1.4$. Circle: Huygens principle-based solution; dashed line: two-term Babich approximation solution; solid line: FDTD solution.

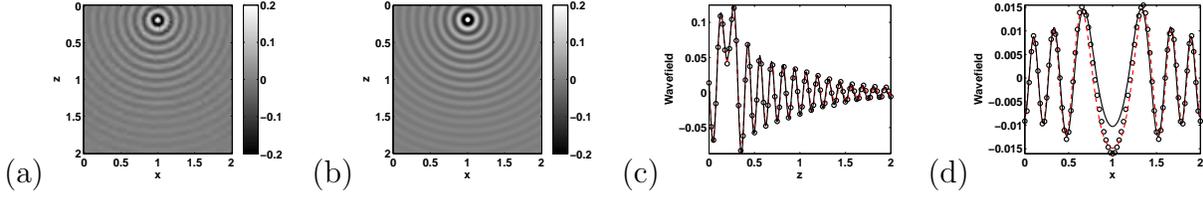


Figure 12: Example 4. Source point $(1.0, 1.0, 0.2)$ and $\omega = 8\pi$. (a) Huygens principle based solution at $y = 1$: $p = 13$ and NPW=5; (b) FDTD solution at $y = 1$. Real part of wavefields at (c): $x = 0.9$ and $y = 1$ and at (d): $y = 1$ and $z = 1.4$.

- The computational domain is $\Omega = [0, 2] \times [0, 2] \times [0, 2]$.
- The mesh at the offline stage is $51 \times 51 \times 51$.

We compute the wavefield excited by the source $\mathbf{r}_0 = [1.0, 1.0, 0.2]^T$ in the whole domain Ω at different frequencies. According to **Algorithm 3**, we set up one secondary-source plane at $z = 1.2$, and compute the wavefield u in the layer $\Omega_1 = [0, 2] \times [0, 2] \times [1.3, 2]$. The total running times of constructing the wavefield in Ω are recorded in Table 9.

Mesh in Ω	$31 \times 31 \times 31$	$61 \times 61 \times 61$	$131 \times 131 \times 131$	$261 \times 261 \times 261$
$\omega/(2\pi)$	4	8	16	32
NPW	5	5	5	5
$T_{all}(p = 9)$	83.0	204.9	701.7	2888.4
$T_{all}(p = 11)$	92.6	251.9	915.2	3733.9
$T_{all}(p = 13)$	109.7	303.4	1299.7	5241.1

Table 9: Example 5: $\mathbf{r}_0 = [1.0, 1.0, 0.2]^T$. T_{all} (unit: s): Total CPU time for computing u in Ω ; NPW: number of points per wavelength; $p = 9, 11, 13$ Chebyshev nodes are used in **Algorithm 2**.

Next, we compare our numerical solution with the FDTD solution at frequency $\omega = 16\pi$. Numerical results at $y = 1$ are shown in Figure 13.

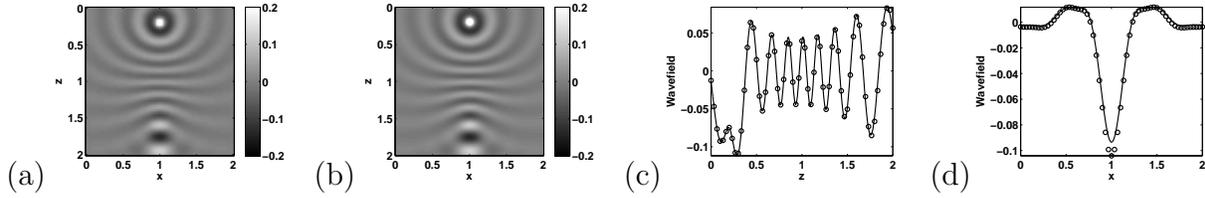


Figure 13: Example 5. Source point $(1.0, 1.0, 0.2)$ and $\omega = 16\pi$. (a) Huygens principle based solution at $y = 1$: $p = 13$ and NPW=5; (b) FDTD solution at $y = 1$. Real part of wavefields at (c): $x = 0.9$ and $y = 1$; and at (d): $y = 1$ and $z = 1.8$. Circle: Huygens principle-based solution; solid line: FDTD solution.

In addition, we compute the wave field u at high frequency $\omega = 64\pi$. Numerical results at $x = 1$, $y = 1$ and $z = 1.8$ are shown in Figure 14(a), (b) and (c), respectively.

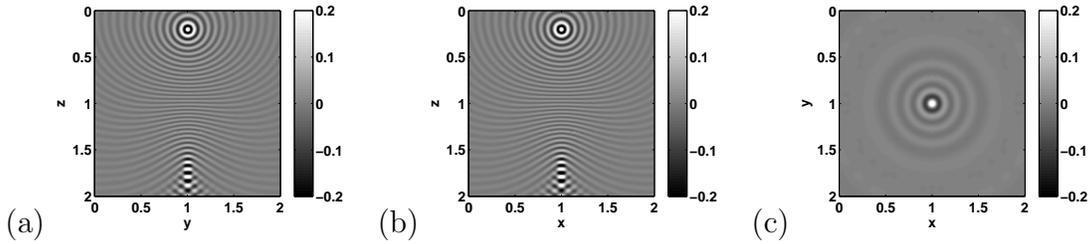


Figure 14: Example 5. Source point $(1.0, 1.0, 0.2)$ and $\omega = 64\pi$. Huygens principle based solution with $p = 13$ and NPW=5 at (a) $x = 1$, (b) $y = 1$, and (c) $z = 1.8$.

Finally, we compute the wavefield u at frequency $\omega = 64\pi$ at the source $\mathbf{r}_0 = [0.6, 0.6, 0.24]^T$. Numerical results at $x = 1.4$, $y = 1.4$ and $z = 1.8$ are shown in Figure 15(a), (b) and (c), respectively.

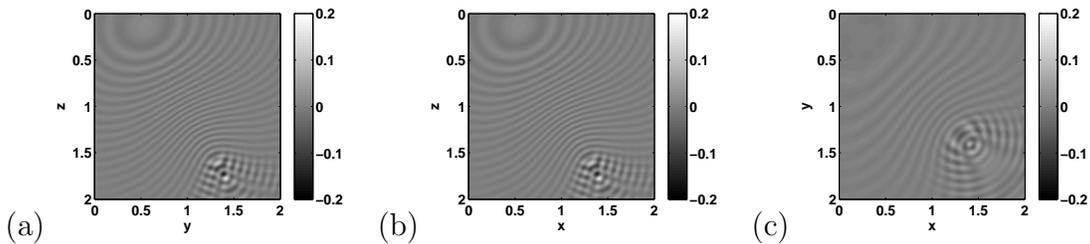


Figure 15: Example 5. Source point $(0.6, 0.6, 0.24)$ and $\omega = 64\pi$. Huygens principle-based solution with $p = 13$ and NPW=5 at (a) $x = 1.4$, (b) $y = 1.4$, and (c) $z = 1.8$.

7 Conclusion

Starting from Babich's expansion, we have developed a new fast Huygens sweeping method for solving the Helmholtz equation in inhomogeneous media in the high frequency regime and in the presence of caustics. The new method utilized the Huygens-Kirchhoff integral to integrate locally valid wavefields to construct globally valid wavefields, which are uniformly accurate near the source and remote from it. The Huygens-Kirchhoff summation was further accelerated by the butterfly algorithm, achieving nearly optimal complexity. Numerical experiments have illustrated the efficiency and accuracy of the method.

Acknowledgment

Qian is supported by NSF grants 1222368 and 1439979.

References

- [1] T. Alkhalifah. Efficient travelttime 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] V. M. Babich. The short wave asymptotic form of the solution for the problem of a point source in an inhomogeneous medium. *USSR Computational Mathematics and Mathematical Physics*, 5(5):247–251, 1965.
- [4] 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.
- [5] A. Bayliss, C. I. Goldstein, and E. Turkel. On accuracy conditions for the numerical computation of waves. *J. Comput. Phys.*, 59:396–404, 1985.
- [6] J. D. Benamou. An introduction to Eulerian geometrical optics (1992 - 2002). *J. Sci. Comp.*, 19:63–93, 2003.
- [7] J.-D. Benamou, O. Lafitte, R. Sentis, and I. Sollic. A geometrical optics-based numerical method for high frequency electromagnetic fields computations near fold caustics. Part I. *Journal of Computational and Applied Mathematics*, 156(1):93 – 125, 2003.
- [8] J.-D. Benamou, O. Lafitte, R. Sentis, and I. Sollic. A geometrical optics-based numerical method for high frequency electromagnetic fields computations near fold caustics. Part II. *Journal of Computational and Applied Mathematics*, 167(1):91–134, 2004.

- [9] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Second edition, Dover, New York, 2001.
- [10] R. Burridge. Asymptotic evaluation of integrals related to time-dependent fields near caustics. *SIAM J. Appl. Math.*, 55:390–409, 1995.
- [11] 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.
- [12] 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.
- [13] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Volume II*. Interscience Publishers, New York, 1964.
- [14] 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.
- [15] B. Engquist and O. Runborg. Computational high frequency wave propagation. *Acta Numerica*, 12:181–266, 2003.
- [16] B. Engquist and H.-K. Zhao. An approximate separability of Green’s functions for high frequency Helmholtz equations. *submitted*, xx:xxx–xxx, 2014.
- [17] S. Fomel, S. Luo, and H. K. Zhao. Fast sweeping method for the factored eikonal equation. *J. Comput. Phys.*, 228:6440–6455, 2009.
- [18] J. Hadamard. *Lectures on Cauchy’s Problem in Linear Partial Differential Equations*. Yale University Press; (reprinted Dover Publications, New York 1952), 1923.
- [19] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.
- [20] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.
- [21] C. Y. Kao, S. J. Osher, and J. Qian. Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations. *J. Comput. Phys.*, 196:367–391, 2004.
- [22] S. Leung and J. Qian. Eulerian Gaussian beam methods for Schrödinger equations in the semi-classical regime. *J. Comput. Phys.*, 228:2951–2977, 2009.
- [23] 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.
- [24] S. Leung, J. Qian, and R. Burridge. Eulerian Gaussian beams for high frequency wave propagation. *Geophysics*, 72:SM61–SM76, 2007.
- [25] X. D. Liu, S. J. Osher, and T. Chan. Weighted Essentially NonOscillatory schemes. *J. Comput. Phys.*, 115:200–212, 1994.

- [26] D. Ludwig. Uniform asymptotic expansions at a caustic. *Comm. Pure Appl. Math.*, 19:215–250, 1966.
- [27] 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.
- [28] S. Luo and J. Qian. Fast sweeping methods for factored anisotropic eikonal equations: multiplicative and additive factors. *J. Sci. Comput.*, 52:360–382, 2012.
- [29] 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.
- [30] 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. Anal.*, 52:23–44, 2014.
- [31] S. Luo, J. Qian, and H.-K. Zhao. Higher-order schemes for 3-D traveltimes and amplitudes. *Geophysics*, 77:T47–T56, 2012.
- [32] V. P. Maslov and M. V. Fedoriuk. *Semi-classical approximation in quantum mechanics*. D. Reidel Publishing Company, 1981.
- [33] E. Michielssen and A. Boag. A multilevel matrix decomposition algorithm for analysing scattering from large structures. *IEEE Trans. Antennas Propagat.*, 44:1086–1093, 1996.
- [34] J. Milnor. *Morse Theory, Annals of Math., No. 51*. Princeton University Press, 1973.
- [35] M. Motamed and O. Runborg. Taylor expansion and discretization errors in Gaussian beam superposition. *Wave Motion*, 47:421–439, 2010.
- [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. Anal.*, 28:907–922, 1991.
- [38] A. Pica. Fast and accurate finite-difference solutions of the 3D eikonal equation parametrized in celerity, in *ann. internat. mtg. Soc. of Expl. Geophysc.*, pages 1774–1777, 1997.
- [39] M. M. Popov. A new method of computation of wave fields using Gaussian beams. *Wave Motion*, 4:85–97, 1982.
- [40] J. Qian, W. Lu, L. Yuan, S. Luo, and R. Burridge. Eulerian geometrical optics and fast Huygens sweeping methods for three-dimensional time-harmonic high-frequency Maxwell’s equations in inhomogeneous meida. *xxx*, submitted:xxx–xxx, 2015.
- [41] J. Qian, S. Luo, and R. Burridge. Fast Huygens sweeping methods for multi-arrival Green’s functions of Helmholtz equations in the high frequency regime. *Geophysics*, 80:T91–T90, 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. Qian, L. Yuan, Y. Liu, S. Luo, and R. Burrige. Babich’s expansion and high-order Eulerian asymptotics for point-source Helmholtz equations. *xxx*, submitted:xxx–xxx, 2014.
- [46] 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.
- [47] 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.
- [48] W. W. Symes and J. Qian. A slowness matching Eulerian method for multivalued solutions of eikonal equations. *J. Sci. Comp.*, 19:501–526, 2003.
- [49] N. Tanushev, B. Engquist, and R. Tsai. Gaussian beam decomposition of high frequency wave fields. *J. Comput. Phys.*, 228:8856–8871, 2009.
- [50] N. Tanushev, J. Qian, and J. Ralston. Mountain waves and Gaussian beams. *SIAM J. Multiscale Modeling and Simulation*, 6:688–709, 2007.
- [51] A. Toflove and S. C. Haganess. *Computational Electrodynamics: The Finite Difference Time Domain Method, Second Editions*. Artech House, Norwood, MA, 2000.
- [52] 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.
- [53] J. E. Vidale and H. Houston. Rapid calculation of seismic amplitudes. *Geophysics*, 55:1504–1507, 1990.
- [54] B. S. White. The stochastic caustic. *SIAM J. Appl. Math.*, 44:127–149, 1984.
- [55] B. S. White, A. Norris, A. Bayliss, and R. Burrige. Some remarks on the Gaussian beam summation method. *Geophys. J. R. Astr. Soc.*, 89:579–636, 1987.
- [56] L. Ying. Sparse Fourier transform via butterfly algorithm. *SIAM J. Sci. Comput.*, 31:1678–1694, 2009.
- [57] L. Zhang, J. W. Rector, and G. M. Hoversten. Eikonal solver in the celerity domain. *Geophysical Journal International*, 162:1–8, 2005.
- [58] Y. T. Zhang, H. K. Zhao, and J. Qian. High order fast sweeping methods for static Hamilton-Jacobi equations. *J. Sci. Comp.*, 29:25–56, 2006.
- [59] H. K. Zhao. Fast sweeping method for eikonal equations. *Math. Comp.*, 74:603–627, 2005.