



# Hadamard Integrators for Wave Equations in Time and Frequency Domain: Eulerian Formulations via Butterfly Algorithms

Yuxiao Wei<sup>1</sup> · Jin Cheng<sup>1</sup> · Shingyu Leung<sup>2</sup> · Robert Burrridge<sup>3</sup> · Jianliang Qian<sup>4</sup> 

Received: 2 January 2024 / Revised: 5 June 2024 / Accepted: 14 July 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

## Abstract

Starting from Kirchhoff-Huygens representation and Duhamel's principle of time-domain wave equations, we propose novel butterfly-compressed Hadamard integrators for self-adjoint wave equations in both time and frequency domain in an inhomogeneous medium. First, we incorporate the leading term of Hadamard's ansatz into the Kirchhoff-Huygens representation to develop a short-time valid propagator. Second, using Fourier transform in time, we derive the corresponding Eulerian short-time propagator in the frequency domain; on top of this propagator, we further develop a time-frequency-time (TFT) method for the Cauchy problem of time-domain wave equations. Third, we further propose a time-frequency-time-frequency (TFTF) method for the corresponding point-source Helmholtz equation, which provides Green's functions of the Helmholtz equation for all angular frequencies within a given frequency band. Fourth, to implement the TFT and TFTF methods efficiently, we introduce butterfly algorithms to compress oscillatory integral kernels at different frequencies. As a result, the proposed methods can construct wave field beyond caustics implicitly and advance spatially overturning waves in time naturally with quasi-optimal computational complexity and memory usage. Furthermore, once constructed the Hadamard integrators can be employed to solve both time-domain wave equations with various initial conditions and frequency-domain wave equations with different point sources. Numerical examples for two-dimensional wave equations illustrate the accuracy and efficiency of the proposed methods.

**Keywords** Time-dependent wave equation · Helmholtz equation · High-frequency wave · Hadamard's ansatz · Butterfly algorithm · Caustics

**Mathematics Subject Classification** 65M80 · 65Y20

## 1 Introduction

We consider the Cauchy problem for the self-adjoint wave equation in  $m$ -dimensional space  $\mathbb{R}^m$ ,

$$\rho u_{tt} - \nabla \cdot (v \nabla u) = f, \quad \mathbf{x} \in \mathbb{R}^m, \quad t > 0 \quad (1.1)$$

with initial conditions

$$u(0, \mathbf{x}) = u^1(\mathbf{x}), \quad u_t(0, \mathbf{x}) = u^2(\mathbf{x}), \quad (1.2)$$

where  $t$  is time, the subscripts  $t$  and  $tt$  represent the first and second time derivatives, respectively, position  $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ , the gradient operator  $\nabla = [\partial_{x_1}, \partial_{x_2}, \dots, \partial_{x_m}]^T$ , both variables  $\rho$  and  $v$  are analytic and positive functions of position  $\mathbf{x}$ , characterizing certain physical parameters of the medium,  $f(t, \mathbf{x})$  is the source term, and  $u^1(\mathbf{x})$  and  $u^2(\mathbf{x})$  are compactly supported, highly oscillatory smooth functions.

Taking

$$f(t, \mathbf{x}) = \delta(t)\delta(\mathbf{x} - \mathbf{z}) \quad \text{and} \quad u^1 = u^2 = 0 \quad (1.3)$$

in equations (1.1) and (1.2) and applying Fourier transform in time to equation (1.1) accordingly, we obtain the frequency-domain wave equation, the Helmholtz equation,

$$\nabla \cdot (v \nabla \hat{u}) + \omega^2 \rho \hat{u} = -\delta(\mathbf{x} - \mathbf{z}), \quad (1.4)$$

with the Sommerfeld radiation condition imposed at infinity, where  $\omega$  is the angular frequency and  $\mathbf{z}$  is the source.

When the initial conditions in (1.2) are highly oscillatory or the angular frequency in (1.4) is large, the wave field will be highly oscillatory; however, direct numerical methods, such as finite-difference or finite-element methods, for such problems may suffer from dispersion or pollution errors [5, 7], so that such methods require an enormous computational grid to resolve these oscillations and are thus very costly in practice. Therefore, alternative methods, such as geometrical-optics-based asymptotic methods, have been sought to resolve these highly oscillatory wave phenomena.

The essential difficulty in applying geometric optics is that it cannot handle caustics easily [4, 8, 12, 22, 24, 37, 39, 40, 42, 57]. One approach is to replace the geometrical-optics ansatz with Gaussian beam (GB) summations [6, 46, 50, 51, 55, 56]. Eulerian formulations of GB summations have been developed for Helmholtz equations in the high-frequency regime [25, 27, 29] and time-dependent Schrödinger equations in the semi-classical regime [26]. The main challenge of this approach is that the computations must be done in the high-dimensional phase space. Another approach is to incorporate the geometrical-optics ansatz into the Huygens secondary-source principle, resulting in fast Huygens sweeping (FHS) methods, and these methods have been designed to solve time-dependent Schrödinger equations [18, 28], Helmholtz equations [37, 40], frequency-domain Maxwell's equations [38, 45], time-harmonic vectorial Maxwell's equations [23], and frequency-domain elastic wave equations [48], all in the presence of caustics. FHS methods can propagate wave fields through appropriately partitioned spatial layers by marching in that preferred spatial direction in a layer-by-layer fashion, but these methods only work in media where geodesics satisfy the sub-horizontal condition [53].

Although caustics occur with a high probability for wave propagation in inhomogeneous media [58], we note that we are still able to use the geometrical-optics type method mainly

because of the following fact [2, 53]: in an isotropic medium, the point-source eikonal equation has a locally smooth solution near the source point except the source point itself; this implies that caustics will not develop right away on the expanding wavefront away from the source. If we compute the geometrical-optics ansatz of frequency-domain Green's function of (1.4) in these caustic-free neighborhoods of point sources, for example, the WKBJ ansatz or Babich's ansatz, the locally valid Green's functions can be incorporated into the Huygens secondary-source principle to update the global wave field, resulting in fast Huygens sweeping (FHS) methods for Helmholtz equations [37, 40]; alternatively, incorporating the Hadamard's ansatz [11, 17] for time-domain Green's function of (1.1) into the Huygens secondary-source principle results in the Hadamard integrator [57] for the wave equation. We denote by  $\tilde{T}(\mathbf{x}_0)$  the time when the first caustic occurs for rays issued from a source point  $\mathbf{x}_0$ . For  $\Delta T < \min_{\mathbf{x}_0 \in \Omega} \tilde{T}(\mathbf{x}_0)$ , where  $\mathbf{x}_0$  is located in a certain bounded domain  $\Omega$ , the Hadamard's ansatz is valid in this domain  $\Omega$  so that we can propagate wave forward in time with the time step  $\Delta T$  using the Huygens secondary-source principle.

Although the frequency-domain Green's function is highly oscillatory, the kernel matrix generated on a uniform Eulerian grid can be constructed and applied with a quasi-linear computational cost using butterfly decomposition. Thus, the FHS method [37, 40] can construct high-frequency wave fields with quasi-linear computational complexity and memory usage. However, as a frequency-domain method, the FHS method propagates the wave field layer by layer along a preferred spatial direction, and the propagation along the artificial spatial direction requires geodesics to satisfy the sub-horizontal assumption, implying that the FHS method is applicable only to a limited set of specific media.

Although the time-domain Green's function is non-oscillatory in time direction, it has singularity on irregular wavefronts in time. The Hadamard integrator developed in [57] propagates the wave fields along the *natural* time direction so that the sub-horizontal assumption of geodesics in the FHS methods [37, 40] is no longer required in this new setting. However, the singularity on irregular wavefronts prevents the Lagrangian Hadamard integrator [57] from achieving quasi-linear computational complexity.

Then we immediately run into a question: is it possible to develop an Eulerian Hadamard integrator and use butterfly decomposition techniques similar to those in the FHS methods to achieve quasi-linear computational cost? The answer is yes! Our strategy is based on the following two observations: (1) the time-domain Green's function of the wave equation is the inverse Fourier transform in time of the corresponding frequency-domain Green's function of the Helmholtz equation, and as shown in [47], each term in Hadamard's ansatz for the time-domain Green's function is indeed the inverse Fourier transform in time of the corresponding term in Babich's ansatz [3, 37, 38] for the frequency-domain Green's function; (2) moreover, while the time-domain Green's function has singularity concentrated on irregular curved wavefronts which is unfriendly to numerical quadrature posed on regular mesh points, the frequency-domain Green's function only has a singularity at the source point so that it is amenable to numerical quadrature defined on regular mesh points. These two observations motivate us to take a detour to apply the time-domain Green's function efficiently by first carrying out relevant applications in the frequency domain and then coming back to the time domain: first, take the Fourier transform in time to go to the frequency domain; second, apply the frequency-domain Green's function to corresponding arguments; third, take the inverse Fourier transform in frequency to get back to time domain. This strategy leads to the time-frequency-time (TFT) method for the Cauchy problem of the wave equation.

Different from the Lagrangian Hadamard integrator developed in [57], the TFT method provides resolution-satisfying time sampling of wave solutions, implying that the Hadamard integrator is also a time-domain method for solving the Helmholtz equation (1.4). After

carefully handling the singularity of the delta function, we can use the TFT method to obtain the time-domain global Green's function numerically. By performing the Fourier transform in time, we obtain the frequency-domain Green's function, which is the solution to (1.4). We dub this strategy the time-frequency-time-frequency (TFTF) method.

To accelerate matrix–vector multiplications in the TFT and TFTF methods, we first construct low-rank separable representations of the Hadamard ingredients, including phase and Hadamard coefficients in Hadamard's ansatz, by solving the corresponding governing equations and utilizing the Chebyshev spectral expansions accordingly. Next we consider an algebraic compression tool called butterfly [10, 30, 31, 43], a multilevel numerical linear algebra algorithm well-suited for representing highly oscillatory operators, to compress the Hadamard integrator. We have applied the method in [34, 37, 40] to deal with integrals of Green's functions of the Helmholtz equation. To deal with integral kernels with singularities, we combine the hierarchical matrix representation and the butterfly algorithm by designing a suitable strategy for selecting proxy rows in the interpolation decomposition of butterfly construction, which allows the butterfly compressed Hadamard integrator to enjoy the quasi-linear complexity in the overall CPU time and memory usage with respect to the number of unknowns; such quasi-optimal complexity is valid for both the wave equation in the time-space domain and the Helmholtz equation in the frequency-space domain.

It is important to note that the Hadamard integrator is independent of initial conditions. Once constructed, it can be used to solve both time-domain wave equations with various initial conditions and frequency-domain wave equations with different point sources. Moreover, once these butterfly representations are constructed, they can be applied recursively when executing the Hadamard integrator; this unique feature allows the Hadamard integrator to be efficiently parallelized.

This paper develops an Eulerian Hadamard integrator. The approach inherits the advantages of the Lagrangian Hadamard integrator in [57] by marching in time direction; moreover, since the Eulerian integrator relies on the butterfly algorithm to speed up the matrix–vector multiplication arising from the integration process, the computational complexity of the Eulerian integrator for the wave equation in both time and frequency domain can be reduced to quasi-linear. Moreover, our Eulerian Hadamard integrator is also able to solve the point-source Helmholtz equation in inhomogeneous media by treating caustics implicitly and handling overturning waves naturally, thus settling the issue of how to remove the sub-horizontal condition in the fast Huygens sweeping methods [37, 38, 40, 45] to accommodate overturning waves.

The rest of the paper is organized as follows. We introduce in Sect. 2 the Kirchhoff-Huygens representation formula which utilizes Green's functions to propagate waves. Connecting Hadamard's ansatz and Babich's ansatz using Fourier transform, we propose in Sect. 3 a novel Eulerian Hadamard-Kirchhoff-Huygens (HKH) propagator for the Cauchy problem of the time-dependent wave equation and the point-source problem of the frequency-dependent wave equation (the Helmholtz equation), and we further obtain the Eulerian Hadamard integrators by applying it recursively. We then develop in Sect. 4 numerical strategies for implementing the Eulerian Hadamard integrators. To accelerate the constructions and applications of the integrators, in Sect. 5, we construct interpolative decomposition butterfly (IDBF) and hierarchically off-diagonal butterfly (HODBF) representations of the integral kernels, resulting in quasi-optimal computational complexity and memory usage. Section 6 presents two-dimensional (2-D) results to demonstrate the accuracy, efficiency and convergence of the proposed methods. We conclude the paper with some comments in Sect. 7.

## 2 Kirchhoff-Huygens Representation Formula

We are interested in solving the Cauchy problem for the self-adjoint wave equation (1.1) with initial conditions (1.2). Let  $S(t)$  be the solution operator corresponding to  $f = 0$  and  $u^1 = 0$ ; namely, for a function  $\psi$ ,  $u(t, \mathbf{x}) = (S(t)\psi)(\mathbf{x})$  solves (1.1) with the source term  $f = 0$  and the initial conditions  $u^1 = 0$  and  $u^2 = \psi$ . Then Duhamel's principle implies that the solution for wave equation (1.1) with initial conditions (1.2) is

$$u(t, \mathbf{x}_0) = \partial_t(S(t)u^1)(\mathbf{x}_0) + (S(t)u^2)(\mathbf{x}_0) + \int_0^t (S(t-s)f(s, \cdot))(\mathbf{x}_0)ds. \quad (2.1)$$

Consider the time-domain Green's function  $G(t, \mathbf{x}_0; \mathbf{x})$  which satisfies equation (1.1) with the source term and initial conditions specified in (1.3). Then, for a compactly supported function  $\psi$  such that the initial condition  $u^2 = \psi$ , by [57] the solution operator  $S$  can be written as

$$(S(t)\psi)(\mathbf{x}_0) = \int_V \rho(\mathbf{x})G(t, \mathbf{x}_0; \mathbf{x})\psi(\mathbf{x})d\mathbf{x}, \quad (2.2)$$

where  $V$  is a region of space which contains the support of  $\psi$  and does not change in time. Since the boundary integrals are not considered, we need to set a sufficiently large computational domain to ensure that the waves do not reach the boundary [57].

Then we have

$$\begin{aligned} u(t, \mathbf{x}_0) &= \int_V \rho(\mathbf{x})[G_t(t, \mathbf{x}_0; \mathbf{x})u^1(\mathbf{x}) + G(t, \mathbf{x}_0; \mathbf{x})u^2(\mathbf{x})]d\mathbf{x} \\ &+ \int_0^t \int_V \rho(\mathbf{x})G(t-s, \mathbf{x}_0; \mathbf{x})f(s, \mathbf{x})d\mathbf{x}ds. \end{aligned} \quad (2.3)$$

Differentiating (2.3) with respect to  $t$ , we obtain

$$\begin{aligned} u_t(t, \mathbf{x}_0) &= \int_V \rho(\mathbf{x})[G_{tt}(t, \mathbf{x}_0; \mathbf{x})u^1(\mathbf{x}) + G_t(t, \mathbf{x}_0; \mathbf{x})u^2(\mathbf{x})]d\mathbf{x} \\ &+ \int_0^t \int_V \rho(\mathbf{x})G_t(t-s, \mathbf{x}_0; \mathbf{x})f(s, \mathbf{x})d\mathbf{x}ds. \end{aligned} \quad (2.4)$$

We refer to (2.3) and (2.4) as the Kirchhoff-Huygens representation formula. When the medium is homogeneous, Green's function for the wave equation is known so that the formula has been used frequently in practice. When the medium is inhomogeneous and the source term  $f = 0$ , in [57] we have used Hadamard's ansatz to compute the needed Green's function for the formula, leading to a short-time valid propagator in a Lagrangian formulation, dubbed the Hadamard-Kirchhoff-Huygens (HKH) propagator, which is able to propagate a highly oscillatory wave field for a short period of time; furthermore, a Lagrangian Hadamard integrator is obtained by recursively applying the HKH propagator in time. In this article, we continue to use Hadamard's ansatz to construct Green's functions in the Kirchhoff-Huygens formula, but we compute the HKH propagator using an Eulerian formulation, resulting in an Eulerian Hadamard integrator.

To simplify the presentation, in the current work we only consider the two-dimensional case, where  $m = 2$ , and we leave the three-dimensional case as a future work. Consequently, we consider the following two cases:

- Case 1. The pure initial-value problem for equation (1.1), where  $f = 0$ , and  $u^1$  and  $u^2$  are compactly supported, highly oscillatory smooth functions.

- Case 2. The point-source problem for equation (1.4), where the frequency-domain Green's function of the Helmholtz equation naturally corresponds to the time-domain Green's function of the wave equation (1.1) with  $f = \delta(t)\delta(\mathbf{x} - \mathbf{x}_0)$  and  $u^1 = u^2 = 0$ .

In Case 1 for pure initial-value problems, we employ the same setup as in [57], aiming to achieve a quasi-optimal computational complexity using Eulerian formulations.

In Case 2 for point-source problems of the Helmholtz equation, since the time-domain Green's function corresponding to  $f = \delta(t)\delta(\mathbf{x} - \mathbf{x}_0)$  and  $u^1 = u^2 = 0$  in the Cauchy problem (1.1)–(1.2) is naturally linked to the frequency-domain Green's function of the Helmholtz equation, we extend the Hadamard integrator to point-source wave equations in frequency domain so as to solve the point-source Helmholtz equation. Because the Hadamard integrator marches in the time direction, it can naturally handle overturning waves in frequency domain, which is in sharp contrast to our previous frequency-domain asymptotic methods in [37, 40], in that the methods in [37, 40] are not able to treat overturning waves.

On the other hand, for a generic source term  $f(x, t) \not\equiv 0$  with compact support in both space and time, the Kirchhoff-Huygens representation (2.3)–(2.4) contains a volume integral of  $m + 1$  dimension, which is expensive to evaluate; therefore, we leave this as a future work.

### 3 Eulerian Hadamard Integrators

#### 3.1 Hadamard's Ansatz for Time-Domain Green's Functions

Consider Green's function of the self-adjoint wave equation

$$\rho u_{tt} - \nabla \cdot (v \nabla u) = \delta(t)\delta(\mathbf{x} - \mathbf{x}_0), \quad \mathbf{x} \in \mathbb{R}^m, \quad t > 0 \quad (3.1)$$

with initial conditions

$$u(0, \mathbf{x}) = 0, \quad u_t(0, \mathbf{x}) = 0. \quad (3.2)$$

Let

$$c = \sqrt{\frac{v}{\rho}}, \quad n = \frac{1}{c} = \sqrt{\frac{\rho}{v}}. \quad (3.3)$$

Then Hadamard's ansatz [57] is written as

$$u(t, \mathbf{x}_0; \mathbf{x}) = \sum_{s=0}^{\infty} v_s(\mathbf{x}_0; \mathbf{x}) f_+^{s-\frac{m-1}{2}} [t^2 - \tau^2(\mathbf{x}_0; \mathbf{x})], \quad (3.4)$$

where the Gelfand-Shilov generalized functions  $f_+^\lambda(\zeta)$  are given in [15]. They are defined for  $\lambda > -1$  as follows:

$$f_+^\lambda(\zeta) = \frac{\zeta_+^\lambda}{\lambda!}, \quad (3.5)$$

where

$$\zeta_+^\lambda = \begin{cases} 0, & \text{for } \zeta < 0, \\ \zeta^\lambda, & \text{otherwise,} \end{cases} \quad (3.6)$$

and by analytic continuation for other values of  $\lambda$ . We have the important relationship [15]

$$(f_+^\lambda)' = f_+^{\lambda-1}. \quad (3.7)$$

In particular,  $\lambda = 0$  corresponds to the Heaviside function, yielding a jump discontinuity at  $\zeta = 0$ , while by (3.7),  $\lambda = -1$  corresponds to the  $\delta$ -function. Here  $\tau(\mathbf{x}_0; \mathbf{x})$  is the phase function, also known as traveltime, satisfying the eikonal equation (3.8),

$$|\nabla \tau| = n, \quad \tau(\mathbf{x}_0; \mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = 0. \quad (3.8)$$

And the coefficient functions  $v_s(\mathbf{x}_0; \mathbf{x})$  in (3.4) satisfy the recurrent system

$$4\rho\tau \frac{dv_s}{d\tau} + v_s [\nabla \cdot (v \nabla \tau^2) + 2(2s - m)\rho] = \nabla \cdot (v \nabla v_{s-1}), \quad s = 0, 1, 2, \dots, \quad (3.9)$$

and  $v_{-1} \equiv 0$ . By leveraging properties of the Gelfand-Shilov function as demonstrated in [57], we have the initial condition for  $v_0$  at  $\mathbf{x}_0$ :

$$v_0(\mathbf{x}_0; \mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = \frac{n^m(\mathbf{x}_0)}{2\rho(\mathbf{x}_0)\pi^{\frac{m-1}{2}}} = \frac{n^{m-2}(\mathbf{x}_0)}{2v(\mathbf{x}_0)\pi^{\frac{m-1}{2}}}. \quad (3.10)$$

We introduce some definitions and properties of generalized functions [15] here. We first denote  $K$  the space of test functions which consists of all real functions  $\phi(\mathbf{x})$  with continuous derivatives of all orders and with bounded support. We next define a generalized function as any linear continuous functional on  $K$ . These functionals (generalized functions) which can be given by an equation such as

$$(\Psi, \phi) = \int_{R^n} \Psi(\mathbf{x})\phi(\mathbf{x})d\mathbf{x} \quad \text{for } \forall \phi \in K \quad (3.11)$$

with  $\Psi$  an absolutely integrable function in every bounded region of  $R^n$  shall be called regular, and all others (including the delta function) will be called singular. It is worth noting that  $f_+^\lambda(\zeta)$  is regular when  $\lambda > -1$  and singular when  $\lambda \leq -1$ .

### 3.2 Babich's Ansatz for Frequency-Domain Green's Functions

In order to develop Eulerian formulations, we introduce Babich's ansatz [3, 37, 38, 45] for the frequency-domain Green's function  $\hat{u} = \hat{G}(\omega, \mathbf{x}_0; \mathbf{x})$  of the Helmholtz equation (1.4), which is written as

$$\hat{u}(\omega, \mathbf{x}_0; \mathbf{x}) = \hat{G}(\omega, \mathbf{x}_0; \mathbf{x}) = \sum_{s=0}^{\infty} v_s(\mathbf{x}_0; \mathbf{x}) f_{s+1-\frac{m}{2}}(\tau(\mathbf{x}_0; \mathbf{x}), \omega), \quad (3.12)$$

where  $\tau$  still represents the traveltime,  $v_s$  represent the coefficient functions,

$$f_p(\tau, \omega) = i \frac{\sqrt{\pi}}{2} e^{ip\pi} \left( \frac{2\tau}{\omega} \right)^p H_p^{(1)}(\omega\tau) = i \frac{\sqrt{\pi}}{2} \left( \frac{2\tau}{\omega} \right)^p H_{-p}^{(1)}(\omega\tau), \quad (3.13)$$

and  $H_p^{(1)}$  is the  $p$ -th Hankel function of the first kind. In [37], using properties of the basis  $f_p(\tau, \omega)$ , we have obtained that  $\tau$  satisfies the eikonal equation (3.8) and  $v_s$  satisfy the recurrent system (3.9) with the initial condition (3.10). That is, the time-independent phase and coefficient functions in Hadamard's ansatz are the same as the frequency-independent phase and coefficient functions in Babich's ansatz. Furthermore, [38] shows that the basis functions of Hadamard's ansatz and Babich's ansatz are linked to each other:

$$\int_{\tau}^{\infty} e^{i\omega t} f_+^{(v-\frac{1}{2})}(t^2 - \tau^2(\mathbf{x}_0; \mathbf{x})) dt = \frac{1}{2} i \sqrt{\pi} \left( \frac{2\tau}{\omega} \right)^v e^{i\omega v} H_v^{(1)}(\omega\tau) \equiv f_v(\omega, \tau). \quad (3.14)$$

Because of the reciprocity, we have  $\hat{G}(\omega, \mathbf{x}_0; \mathbf{x}) = \hat{G}(\omega, \mathbf{x}; \mathbf{x}_0)$ , which will be used frequently in the following algorithmic development. Letting  $G_0(t, \mathbf{x}_0; \mathbf{x})$  and  $\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x})$  be the leading term of Hadamard's ansatz (3.4) and Babich's ansatz (3.12), respectively, for a point source located at  $\mathbf{x}_0$ , we have the following crucial relations,

$$G_0(t, \mathbf{x}_0; \mathbf{x}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) e^{-i\omega t} d\omega = \frac{1}{\pi} \Re \left[ \int_0^{\infty} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) e^{-i\omega t} d\omega \right], \quad (3.15)$$

where  $\Re$  represents the real part, and

$$\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) = \int_0^{\infty} G_0(t, \mathbf{x}_0; \mathbf{x}) e^{i\omega t} dt, \quad (3.16)$$

where  $G_0(t, \mathbf{x}_0; \mathbf{x})|_{t < 0} \equiv 0$ .

The above observations inspire us to replace the time-domain Green's function with a truncated finite-bandwidth Fourier summation of the frequency-domain Green's function, where the singularities of the time-domain Green's function are naturally transferred to the singularities at source points of the frequency-domain Green's function, resulting in the time-frequency-time (TFT) method for Case 1 and the time-frequency-time-frequency (TFTF) method for Case 2, as alluded to in Sect. 2.

### 3.3 TFT Eulerian Hadamard Integrator

Now we start with developing the TFT method for Case 1, where  $f \equiv 0$ , and  $u^1$  and  $u^2$  are compactly supported, highly oscillatory smooth functions, and we will compute the integrals of  $G_0$ ,  $\dot{G}_0$  and  $\ddot{G}_0$ , respectively, where, by the integration or integral of a Green's function, we mean the integral of the product of a Green's function and a sufficiently well-behaved function.

First, we apply the time-domain Green's function to  $\phi(\mathbf{x})$ , a test function in the space  $K$  as defined above, and use (3.15) to obtain

$$\begin{aligned} \int_{V_{x_0}} G_0(t, \mathbf{x}_0; \mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} &= \int_{V_{x_0}} \frac{1}{\pi} \int_{-\infty}^{\infty} e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \phi(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} e^{-i\omega t} d\omega \\ &= \frac{1}{\pi} \Re \left[ \int_0^{\infty} \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} e^{-i\omega t} d\omega \right]. \end{aligned} \quad (3.17)$$

Here  $V_{x_0}$  is an integration region independent of time  $t$ , satisfying

$$V_{x_0} \supseteq \{\mathbf{x} : \tau < t_0\} \supseteq \{\mathbf{x} : \tau < \Delta T\}, \quad (3.18)$$

where  $t_0$  is a constant that depends on the medium and satisfies  $t < \Delta T < t_0 < \bar{T}(\mathbf{x}_0)$ .

To apply the time derivatives of a Green's function in the frequency domain, we follow [57] to introduce the geodesic (ray) polar transformation for a given source  $\mathbf{x}_0$ :

$$P[\mathbf{x}_0] : \mathbf{x} \rightarrow (\tau, \omega), \quad (3.19)$$

where  $\tau$  is the travelttime,  $\omega \in \mathbb{S}^{m-1}$  is the take-off angle of the ray, and  $(\tau, \omega)$  are the geodesic polar coordinates. Within any neighborhood of  $\mathbf{x}_0$  not containing any caustics other than  $\mathbf{x}_0$ ,



there is one and only one ray connecting  $\mathbf{x}$  and  $\mathbf{x}_0$ , which means  $P[\mathbf{x}_0]$  is well-defined and one-to-one, and  $\mathbf{x}$  is a smooth function of the point  $\mathbf{y} = \mathbf{x}_0 + \tau\boldsymbol{\omega}$ . To facilitate our following discussions, we also write down the volume element as the following,

$$d\mathbf{x} = ds dS = c d\tau dS = c \left| \frac{\partial S}{\partial \boldsymbol{\omega}} \right| d\tau d\boldsymbol{\omega}, \quad (3.20)$$

where  $s$  is the arc length along the ray,  $dS = \left| \frac{\partial S}{\partial \boldsymbol{\omega}} \right| d\boldsymbol{\omega}$  is the element of area cut out on the wave front  $\tau = \text{const.}$  by rays emanating from the solid angle element  $d\boldsymbol{\omega}$  at the source. According to the appendix of [57], we have

$$\left| \frac{\partial S}{\partial \boldsymbol{\omega}} \right| = \frac{\tau^{m-1}}{4\rho_0 c_0^m \pi^{m-1} \rho c v_0^2}, \quad (3.21)$$

where  $\rho_0 = \rho(\mathbf{x}_0)$  and  $c_0 = c(\mathbf{x}_0)$ .

We now apply  $\hat{G}(t, \mathbf{x}_0; \mathbf{x})$  to a test function  $\phi \in K$  and use the geodesic polar transformation and its inverse,

$$\begin{aligned} \int \hat{G}(t, \mathbf{x}_0; \mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} &= 2t \int f_+^{-\frac{1}{2}-1} [t^2 - \tau^2] v_0(\mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} \\ &= 2t \int \left( -\frac{\partial}{2\tau \partial \tau} \right) f_+^{-\frac{1}{2}} [t^2 - \tau^2] v_0 \phi d\mathbf{x} \\ &= 2t \int_0^{2\pi} \int_0^t \left( -\frac{\partial}{2\tau \partial \tau} \right) f_+^{-\frac{1}{2}} [t^2 - \tau^2] v_0 \phi \left| \frac{\partial S}{\partial \boldsymbol{\omega}} \right| c d\tau d\theta \\ &= t \int_0^{2\pi} \left[ f_+^{-\frac{1}{2}} [t^2 - \tau^2] \frac{\phi}{4\rho_0 c_0^2 \pi \rho v_0} \right]_{\tau=t}^0 d\theta \\ &\quad + t \int_0^{2\pi} \int_0^t f_+^{-\frac{1}{2}} [t^2 - \tau^2] \left( \frac{\partial}{\partial \tau} \right) \left[ \frac{\phi}{4\rho_0 c_0^2 \pi \rho v_0} \right] d\tau d\theta \\ &= \frac{\phi(\mathbf{x}_0)}{\rho(\mathbf{x}_0)} + t \int_0^{2\pi} \int_0^t G_0(t, \mathbf{x}_0; \mathbf{x}) \frac{1}{v_0} \left( \frac{\partial}{\partial \tau} \right) \left[ \frac{\phi}{4\rho_0 c_0^2 \pi \rho v_0} \right] d\tau d\theta \\ &= \frac{\phi(\mathbf{x}_0)}{\rho(\mathbf{x}_0)} + t \int_{V_{\mathbf{x}_0}} G_0(t, \mathbf{x}_0; \mathbf{x}) \frac{1}{v_0} \left( \frac{\partial}{\partial \tau} \right) \left[ \frac{\phi}{4\rho_0 c_0^2 \pi \rho v_0} \right] \frac{1}{c \left| \frac{\partial S}{\partial \boldsymbol{\omega}} \right|} d\mathbf{x} \\ &= \frac{\phi(\mathbf{x}_0)}{\rho(\mathbf{x}_0)} + t \int_{V_{\mathbf{x}_0}} G_0(t, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left[ \frac{\phi}{\rho v_0} \right] \frac{\rho}{\tau} d\mathbf{x}, \end{aligned} \quad (3.22)$$

where  $\theta$  is the take-off angle  $\boldsymbol{\omega}$  when  $m = 2$ , and we have used (3.21), (3.10) and  $f_+^\lambda[0] = 0$ . Further using (3.17), we have

$$\begin{aligned} \int \hat{G}_0(t, \mathbf{x}_0; \mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} &= \frac{\phi(\mathbf{x}_0)}{\rho(\mathbf{x}_0)} + \frac{t}{\pi} \Re \\ &\quad \left[ \int_0^\infty e^{-i\omega t} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left[ \frac{\phi}{\rho v_0} \right] \frac{\rho v_0}{\tau} d\mathbf{x} d\omega \right]. \end{aligned} \quad (3.23)$$

Here the directional derivative

$$\frac{\partial}{\partial \tau} = c^2 \nabla \tau \cdot \nabla, \quad (3.24)$$

is obtained by utilizing the method of characteristics to solve the eikonal equation (3.8).

As for  $\ddot{G}_0$ , for  $t > 0$ , we have

$$\rho \ddot{G}_0(t, \mathbf{x}_0; \mathbf{x}) - \nabla \cdot (\nu \nabla G_0(t, \mathbf{x}_0; \mathbf{x})) = 0. \quad (3.25)$$

Applying  $\ddot{G}_0(t, \mathbf{x}_0; \mathbf{x})$  to a test function  $\phi(\mathbf{x})$  and using integration by parts and (3.17), we have

$$\begin{aligned} \int_{V_{x_0}} \ddot{G}_0(t, \mathbf{x}_0; \mathbf{x}) \phi(\mathbf{x}) d\mathbf{x} &= \int_{V_{x_0}} \nabla \cdot (\nu \nabla G_0(t, \mathbf{x}_0; \mathbf{x})) \frac{\phi}{\rho} d\mathbf{x} \\ &= \int_{V_{x_0}} G_0(t, \mathbf{x}_0; \mathbf{x}) \nabla \cdot (\nu \nabla \frac{\phi}{\rho}) d\mathbf{x} \\ &= \frac{1}{\pi} \Re \left[ \int_0^\infty \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \nabla \cdot (\nu \nabla \frac{\phi}{\rho}) d\mathbf{x} e^{-i\omega t} d\omega \right]. \end{aligned} \quad (3.26)$$

Now, taking  $\phi(\mathbf{x})$  as  $\rho(\mathbf{x})u^1(\mathbf{x})$  and  $\rho(\mathbf{x})u^2(\mathbf{x})$ , respectively, in the Kirchhoff-Huygens representation formulas (2.3)-(2.4), we obtain the following Eulerian formulations,

$$\begin{aligned} u(t, \mathbf{x}_0) &= u^1(\mathbf{x}_0) + \frac{1}{\pi} \Re \left[ \int_0^\infty \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \rho u^2(\mathbf{x}) d\mathbf{x} e^{-i\omega t} d\omega \right] \\ &\quad + \frac{t}{\pi} \Re \left[ \int_0^\infty \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left( \frac{u^1}{v_0} \right) \frac{\rho v_0}{\tau} d\mathbf{x} e^{-i\omega t} d\omega \right], \end{aligned} \quad (3.27)$$

and

$$\begin{aligned} u_t(t, \mathbf{x}_0) &= u^2(\mathbf{x}_0) + \frac{t}{\pi} \Re \left[ \int_0^\infty \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left( \frac{u^2}{v_0} \right) \frac{\rho v_0}{\tau} d\mathbf{x} e^{-i\omega t} d\omega \right] \\ &\quad + \frac{1}{\pi} \Re \left[ \int_0^\infty \int_{V_{x_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \nabla \cdot (\nu \nabla u^1) d\mathbf{x} e^{-i\omega t} d\omega \right]. \end{aligned} \quad (3.28)$$

Since the frequency-domain Green's function is only singular at the source point which is amenable to Eulerian evaluation on uniform meshes, we refer to (3.27)-(3.28) as the Eulerian Hadamard-Kirchhoff-Huygens (HKH)-TFTF propagator, which is used to propagate the wave field from  $\tau = 0$  to  $\tau = t$ , where  $0 < t \leq \Delta T$ . Although the HKH-TFTF propagator is only valid for a short-time period in a caustic-free neighborhood, recursively applying this propagator in time yields the *TFT Eulerian Hadamard integrator* to solve time-dependent wave equations globally in time, where caustics are treated implicitly and spatially overturning waves are handled naturally in *time domain*.

Here we remark in passing that we use the term “propagator” to indicate a short-time valid solution operator, while we use the term “integrator” to indicate a globally valid solution operator in either time-space or frequency-space domain.

### 3.4 TFTF Eulerian Hadamard Integrator

Now we develop the TFTF method for Case 2, point-source problems for the Helmholtz equation, by first solving the corresponding time-domain wave equations and then applying the Fourier transform in time to the resulting time-domain solutions.

According to Duhamel's principle, we can rewrite the corresponding time-domain wave equation of the point-source Helmholtz equation (1.4) in the same form as Case 1,

$$\rho(\mathbf{x})u_{tt}(t, \mathbf{x}) - \nabla \cdot [\nu(\mathbf{x})\nabla u(t, \mathbf{x})] = 0 \quad \text{for } t > 0, \quad (3.29)$$

with initial conditions

$$u(0, \mathbf{x}) = 0 \quad \text{and} \quad u_t(0, \mathbf{x}) = \frac{1}{\rho(\mathbf{z})} \delta(\mathbf{x} - \mathbf{z}). \quad (3.30)$$

The challenge arises from the fact that the wave solution in Case 2 is the time-domain Green's function, a generalized function. In each time step, we need to numerically apply generalized functions to other generalized functions. Specifically, with the initial conditions at  $T = 0$  containing the singular generalized function  $\delta$ , we have

$$u(t, \mathbf{x}_0) = \int \rho(\mathbf{x}) G(t, \mathbf{x}_0; \mathbf{x}) \frac{\delta(\mathbf{x} - \mathbf{z})}{\rho(\mathbf{z})} d\mathbf{x} = G(t, \mathbf{x}_0; \mathbf{z}) \quad (3.31)$$

and

$$u_t(t, \mathbf{x}_0) = \int \rho(\mathbf{x}) G_t(t, \mathbf{x}_0; \mathbf{x}) \frac{\delta(\mathbf{x} - \mathbf{z})}{\rho(\mathbf{z})} d\mathbf{x} = G_t(t, \mathbf{x}_0; \mathbf{z}) \quad (3.32)$$

in the *distributional sense*. Since  $G_t(t, \mathbf{x}_0; \mathbf{z})$  is *singular*, no locally integrable function is available that can be used to update  $u_t(t, \mathbf{x}_0)$  through *direct* integration. To overcome this challenge, instead of initializing at  $T = 0$  using the singular delta function, we initialize the wave field at  $T = \Delta T$  with a smooth approximation, which is obtained by truncating the inverse Fourier transform of the frequency-domain Green's function.

According to (2.3) and (2.4), we approximate the initial wave fields at time  $t \in (0, \Delta T]$  as follows,

$$u(t, \mathbf{x}) \approx G(t, \mathbf{z}; \mathbf{x}) \approx \frac{1}{\pi} \Re \left[ \int_0^B e^{-i\omega t} \hat{G}_0(\omega, \mathbf{z}; \mathbf{x}) d\omega \right], \quad (3.33)$$

and

$$u_t(t, \mathbf{x}) \approx \frac{1}{\pi} \Re \left[ \int_0^B (-i\omega) e^{-i\omega t} \hat{G}_0(\omega, \mathbf{z}; \mathbf{x}) d\omega \right], \quad (3.34)$$

where  $B$  is an artificial bandwidth, leading to truncated, frequency-domain smooth approximations to the wave fields. Although here we have dropped the frequency information beyond the bandwidth  $B$  of the time-domain solution, the Shannon-Nyquist Sampling theorem indicates that the remaining frequency content is sufficient for reconstructing the band-limited frequency-domain solution with angular frequency  $\omega < \frac{B}{2}$  from the truncated wave solutions.

Now recursively applying the HKH-TFTF propagator starting from the initial time  $t = \Delta T$  yields the global time-domain Green's function. Further we can utilize the Fourier transform to get the frequency-domain solution  $\hat{u}(\omega, \mathbf{x})$

$$\hat{u}(\omega, \mathbf{x}) = \int_0^\infty e^{i\omega t} u(t, \mathbf{x}) dt, \quad (3.35)$$

leading to the *TFTF Eulerian Hadamard integrator* to solve frequency-domain point-source wave equations (point-source Helmholtz equations) globally in time and thus globally in space, where the latter ("globally in space") is implied by the eikonal-defined traveltimes function.

Consequently, the TFTF Eulerian Hadamard integrator is able to treat caustics implicitly and handle spatially overturning waves naturally in *frequency domain*.

## 4 Numerics for Eulerian Hadamard Integrators

We present numerics for implementing TFT and TFTF Eulerian Hadamard integrators. We will design Eulerian methods to numerically discretize the HKH-TFTF propagator so as to obtain the corresponding Eulerian Hadamard integrators by recursively applying the propagator in time. For highly oscillatory wave fields, we maintain a fixed number of points per wavelength (PPW) to uniformly discretize the computational domain into regular grid points.

To start with, we briefly discuss higher-order sweeping schemes for the eikonal and transport equations, yielding the squared-phase function  $\tau^2$  and the Hadamard coefficient  $v_0$ , respectively. Subsequently, the low-rank representations based on Chebyshev interpolation are introduced to compress those smooth ingredients used in the propagator. Finally, we present preliminary algorithms for implementing TFT and TFTF Eulerian Hadamard integrators.

### 4.1 Hadamard Ingredients

In order to form the HKH-TFTF propagator (3.27)–(3.28), we need to compute some ingredients corresponding to  $\tau$  and  $v_0$ , including

$$\tau, v_0, \nabla \tau, \nabla \tau \cdot \nabla v_0. \quad (4.1)$$

Since  $\tau^2$  rather than  $\tau$  is smooth at source points, we obtain smooth representations for the following quantities:

$$\tau^2, v_0, \nabla \tau^2, \nabla \tau^2 \cdot \nabla v_0, \quad (4.2)$$

which can be transformed back to (4.1) via dividing by  $\sqrt{\tau^2}$ . We refer to the ingredients in (4.2) as Hadamard ingredients.

$\tau^2$  and  $v_0$  are obtained by solving the eikonal equation (3.8) first and the transport equation (3.9) afterwards.  $\nabla \tau^2$  is obtained by a proper finite-difference method. As for  $\nabla \tau^2 \cdot \nabla v_0$ , instead of using finite differencing which may reduce accuracy, we again use the transport equation (3.9) to obtain

$$\nabla \tau^2 \cdot \nabla v_0 = \frac{2m\rho v_0 - \nabla \cdot [v \nabla \tau^2] v_0}{2v}. \quad (4.3)$$

### 4.2 Numerical Schemes

The leading-order term of the Hadamard's ansatz is defined by two functions, the eikonal  $\tau$  satisfying the eikonal equation (3.8) and the Hadamard coefficient  $v_0$  satisfying the transport equation (3.9). Since we have assumed that the Hadamard's ansatz is valid locally around the source point, we need access to these two functions in order to construct the ansatz. Since the eikonal equation as a first-order nonlinear partial differential equation does not have analytical solutions in general, we have to use a robust, high-order numerical scheme to numerically solve this equation; moreover, the eikonal equation equipped with a point-source condition is even more tricky to deal with due to the upwind singularity at the source point [49]. To make the situation even more complicated, the transport equation (3.9) for the Hadamard coefficient  $v_0$  is weakly coupled with the eikonal equation (3.8) in that the coefficients of the former equation depend on the solution of the latter.

Fortunately, this set of weakly coupled equations with point-source conditions has been solved to high-order accuracy by using Lax-Friedrichs weighted essentially non-oscillatory (LxF-WENO) sweeping schemes as demonstrated in [52]. The high-order schemes in [52] have adopted essential ideas from many sources including [14, 19, 21, 32, 37, 41, 44, 49, 59, 60] and have been used in many applications. Consequently, we will adopt these schemes to our setting as well and we omit details here.

### 4.3 Multivariate Low-Rank Representation

The Hadamard ingredients (4.1) are time-independent. According to [52], if variable functions  $\rho$  and  $v$  are analytic, then  $\tau^2$  and  $v_0$  are analytic in the region of space containing a point source but no other caustics. Consider the source region  $\Omega_S$  and the receiver region  $\Omega_R$ , with  $\Omega_S \subset \Omega_R$ , such that rays originating from any  $x_0 \in \Omega_S$  will not intersect in  $\Omega_R$ , which means that no caustics will develop in  $\Omega_R$ ; consequently,  $\tau^2(x_0; x)$  and  $v_0(x_0; x)$  are analytic in  $\Omega_S \times \Omega_R$  so that we can construct Chebyshev-polynomials based low-rank representations of Hadamard ingredients with respect to both  $x_0$  and  $x$  [34, 36–38].

Letting  $f(x, y, x_0, y_0) = \tau^2, v_0, \frac{\partial \tau^2}{\partial x}, \frac{\partial \tau^2}{\partial y}, \nabla \tau^2 \cdot \nabla v_0$ , respectively, we consider the following analytical low-rank representation using Chebyshev interpolation,

$$f(x, y, x_0, y_0) \approx \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \sum_{l=1}^{n_l} F(i, j, k, l) \bar{T}_i(x) \bar{T}_j(y) \bar{T}_k(x_0) \bar{T}_l(y_0), \quad (4.4)$$

where  $n_i, n_j, n_k$ , and  $n_l$  are the orders of Chebyshev interpolation,  $F$  is a 4-D tensor of size  $n_i \times n_j \times n_k \times n_l$  which contains the spectral coefficients to be determined, and  $\bar{T}_i, \bar{T}_j, \bar{T}_k$ , and  $\bar{T}_l$  represent the Chebyshev interpolants defined via translating the standard Chebyshev polynomials  $T_m$  defined on  $[-1, 1]$  to the corresponding domain,

$$T_m(s) = \cos(m \arccos(s)), \quad s \in [-1, 1]. \quad (4.5)$$

We can obtain  $F$  by applying fast cosine transforms with respect to  $x, y, x_0$ , and  $y_0$  to the tensor  $f(x_i^c, y_j^c, x_{0,k}^c, y_{0,l}^c)$ , where  $x_i^c, y_j^c, x_{0,k}^c$ , and  $y_{0,l}^c$  are  $n_i, n_j, n_k$ , and  $n_l$ -order Chebyshev nodes in  $\Omega_R \times \Omega_S$ , respectively, which are also obtained by translating the  $n$ -order Chebyshev nodes  $\{s_m\}$  in  $[-1, 1]$ ,

$$s_m = \cos\left(\frac{2m-1}{2n}\right), \quad m = 1, 2, \dots, n. \quad (4.6)$$

To calculate  $f(x_i^c, y_j^c, x_{0,k}^c, y_{0,l}^c)$ , we adopt a computational strategy as used in [34]: first choose a region  $\Omega_c$  which is slightly larger than  $\Omega_R$  and on which the ingredients are still analytic, then compute numerically the ingredients in  $\Omega_c$  with sources located at  $(x_{0,k}^c, y_{0,l}^c)$ , respectively, and finally use cubic spline interpolation of the just computed ingredients on uniform grids to obtain  $f(x_i^c, y_j^c, x_{0,k}^c, y_{0,l}^c)$ . Here  $\Omega_c$  is introduced to ensure the accuracy of the interpolations near the boundary of  $\Omega_R$ .

After obtaining  $F$ , we can evaluate  $f(x, y, x_0, y_0)$  at any point  $(x, y, x_0, y_0) \in \Omega_R \times \Omega_S$  using (4.4), and the evaluation can be further accelerated by partial summation [40]. So far, we have finished the pre-computation and compression of the Hadamard ingredients in  $\Omega_R \times \Omega_S$  so that the integral kernels can be constructed by assembling these ingredients into the Hadamard's or Babich's ansatz.

Different from marching along a spatial direction [37], when we use the Hadamard integrator to march in time direction, we always take  $\Omega_S$  as a subset of  $\Omega_R$ . Actually, for

given caustic-free regions  $\Omega_S$  and  $\Omega_R$ , the maximum time step is bounded by the minimum traveltime

$$\tau_{\min} = \min_{\mathbf{x}_0 \in \partial\Omega_S, \mathbf{x} \in \partial\Omega_R} \tau(\mathbf{x}_0; \mathbf{x}). \quad (4.7)$$

We do not need to obtain the strict upper bound. Instead, we can determine the time step by dividing the minimum distance between  $\partial\Omega_S$  and  $\partial\Omega_R$  by the maximum wave speed  $\max c$ . Due to possible caustics in an inhomogeneous medium, we need to partition the computational domain  $\Omega$  into  $N_I$  source regions  $\{\Omega_S^\ell, \ell = 1, 2, \dots, N_I\}$  and designate a corresponding receiver region  $\Omega_R^\ell$  for each source region  $\Omega_S^\ell$ , such that we can utilize the locally valid short-time HKH propagator to update the wave fields in  $\Omega_S^\ell$  using the information in  $\Omega_R^\ell$ , separately. Since the waves will not propagate to the boundary of the computational domain, the wave field in the parts of the receiver regions that lie outside the computational domain is naturally set to zero. The partitioning of regions and the time step only depend on the medium.

With the low-rank approximations, we are able to evaluate the Hadamard ingredients with  $O(1)$  computational complexity, which is crucial for the numerical algorithm that follows. Additionally, we will demonstrate that the computational cost of constructing these low-rank approximations is independent of the oscillatory wave field.

Since both  $\tau^2$  and  $v_0$  are non-oscillatory analytic functions in a caustic-free region, this means that we can solve the eikonal and transport equations with high-order accuracy on a coarser grid (compared to the sampling grid for oscillatory wave fields); on the other hand, the orders of the multivariable Chebyshev interpolations are chosen according to the accuracy requirement in compressing the Hadamard ingredients, which are in turn medium-dependent. For instance, in the numerical examples that we are going to show, we set  $n_i = n_j = 13$  and  $n_k = n_l = 11$ , which do not change with the oscillatory nature of the wave field.

#### 4.4 Numerical Discretization of HKH-TFTF Propagator

To numerically implement the HKH-TFTF propagator, we shall first truncate the infinite frequency-domain integration interval  $[0, \infty)$  with an artificial frequency-bandwidth  $B$  that is problem-specific in the sense that the bandwidth depends on the highly-oscillatory initial conditions in Case 1 and on the angular frequency  $\omega$  in Case 2.

In the TFT method, once the oscillatory, smooth initial conditions (1.2) are given, the wave solution is essentially bandlimited in the sense that the frequency content of the corresponding wave solution is determined and decays rapidly outside a certain frequency band. Thus we can select a sufficiently large artificial frequency-domain bandwidth  $B$  to truncate the Fourier integral in frequency, where the fast decaying property ensures that truncation does not introduce significant truncation errors.

In the TFTF method, the bandwidth  $B = \kappa\omega$  is selected according to the angular frequency  $\omega$ , where  $\kappa$  is an oversampling parameter.

Using asymptotic properties of Hankel functions [1], for  $m = 2$ , we have the following asymptotic approximation for the frequency-domain Green's function,

$$\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \sim i \frac{\sqrt{\pi}}{2} v_0 \sqrt{\frac{2}{\pi\omega\tau}} e^{i\omega\tau - \frac{\pi}{4}} \quad \text{as } \omega\tau \rightarrow \infty. \quad (4.8)$$

Hence, truncation errors may occur when we reconstruct the wave solution in Case 2 by

$$G_0(t, \mathbf{x}_0; \mathbf{x}) = \Re \left[ \int_0^\infty e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \right] \approx \Re \left[ \int_0^B e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \right]. \quad (4.9)$$

For fixed  $t$  and  $\mathbf{x} \in \{\mathbf{x} : \tau(\mathbf{x}_0; \mathbf{x}) \neq t\}$ ,  $e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x})$  is oscillatory and decays at  $O(\frac{1}{\sqrt{\omega}})$ . Therefore, truncation of (4.9) to bandwidth  $B$  will introduce an error of no more than  $O(\frac{1}{\sqrt{B}})$ .

After truncating the infinite frequency domain to a bounded interval, we uniformly discretize the frequency interval  $[0, B]$  into

$$\{\omega_k : \omega_k = (k + \frac{1}{2})\Delta\omega, k = 0, 1, 2, \dots, N_\omega - 1\}, \quad (4.10)$$

where  $B = N_\omega \Delta\omega$ . Since the frequency-domain Green's function  $\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x})$  in the two-dimensional case has logarithmic singularity when  $\omega\tau \rightarrow 0$ ,

$$\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \sim \frac{\sqrt{\pi}}{2} v_0 \frac{2}{\pi} \ln(\omega\tau) \quad \text{as } \omega\tau \rightarrow 0, \quad (4.11)$$

we need to treat the integrals of  $\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x})$  over  $[0, \Delta\omega]$  with care.

Now we discretize the HKH-TFTF propagator (3.27)–(3.28) in the frequency domain as follows,

$$\begin{aligned} u(t, \mathbf{x}_0) &\approx u^1(\mathbf{x}_0) + \frac{1}{\pi} \Re \left[ \int_0^{\Delta\omega} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \rho u^2(\mathbf{x}) d\mathbf{x} e^{-i\omega t} d\omega \right] \\ &+ \frac{\Delta\omega}{\pi} \Re \left[ \sum_{k=1}^{N_\omega-1} e^{-i\omega_k t} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega_k, \mathbf{x}_0; \mathbf{x}) \rho u^2(\mathbf{x}) d\mathbf{x} \right] \\ &+ \frac{t}{\pi} \Re \left[ \int_0^{\Delta\omega} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left( \frac{u^1}{v_0} \right) \frac{\rho v_0}{\tau} d\mathbf{x} e^{-i\omega t} d\omega \right] \\ &+ \frac{t\Delta\omega}{\pi} \Re \left[ \sum_{k=1}^{N_\omega-1} e^{-i\omega_k t} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega_k, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left( \frac{u^1}{v_0} \right) \frac{\rho v_0}{\tau} d\mathbf{x} \right], \end{aligned} \quad (4.12)$$

and

$$\begin{aligned} u_t(t, \mathbf{x}_0) &\approx u^2(\mathbf{x}_0) + \frac{t}{\pi} \Re \left[ \int_0^{\Delta\omega} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left( \frac{u^2}{v_0} \right) \frac{\rho v_0}{\tau} d\mathbf{x} e^{-i\omega t} d\omega \right] \\ &+ \frac{t\Delta\omega}{\pi} \Re \left[ \sum_{k=1}^{N_\omega-1} e^{-i\omega_k t} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega_k, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\partial \tau} \right) \left( \frac{u^2}{v_0} \right) \frac{\rho v_0}{\tau} d\mathbf{x} \right] \\ &+ \frac{1}{\pi} \Re \left[ \int_0^{\Delta\omega} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \nabla \cdot (v \nabla u^1) d\mathbf{x} e^{-i\omega t} d\omega \right] \\ &+ \frac{\Delta\omega}{\pi} \Re \left[ \sum_{k=1}^{N_\omega-1} e^{-i\omega_k t} \int_{V_{\mathbf{x}_0}} \hat{G}_0(\omega_k, \mathbf{x}_0; \mathbf{x}) \nabla \cdot (v \nabla u^1) d\mathbf{x} \right]. \end{aligned} \quad (4.13)$$

Before we further discretize the propagator in space, we need to deal with the frequency-space integral

$$\int_{V_{x_0}} \int_0^{\Delta\omega} e^{-i\omega\tau} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \psi(\mathbf{x}_0; \mathbf{x}) d\mathbf{x}, \quad (4.14)$$

where  $\psi(\mathbf{x}_0; \mathbf{x})$  represents the frequency-independent part of the integrand in the propagator. The difficulty comes from the singularity of the frequency-domain Green's function when  $\omega \rightarrow 0$ . However, since  $\tau$  has a small upper bound that only depends on the medium, we have  $\omega\tau \sim 0$  uniformly; we thus use the polynomial approximation of the Hankel function  $H_0^1(\omega\tau)$  [1] to approximate the kernel  $\hat{G}_0$  so that

$$I(t, \mathbf{x}_0; \mathbf{x}) \doteq \int_0^{\Delta\omega} e^{-i\omega\tau} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \quad (4.15)$$

can be evaluated, and the technical details can be found in Appendix A.1. Then integral (4.14) can be rewritten using the time-dependent kernel  $I(t, \mathbf{x}_0; \mathbf{x})$

$$\int_{V_{x_0}} \int_0^{\Delta\omega} e^{-i\omega\tau} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \psi(\mathbf{x}_0; \mathbf{x}) d\mathbf{x} = \int_{V_{x_0}} I(t, \mathbf{x}_0; \mathbf{x}) \psi(\mathbf{x}_0; \mathbf{x}) d\mathbf{x}. \quad (4.16)$$

We now discretize the HKH-TFTF propagator in space. Consider a source region  $\Omega_S$  and the corresponding receiver region  $\Omega_R$  which are uniformly discretized into  $N$  and  $M$  grid points, respectively. Then, a key issue is how to evaluate the self-interaction (diagonal) terms of the kernels. In (4.12) and (4.13), we have the following singular diagonal terms

$$\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x})|_{\mathbf{x}=\mathbf{x}_0}, \quad (4.17)$$

$$I(t, \mathbf{x}_0; \mathbf{x})|_{\mathbf{x}=\mathbf{x}_0}, \quad (4.18)$$

$$\left[ \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \frac{\partial}{\tau \partial \tau} \left( \frac{u^k}{v_0} \right) \right] \Big|_{\mathbf{x}=\mathbf{x}_0}, \quad k = 1, 2, \quad (4.19)$$

$$\left[ \int_0^{\Delta\omega} e^{-i\omega\tau} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \frac{\partial}{\tau \partial \tau} \left( \frac{u^k}{v_0} \right) d\omega \right] \Big|_{\mathbf{x}=\mathbf{x}_0}, \quad k = 1, 2. \quad (4.20)$$

After uniformly discretizing the computational regions, the diagonal terms can be replaced by integral averages over a cell centered at  $\mathbf{x}_0$ , where the non-singular parts of the integrand are treated as constants. Letting  $c_j$  be the cell of size  $h$  centered at  $\mathbf{x}_0$ , the Appendix of [34] provides

$$\begin{aligned} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}_0) &\doteq \frac{1}{h^2} \int_{c_j} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\mathbf{x} \\ &\approx \frac{i}{(2hn_0\omega)^2} \left[ 8 \int_0^{\frac{\pi}{4}} \frac{hn_0\omega}{2\cos\theta} H_1^{(1)} \left( \frac{hn_0\omega}{2\cos\theta} \right) d\theta + 4i \right], \end{aligned} \quad (4.21)$$

where  $n_0 = n(\mathbf{x}_0)$ . Based on (4.21), we derive the approximation for

$$I(t, \mathbf{x}_0; \mathbf{x}_0) \doteq \frac{1}{h^2} \int_{c_j} \int_0^{\Delta\omega} e^{-i\omega\tau} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega d\mathbf{x} \quad (4.22)$$



in Appendix A.2, where we have used the polynomial approximation of the Hankel function  $H_1^{(1)}$ . We further show in Appendix A.3 that

$$\begin{aligned} & \left[ \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \frac{\partial}{\tau \partial \tau} \left( \frac{u^k}{v_0} \right) \right] \Big|_{\mathbf{x}=\mathbf{x}_0} \\ & \doteq \frac{1}{h^2} \int_{c_j} \left[ \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \frac{\partial}{\tau \partial \tau} \left( \frac{u^k}{v_0} \right) \right] d\mathbf{x} = 0, \quad k = 1, 2. \end{aligned} \quad (4.23)$$

Multiplying (4.23) by  $e^{-i\omega t}$ , we integrate the resulting formula over  $[0, \Delta\omega]$  to compute the last diagonal term, leading to

$$\begin{aligned} & \left[ \int_0^{\Delta\omega} e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \frac{\partial}{\tau \partial \tau} \left( \frac{u^k}{v_0} \right) d\omega \right] \Big|_{\mathbf{x}=\mathbf{x}_0} \\ & \doteq \frac{1}{h^2} \int_{c_j} \int_0^{\Delta\omega} e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \frac{\partial}{\tau \partial \tau} \left( \frac{u^k}{v_0} \right) d\mathbf{x} = 0, \quad k = 1, 2. \end{aligned} \quad (4.24)$$

At this point, we define the following frequency-dependent kernels:

$$U_1^\omega = [(U_1^\omega)_{i,j}] = \left[ \hat{G}_0(\omega, \mathbf{x}_0^i; \mathbf{x}^j) \right]_{1 \leq i \leq N, 1 \leq j \leq M}, \quad (4.25)$$

$$U_2^\omega = [(U_2^\omega)_{i,j}] = \left[ \frac{\hat{G}_0(\omega, \mathbf{x}_0^i; \mathbf{x}^j)}{2\tau^2(\mathbf{x}_0^i; \mathbf{x}^j)} \frac{\partial \tau^2(\mathbf{x}_0^i; \mathbf{x}^j)}{\partial x} \right]_{1 \leq i \leq N, 1 \leq j \leq M}, \quad (4.26)$$

$$U_3^\omega = [(U_3^\omega)_{i,j}] = \left[ \frac{\hat{G}_0(\omega, \mathbf{x}_0^i; \mathbf{x}^j)}{2\tau^2(\mathbf{x}_0^i; \mathbf{x}^j)} \frac{\partial \tau^2(\mathbf{x}_0^i; \mathbf{x}^j)}{\partial y} \right]_{1 \leq i \leq N, 1 \leq j \leq M}, \quad (4.27)$$

$$U_4^\omega = [(U_4^\omega)_{i,j}] = \left[ \frac{\hat{G}_0(\omega, \mathbf{x}_0^i; \mathbf{x}^j)}{2v_0(\mathbf{x}_0^i; \mathbf{x}^j)\tau^2(\mathbf{x}_0^i; \mathbf{x}^j)} \nabla \tau^2(\mathbf{x}_0^i; \mathbf{x}^j) \cdot \nabla v_0(\mathbf{x}_0^i; \mathbf{x}^j) \right]_{1 \leq i \leq N, 1 \leq j \leq M}, \quad (4.28)$$

and the corresponding time-dependent kernels

$$\check{U}_s^t = [(\check{U}_s^t)_{i,j}] = \left[ \int_0^{\Delta\omega} e^{i\omega t} U_s^\omega(\mathbf{x}_0^i; \mathbf{x}^j) d\omega \right]_{1 \leq i \leq N, 1 \leq j \leq M}, \quad s = 1, 2, 3, 4. \quad (4.29)$$

We further define vectors

$$\begin{aligned} f_1 &= [\rho(\mathbf{x}^1)u^2(\mathbf{x}^1), \quad \dots, \quad \rho(\mathbf{x}^M)u^2(\mathbf{x}^M)]^T, \\ f_2 &= [c(\mathbf{x}^1)u_x^1(\mathbf{x}^1), \quad \dots, \quad c(\mathbf{x}^M)u_x^1(\mathbf{x}^M)]^T, \\ f_3 &= [c(\mathbf{x}^1)u_y^1(\mathbf{x}^1), \quad \dots, \quad c(\mathbf{x}^M)u_y^1(\mathbf{x}^M)]^T, \\ f_4 &= [v(\mathbf{x}^1)u^1(\mathbf{x}^1), \quad \dots, \quad v(\mathbf{x}^M)u^1(\mathbf{x}^M)]^T, \\ f_5 &= [\nabla \cdot (v(\mathbf{x}^1)\nabla u^1(\mathbf{x}^1), \quad \dots, \quad \nabla \cdot (v(\mathbf{x}^M)\nabla u^1(\mathbf{x}^M))]^T, \\ f_6 &= [c(\mathbf{x}^1)u_x^2(\mathbf{x}^1), \quad \dots, \quad c(\mathbf{x}^M)u_x^2(\mathbf{x}^M)]^T, \\ f_7 &= [c(\mathbf{x}^1)u_y^2(\mathbf{x}^1), \quad \dots, \quad c(\mathbf{x}^M)u_y^2(\mathbf{x}^M)]^T, \\ f_8 &= [v(\mathbf{x}^1)u^2(\mathbf{x}^1), \quad \dots, \quad v(\mathbf{x}^M)u^2(\mathbf{x}^M)]^T. \end{aligned} \quad (4.30)$$

Then we introduce the following frequency-sampling functions of size  $N \times 1$  for  $1 \leq k \leq N_\omega$

$$\begin{aligned}\hat{u}_1(\omega_k) &= U_1^{\omega_k} f_1, \\ \hat{u}_2(\omega_k) &= U_2^{\omega_k} f_2 + U_3^{\omega_k} f_3 - U_4^{\omega_k} f_4, \\ \hat{u}_3(\omega_k) &= U_1^{\omega_k} f_5, \\ \hat{u}_4(\omega_k) &= U_2^{\omega_k} f_6 + U_3^{\omega_k} f_7 - U_4^{\omega_k} f_8,\end{aligned}\quad (4.31)$$

and the time-sampling functions of size  $N \times 1$  for  $0 < t_\ell \leq \Delta T$

$$\begin{aligned}\check{u}_1(t_\ell) &= \check{U}_1^{t_\ell} f_1, \\ \check{u}_2(t_\ell) &= \check{U}_2^{t_\ell} f_2 + \check{U}_3^{t_\ell} f_3 - \check{U}_4^{t_\ell} f_4, \\ \check{u}_3(t_\ell) &= \check{U}_1^{t_\ell} f_5, \\ \check{u}_4(t_\ell) &= \check{U}_2^{t_\ell} f_6 + \check{U}_3^{t_\ell} f_7 - \check{U}_4^{t_\ell} f_8.\end{aligned}\quad (4.32)$$

Once we obtain all the above sampling functions, according to (3.27) and (3.28), we update  $u(t_\ell, \Omega_S)$  and  $u_t(t_\ell, \Omega_S)$  for all  $0 < t_\ell \leq \Delta T$  as

$$u(t_\ell, \Omega_S) \approx \frac{1}{\pi} \Re \left[ \check{u}_1(t_\ell) + t_\ell \check{u}_2(t_\ell) + \Delta \omega \sum_{k=1}^{N_\omega-1} e^{-i\omega_k t_\ell} (\hat{u}_1(\omega_k) + t_\ell \hat{u}_2(\omega_k)) \right] \quad (4.33)$$

and

$$u_t(t_\ell, \Omega_S) \approx \frac{1}{\pi} \Re \left[ \check{u}_3(t_\ell) + t_\ell \check{u}_4(t_\ell) + \Delta \omega \sum_{k=1}^{N_\omega-1} e^{-i\omega_k t_\ell} (\hat{u}_3(\omega_k) + t_\ell \hat{u}_4(\omega_k)) \right]. \quad (4.34)$$

## 4.5 Direct Computation of the Hadamard Integrators

We first present a direct summation algorithm for the TFT Eulerian Hadamard integrator for Case 1, and based on this, we further present the TFTF Eulerian Hadamard integrator for Case 2.

**Algorithm 1** (TFT Eulerian Hadamard integrator for Case 1)

1. Uniformly discretize the computational domain  $\Omega$  into a wave-resolution-satisfying uniform grid; partition  $\Omega$  into  $N_I$  non-overlapping source regions  $\{\Omega_S^\ell, \ell = 1, 2, \dots, N_I\}$  and designate the corresponding receiver regions to be  $\{\Omega_R^\ell, \ell = 1, 2, \dots, N_I\}$ , set the initial time  $T_0$ , the large time step  $\Delta T < \tau_{\min}$  for updating the initial condition, the small time step  $\Delta t = \frac{\Delta T}{N_T}$  for wave resolution, and an ending time  $T_{\text{end}}$  to ensure that the waves do not reach the computational boundary; determine the bandwidth  $B$  according to the initial conditions; set the frequency step  $\Delta \omega = \frac{B}{N_\omega}$ , where  $N_\omega = O(B)$ , and generate  $\omega_k = (k + \frac{1}{2})\Delta \omega$  for  $k = 0, 1, 2, \dots, N_\omega - 1$ ; pre-compute and compress Hadamard ingredients (4.2) in all sub-region pairs; precompute the diagonal terms for the kernel matrices.
2. Initialize  $u(T_0, \mathbf{x}) = u^1(\mathbf{x})$  and  $u_t(T_0, \mathbf{x}) = u^2(\mathbf{x})$ ; set time  $T = T_0$  and the loop variable  $p = 0$ .
3. For the current time step with  $T = T_0 + p\Delta T$ :
  - (a) compute finite-difference derivatives of  $u(T, \mathbf{x})$  and  $u_t(T, \mathbf{x})$  and construct vectors  $f_1, f_2, \dots, f_8$ .

- (b) determine a sub-region  $\Omega_p$  of  $\Omega$  that contains the region of influence of current data by extending each direction of the non-zero region of the solution  $u(T, \mathbf{x})$  outward by  $\max c \Delta T$ ; designate the source regions intersecting with  $\Omega_p$  as the regions to be updated.
- (c) for all the source regions  $\Omega_S^\ell$  to be updated:
- i Interpolate the Hadamard ingredients via partial summation; construct the kernels

$$U_s^{\omega_k}, s = 1, 2, 3, 4, 1 \leq k \leq N_\omega - 1, \quad \text{and} \quad \check{U}_s^{n\Delta t}, s = 1, 2, 3, 4, 1 \leq n \leq N_T;$$

compute the matrix–vector multiplications to obtain the corresponding frequency- and time-sampling functions  $\{\hat{u}_i(\omega_k)\}$  and  $\{\check{u}_i(n\Delta t)\}$ :

$$\hat{u}_i(\omega_k), i = 1, 2, 3, 4, 1 \leq k \leq N_\omega - 1, \quad \text{and} \\ \check{u}_i(n\Delta t), i = 1, 2, 3, 4, 1 \leq n \leq N_T.$$

- ii Update wavefields  $u(T + n\Delta t, \Omega_S^\ell)$  and  $u_t(T + n\Delta t, \Omega_S^\ell)$  for  $n = 1, 2, \dots, N_T$  via (4.33) and (4.34).

- (d) Set to zero the wave fields within the source regions that do not require updating.

4. Update  $T = T_0 + (p + 1)\Delta T$ . If  $T < T_{end}$ , then  $p \leftarrow p + 1$  and go to Step 3; else, stop.

Now we consider the computational cost of Algorithm 1. For a given frequency bandwidth  $B$ , according to the Shannon-Nyquist Sampling theorem, we will have a spatial grid of size  $O(B^2)$  in the two-dimensional space. To ensure that wave field is well resolved, we have  $\Delta t = O(1/B)$ , leading to  $O(B^3)$  unknowns to compute.

As mentioned before,  $\Delta T = O(1)$  since it only depends on the medium, so is  $N_I$ . Thus we will perform Step 3 for  $O(1)$  times, and in each iteration, there are  $O(B)$  matrix–vector multiplications of size  $O(B^2 \times B^2)$ , resulting in a computational complexity of at least  $O(B^5)$ . It is extremely expensive and impractical compared to the number of unknowns. To resolve this issue, we will utilize butterfly algorithms and their hierarchical extensions to accelerate matrix–vector multiplication.

As for Case 2, we first use (3.33) to approximate the time-domain solution in the interval  $0 < t \leq \Delta T$  and then apply **Algorithm 1** to solve the following wave equation in the interval  $\Delta T < t \leq T_{end}$ ,

$$\rho u_{tt} - \nabla \cdot (\nu \nabla u) = 0, \quad \mathbf{x} \in \mathbb{R}^m, \quad \Delta T < t \leq T_{end}, \quad (4.35)$$

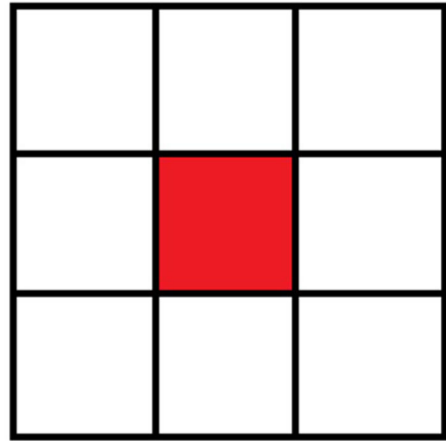
with initial conditions

$$u(\Delta T, \mathbf{x}) = \frac{1}{\pi} \Re \left[ \sum_k e^{-i\omega_k \Delta T} \hat{G}_0(\omega_k, \mathbf{z}, \mathbf{x}) \right], \quad (4.36)$$

$$u_t(\Delta T, \mathbf{x}) = \frac{1}{\pi} \Re \left[ \sum_k (-i\omega_k) e^{-i\omega_k \Delta T} \hat{G}_0(\omega_k, \mathbf{z}, \mathbf{x}) \right]. \quad (4.37)$$

Consequently, the frequency domain solution  $\hat{u}(\omega, \Omega)$  is constructed by (3.35). This leads to the TFTF Eulerian Hadamard integrator.

**Fig. 1** Illustration of  $\Omega_S^\ell$  and  $\Omega_R^\ell$ : The red square represents the source region  $\Omega_S^\ell$ , and the largest square divided into nine parts represents the corresponding receiver region  $\Omega_R^\ell$



## 5 Fast Computation of Hadamard Integrators

Here we propose a butterfly-based algorithm for the construction and application of each kernel, which is quasi-optimal in the sense that the complexity and memory usage are reduced from  $O(B^4)$  to  $O(B^2 \log^2(B))$ .

For large  $\omega_k$ , the discretized integral operators  $U_i^{\omega_k}$  are full rank due to the oscillation of  $\hat{G}(\omega_k, \mathbf{x}_0; \mathbf{x})$ , but the judiciously selected submatrices of the discretized operators are low-rank compressible due to the so-called complementary low-rank property [37]. Consequently, we can use the butterfly algorithm to compress those frequency-dependent kernels.

Even though the time-dependent kernels do not oscillate (which means that the complementary low-rank property is naturally satisfied), in this work we will use the same butterfly algorithm to compress them. Since the ranks of the submatrices of the time-dependent kernels are smaller than those of the frequency-dependent kernels, this fact naturally helps us reduce the computational complexity and memory usage required for compression and application.

We have partitioned the computational region  $\Omega$  into  $N_I$  source regions  $\{\Omega_S^\ell, \ell = 1, 2, \dots, N_I\}$  and designated a corresponding receiver region  $\Omega_R^\ell$  for each source region  $\Omega_S^\ell$ . As shown in Fig. 1, we further partition each receiver region  $\Omega_R^\ell$  into nine sub-regions, one of which overlaps with the corresponding source region  $\Omega_S^\ell$  (the red square), while the remaining eight (the white squares) are adjacent to the corresponding source region.

We first describe the interpolative-decomposition butterfly representation for the non-overlapping sub-regions and then introduce the hierarchical structure to handle the overlapping sub-region.

### 5.1 Interpolative-Decomposition Butterfly (IDBF) Representation

For the integral kernel  $K(\Omega_1, \Omega_2)$  defined on two non-overlapping regions  $\Omega_1$  and  $\Omega_2$ , we closely follow [10, 30, 31, 34, 35] to introduce the interpolative-decomposition butterfly algorithm. The algorithm first recursively subdivides the geometrical point sets  $\Omega_1$  and  $\Omega_2$  associated with the rows and columns of these operators into two subsets of approximately equal sizes using a k-dimensional tree clustering algorithm, until the number of elements in each subset is less than a preset value. When  $\Omega_1$  and  $\Omega_2$  have different sizes, the number of

levels of the binary tree will depend on the set of a smaller size. This procedure generates two complete binary trees  $\mathcal{T}_1$  and  $\mathcal{T}_2$  of  $L$  levels with root level 0 and leaf level  $L$ . In both trees, a non-leaf node  $\tau$  at level  $l$  has two children  $\tau_1$  and  $\tau_2$ , where  $\tau = \tau_1 \cup \tau_2$  and  $\tau_1 \cap \tau_2 = \emptyset$ . For a non-root node  $\tau$ , its parent is denoted  $p_\tau$ .

The  $L$ -level butterfly representation of the kernel  $K(\Omega_1, \Omega_2)$  requires the complementary low-rank property: at any level  $0 \leq l \leq L$ , for any node  $\tau$  at level  $l$  of  $\mathcal{T}_1$  and any node  $\nu$  at level  $L - l$  of  $\mathcal{T}_2$ , the subblock  $K(\tau, \nu)$  is numerically low-rank with rank  $r_{\tau, \nu}$  bounded by a small number  $r$  called the butterfly rank. As discussed in [34], even if there is a singularity in the integral kernel which may cause some subblocks to have non-constant ranks, the butterfly algorithm will still work when the singularity only exists at the interface of adjacent regions.

Thus, for any subblock  $K(\tau, \nu)$ , where  $\tau \in \mathcal{T}_1$  and  $\nu \in \mathcal{T}_2$ , the complementary low-rank property permits a low-rank representation using interpolative decomposition (ID) as

$$K(\tau, \nu) \approx K(\tau, \bar{\nu})V_{\tau, \nu}, \quad (5.1)$$

where the skeleton matrix  $K(\tau, \bar{\nu})$  contains  $r_{\tau, \nu}$  skeleton columns indexed by  $\bar{\nu}$ , and the interpolation matrix  $V_{\tau, \nu}$  has bounded entries. The ID can be computed via, for instance, rank-revealing QR decomposition with a relative tolerance  $tol$ , which actually determines the accuracy of the overall butterfly representation. Here we briefly describe the so-called column-wise butterfly representation [31] [34].

At level  $l = 0$ , the interpolation matrices  $V_{\tau, \nu}$  are explicitly formed, and at level  $l > 0$ , they are represented in a nested fashion. To see this, consider a node pair  $(\tau, \nu)$  at level  $l > 0$ , and let  $\nu_1$  and  $\nu_2$ , and  $p_\tau$  be the children of  $\nu$  and parent of  $\tau$ , respectively. From (5.1), we have

$$\begin{aligned} K(\tau, \nu) &= [K(\tau, \nu_1), K(\tau, \nu_2)] \\ &\approx [K(\tau, \bar{\nu}_1), K(\tau, \bar{\nu}_2)] \begin{bmatrix} V_{p_\tau, \nu_1} & \\ & V_{p_\tau, \nu_2} \end{bmatrix} \\ &\approx K(\tau, \bar{\nu})W_{\tau, \nu} \begin{bmatrix} V_{p_\tau, \nu_1} & \\ & V_{p_\tau, \nu_2} \end{bmatrix}. \end{aligned} \quad (5.2)$$

Here  $W_{\tau, \nu}$  and  $\bar{\nu}$  are the interpolation matrix and skeleton columns from the ID of  $[K(\tau, \bar{\nu}_1), K(\tau, \bar{\nu}_2)]$ , respectively. This allows representing  $V_{\tau, \nu}$  as

$$V_{\tau, \nu} = W_{\tau, \nu} \begin{bmatrix} V_{p_\tau, \nu_1} & \\ & V_{p_\tau, \nu_2} \end{bmatrix}, \quad (5.3)$$

where we refer  $W_{\tau, \nu}$  as the transfer matrix. We will select  $O(r_{\tau, \nu})$  proxy rows  $\hat{\tau} \subset \tau$  to approximately compute  $V_{\tau, \nu}$  and  $W_{\tau, \nu}$  via ID:

$$K(\hat{\tau}, \nu) \approx K(\hat{\tau}, \bar{\nu})V_{\tau, \nu}, \quad l = 0, \quad (5.4)$$

$$[K(\hat{\tau}, \bar{\nu}_1), K(\hat{\tau}, \bar{\nu}_2)] \approx K(\hat{\tau}, \bar{\nu})W_{\tau, \nu}, \quad 0 < l < L. \quad (5.5)$$

And we do not select proxy rows at level  $L$ .

In the current work, we select the proxy rows as follows. Consider a subblock  $(\tau, \nu)$  at level  $0 < l < L$ . Let  $\mathbf{x}^i$  be the geometrical centroid in the computational domain corresponding to the row index  $i$ . Let  $s_i$  be the index set consisting of the indices of neighboring centroids of the centroid  $\mathbf{x}^i$ , where  $i \in \bar{\nu}_1 \cup \bar{\nu}_2$ ; for example,  $s_i$  may be the index set of all indices for centroids that are within a  $10 * h$  distance of  $\mathbf{x}^i$ , where  $h$  is the mesh size. Let  $f_\tau$  be the index set consisting of  $\chi_1 |\bar{\nu}_1 \cup \bar{\nu}_2|$  indices for uniformly selected centroids near the boundary of the subdomain  $\tau$  with an oversampling parameter  $\chi_1$ , and let  $g_\tau$  denote the index set consisting of  $\chi_2 |\bar{\nu}_1 \cup \bar{\nu}_2|$  indices for randomly selected centroids inside the subdomain

$\tau$  with an oversampling parameter  $\chi_2$ , where  $|\bar{v}_1|$  indicates the cardinality of the set  $\bar{v}_1$ . Accordingly, we choose the proxy rows as

$$\hat{\tau} = \left( \bigcup_{i \in \bar{v}_1 \cup \bar{v}_2} s_i \right) \cap \tau \cup f_\tau \cup g_\tau. \quad (5.6)$$

Numerically, we take the oversampling parameters  $\chi_1 = \chi_2 = 3$ . Here, the selection criteria for proxy rows and oversampling parameters are empirical, determined by the properties of the kernel function and numerical accuracy tests. This ensures that the error introduced by the butterfly decomposition is significantly smaller than that from the numerical Hadamard integrator itself. The proxy rows can greatly reduce the complexity. With all the interpolation and transfer matrices computed, the butterfly representation of  $K(\Omega_1, \Omega_2)$  is:

$$K(\Omega_1, \Omega_2) \approx K^L W^L W^{L-1} \dots W^1 V^0. \quad (5.7)$$

Let  $v_1, v_2, \dots$ , and  $v_{2^{L-l}}$  denote the nodes at level  $L-l$  of  $\mathcal{T}_2$ , and  $\tau_1, \tau_2, \dots$ , and  $\tau_{2^l}$  denote the nodes at level  $l$  of  $\mathcal{T}_1$ . The interpolation factor  $V^0$ , the transfer factors  $W^l$  for  $l = 1, \dots, L$ , and the skeleton factor  $K^L$  are assembled in the following way,

$$V^0 = \text{diag} \left( V_{\tau, v_1}, \dots, V_{\tau, v_{2^L}} \right), \quad (\tau, v_i) \text{ at level } l = 0, \quad (5.8)$$

$$K^L = \text{diag} \left( K(\tau_1, \bar{v}), \dots, K(\tau_{2^L}, \bar{v}) \right), \quad (\tau_i, v) \text{ at level } l = L, \quad (5.9)$$

$$W^l = \text{diag} \left( W_{\tau_1}, \dots, W_{\tau_{2^{l-1}}} \right), \quad l = 1, \dots, L, \quad (5.10)$$

$$W_{\tau_i} = \begin{bmatrix} \text{diag} \left( W_{\tau_i^1, v_1}, \dots, W_{\tau_i^1, v_{2^{L-l}}} \right) \\ \text{diag} \left( W_{\tau_i^2, v_1}, \dots, W_{\tau_i^2, v_{2^{L-l}}} \right) \end{bmatrix}, \quad (\tau_i^{\{1,2\}}, v_j) \text{ at level } l, \quad (5.11)$$

where  $\tau_i^1$  and  $\tau_i^2$  denote the children of  $\tau_i$ .

## 5.2 Hierarchically Off-Diagonal Butterfly (HODBF) Representation

When  $\Omega_1 = \Omega_2$ , the resulting integral kernels  $K(\Omega_1, \Omega_1)$  have singular diagonal terms. We use the hierarchically off-diagonal butterfly (HODBF) [33, 34] representation to resolve such singularities.

We still use the  $L$ -level binary tree  $\mathcal{T}_1$  for  $\Omega_1$  as in the last subsection. For any two siblings  $\tau_1$  and  $\tau_2$  with parent  $\tau$  at level  $l$  of  $\mathcal{T}_1$ , we can directly extract two  $(L-l)$ -level subtrees  $\mathcal{T}_{\tau_1}$  and  $\mathcal{T}_{\tau_2}$  rooted at  $\tau_1$  and  $\tau_2$  from  $\mathcal{T}_1$ , respectively, due to the nested structure of the binary tree. Next, we construct the  $(L-l)$ -level butterfly representation for  $K(\tau_1, \tau_2)$ , where  $\tau_1 \cap \tau_2 = \emptyset$ . Finally, there are  $2^l$  butterfly representations at each level  $l = 1, 2, \dots, L$ . The subblocks  $K(\tau, \tau)$  for  $\tau$  at level  $L$  are kept as the dense blocks. Such a representation is called the hierarchically off-diagonal butterfly (HODBF) representation.

## 5.3 The Complexity and Memory Usage

We analyze the complexity and memory usage of IDBF and HODBF for  $K = U_i^\omega$  for  $i = 1, 2, 3, 4$  and  $\check{U}_j^t$  for  $j = 1, 2, 3, 4$ . Using the Chebyshev interpolation and partial summation, we can evaluate each element in those kernels with  $O(1)$  complexity [34].

For IDBF of  $K(\Omega_1, \Omega_2) \in \mathbb{C}^{O(n) \times O(n)}$ , we see that, in (5.8)-(5.11),  $V^0$  and  $K^L$  contain  $2^L$  diagonal blocks each with  $O(r_{\tau, v})$  non-zeros, and  $W^l$  contains  $2^L$  blocks  $W_{\tau, v}$  each with  $O(r_{\tau, v}^2)$  non-zeros. After selecting proxy rows, we will evaluate  $O(n)$  IDs of size

$O(r_{\tau,v}) \times O(r_{\tau,v})$  for  $O(\log(n))$  levels, resulting in  $O(n \log(n))$  complexity and memory usage if  $O(r_{\tau,v}) = O(1)$ , analogous to the classic result of the butterfly algorithm.

However, for the kernel with singularity near the interface in our setting, we follow [34] to analyze the sub-blocks near the interface separately. Letting  $l_m = \frac{L}{2}$  denote the middle level of the butterfly, we can show that among the  $O(n)$  subblocks  $K(\tau, v)$  at each level  $l$ , there are  $O(2^{|l-l_m|/2} n^{1/4})$  subblocks representing interactions between adjacent or close-by geometric subdomains. The non-constant ranks of these subblocks scale as  $r_{\tau,v} = O(2^{-|l-l_m|/2} n^{1/4})$ , which are dominated by the interface degrees of freedom (DOFs) between the two subdomains. Moreover, each non-constant-rank subblock requires  $O(r_{\tau,v}^2)$  memory usage and matrix entry computation and  $O(r_{\tau,v}^3)$  ID cost. Thus, they require

$$\sum_l r_{\tau,v}^2 O(2^{|l-l_m|/2} n^{1/4}) = O(n^{3/4}) \leq O(n) \quad \text{memory usage} \quad (5.12)$$

and

$$\sum_l r_{\tau,v}^3 O(2^{|l-l_m|/2} n^{1/4}) = O(n) \quad \text{complexity}. \quad (5.13)$$

Hence, we can still state that the memory usage and complexity of the IDBF for  $K \in \mathbb{C}^{O(n) \times O(n)}$  are  $O(n \log(n))$ . Due to the  $O(n \log(n))$  nonzeros, we immediately conclude that the complexity of the application of the IDBF matrix is  $O(n \log(n))$ .

For HODBF of  $K(\Omega_1, \Omega_1) \in \mathbb{C}^{O(N) \times O(N)}$ , letting  $L = O(\log(N))$  denote the level of the binary tree  $\mathcal{T}_1$ , we follow the DOFs analysis in [43] to sum up the complexity and memory usage of the  $(L-l)$ -th level butterfly for all levels  $l$  and the  $2^L$  dense blocks at level  $L$ ,

$$2^L O(1) + \sum_{l=1}^L 2^l O(N/2^l \log(N/2^l)) = O(N \log^2(N)). \quad (5.14)$$

That is, the memory usage and complexity of the construction and application of the HODBF for  $K(\Omega_1, \Omega_1) \in \mathbb{C}^{O(N) \times O(N)}$  are  $O(N \log^2(N))$ .

## 5.4 Butterfly-Compressed Eulerian Hadamard Integrators

For Case 1, starting from Algorithm 1 and using IDBF and HODBF, we precompute the low-rank representations of frequency-dependent kernels  $\{U_i^{\omega_k}\}$  and corresponding time-dependent kernels  $\{\check{U}_i^{t_\ell}\}$  and apply these compressed kernels in each time step, resulting in the butterfly-compressed TFT Hadamard integrator (Algorithm 2) with quasi-linear memory usage and computational complexity.

### Algorithm 2: Butterfly-compressed TFT Eulerian Hadamard integrator for Case 1

1. Uniformly discretize the computational domain  $\Omega$  into a wave-resolution-satisfying uniform grid; partition  $\Omega$  into  $N_I$  non-overlapping source regions  $\{\Omega_S^\ell, \ell = 1, 2, \dots, N_I\}$  and designate the corresponding receiver regions to be  $\{\Omega_R^\ell, \ell = 1, 2, \dots, N_I\}$ , set the initial time  $T_0$ , the large time step  $\Delta T < \tau_{\min}$  for updating the initial condition, the small time step  $\Delta t = \frac{\Delta T}{N_T}$  for wave resolution, and an ending time  $T_{\text{end}}$  to ensure that the waves do not reach the computational boundary; determine the bandwidth  $B$  according to the initial conditions, set the frequency step  $\Delta\omega = \frac{B}{N_\omega}$ , where  $N_\omega = O(B)$ , and generate  $\omega_k = (k + \frac{1}{2})\Delta\omega$  for  $k = 0, 1, 2, \dots, N_\omega - 1$ ; precompute and compress Hadamard ingredients (4.2) in all sub-region pairs.

2. For each source region  $\Omega_S^\ell$ , divide the corresponding receiver region  $\Omega_R^\ell$  into several subregions that are similar in size to the source region, including a subregion that coincides with the source region and several adjacent subregions; compute the IDBF and HODBF representations of

$$U_s^{\omega_k}, s = 1, 2, 3, 4, 1 \leq k \leq N_\omega - 1 \quad \text{and} \quad \check{U}_s^{n\Delta t}, s = 1, 2, 3, 4, 1 \leq n \leq N_T.$$

3. Initialize  $u(T_0, \mathbf{x}) = u^1(\mathbf{x})$ ,  $u_t(T_0, \mathbf{x}) = u^2(\mathbf{x})$ ; set time  $T = T_0$  and loop variable  $p = 0$ .  
 4. For the current time step with  $T = T_0 + p\Delta T$ :

- (a) compute finite-difference derivatives of  $u(T, \mathbf{x})$  and  $u_t(T, \mathbf{x})$  and construct vectors  $f_1, f_2, \dots, f_8$ .
- (b) determine a subregion  $\Omega_p$  of  $\Omega$  that contains the region of influence of current data by extending each direction of the non-zero region of the solution  $u(T, \mathbf{x})$  outward by  $\max c\Delta T$ ; designate the source regions intersecting with  $\Omega_p$  as the regions to be updated.
- (c) for all the source regions  $\Omega_S^\ell$  to be updated:
  - i apply the IDBF matrices and HODBF matrices to obtain the corresponding frequency- and time-sampling functions

$$\hat{u}_i(\omega_k), i = 1, 2, 3, 4, 1 \leq k \leq N_\omega - 1, \quad \text{and}$$

$$\check{u}_i(n\Delta t), i = 1, 2, 3, 4, 1 \leq n \leq N_T.$$

- ii update wavefields  $u(T + n\Delta t, \Omega_S^\ell)$  and  $u_t(T + n\Delta t, \Omega_S^\ell)$  via (4.33) and (4.34).

- (d) set to zero the wave fields within the source regions that do not require updating.

5. Update  $T = T_0 + (p + 1)\Delta T$ . If  $T < T_{end}$ , then  $p \leftarrow p + 1$  and go to Step 4; else, stop.

Equipped with butterfly representations, we construct the frequency-domain and time-domain kernels in the precomputation step and recursively apply them during time marching. The computational complexity and memory usage of constructing and applying each kernel are reduced from  $O(B^4)$  to  $O(B^2 \log^2(B))$ , resulting in the Eulerian Hadamard integrator with overall complexity and memory usage of  $O(B^3 \log^2(B))$ . Although we divide  $\Omega$  into  $N_I$  parts,  $N_I = O(1)$  which only depends on the medium; consequently, such partition does not affect the overall complexity estimate.

Further, assume that we solve the wave equation with  $m_1$  different initial conditions simultaneously. In this case, we only construct the  $O(B)$  kernels once, and the computational cost is

$$O(B)O(B^2 \log^2(B)) \doteq O(C_1 B^3 \log^2(B));$$

applying all these kernels to  $m_1$  initial conditions, the computational cost is  $O(m_1 C_2 B^3 \log^2(B))$ ; we use the fast Fourier transform to obtain the time-domain solutions on  $O(B^3)$  mesh points, with a cost of  $O(m_1 B^3 \log(B))$ . Since we only need to update the initial conditions for  $\frac{T_{end}}{\Delta T} = O(1)$  times, the overall computational complexity is

$$O\left((C_1 + \frac{T_{end}}{\Delta T} m_1 C_2) B^3 \log^2(B)\right).$$

For the butterfly algorithm,  $C_1$  depends on the accuracy of the representation and  $r_{\tau,v}^3$  from the IDs. In any case,  $C_1$  is far greater than  $C_2$ ; thus, when solving different Cauchy problems in the same medium, the efficiency of the algorithm will be more significant. On the other hand, even if IDBF and HODBF need to be recalculated at each step due to limited computing resources, the overall computational complexity remains quasi-linear.



The quasi-linear memory usage makes parallel computing feasible. Since there are  $O(B)$  kernels to compress, we do not embed parallel computing into the decomposition of each kernel matrix. Instead, we use the parallel toolbox in Matlab to simultaneously calculate the butterfly representations of  $O(B)$  kernel matrices.

For Case 2, we first use (3.33) to approximate the time-domain solution in the interval  $0 < t \leq \Delta T$  and then apply **Algorithm 2** to solve the wave equation (4.35)–(4.37) in the interval  $\Delta T < t \leq T_{\text{end}}$ . Subsequently, we perform the Fourier transform (3.35) to obtain the frequency-domain solution. This leads to the butterfly-compressed TFTF Eulerian Hadamard integrator. Here, we only perform an additional Fourier transform, which means that the computational complexity and memory usage of the butterfly-compressed TFTF Eulerian Hadamard integrator remain  $O(C_1 B^3 \log^2(B))$ .

Compared to the TFT method, which yields  $O(B^3)$  unknowns in the time-space domain, the TFTF method obtains  $O(B^3)$  unknowns in the frequency-space domain. Namely, both Eulerian Hadamard integrators achieve a quasi-linear computational complexity.

## 6 Numerical Examples

This section provides several numerical examples to demonstrate the accuracy and efficiency of the proposed butterfly-compressed Hadamard integrators by solving wave equations in time and frequency domains. All computations were performed on a computer equipped with 512GB of RAM and two 28-core Intel Xeon 6258R processors.

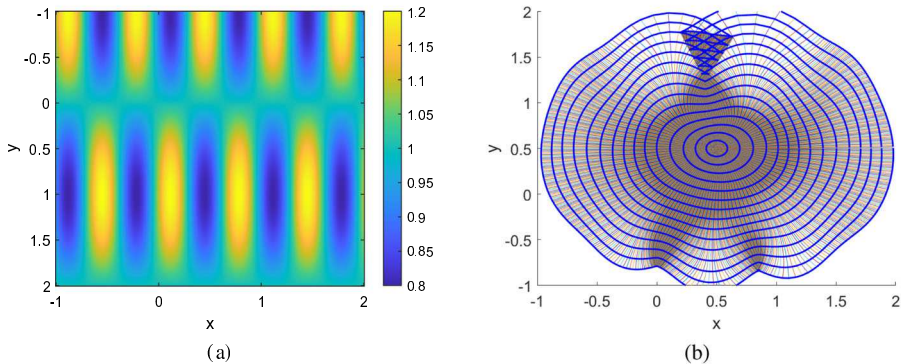
Because exact solutions for the wave equations are not available in general, we solve the wave equation using the FDTD method [54] for time-domain “reference” solutions and a finite-difference based direct solver [20] for frequency-domain “reference” solutions, where the perfectly-matched layer (PML) absorbing boundary conditions are imposed [9, 13, 16], and we always keep the number of points per wavelength (PPW) greater than 40 to reduce the dispersion error in the two direct methods. For the proposed Hadamard integrator, we set  $\text{PPW} \sim 5$  with respect to the artificial bandwidth  $B$ .

### 6.1 Sinusoidal Model

We introduce a sinusoidal model with the following setup.

- $\rho = \frac{1}{(1 + 0.2 \sin(3\pi(x + 0.05)) \sin(0.5\pi y))^2}$ ,  $\nu = 1$  such that  $c = 1 + 0.2 \sin(3\pi(x + 0.05)) \sin(0.5\pi y)$ .
- $B = 64\pi$ ,  $h = \frac{1}{200}$ ,  $\Delta\omega = \frac{\pi}{2}$ , and  $\Delta t = \frac{1}{192}$ .
- $\Omega = [-1, 2] \times [-1, 2]$ ; the sizes of  $\Omega_S^\ell$  and  $\Omega_R^\ell$  are  $0.2 \times 0.2$  and  $0.6 \times 0.6$ , respectively;  $\Delta T = \frac{1}{8}$ .
- Orders of Chebyshev interpolations of Hadamard ingredients in the four variables  $x_0$ ,  $y_0$ ,  $x$ , and  $y$  are 11, 11, 13, and 13, respectively.
- Tolerance used in the interpolative decomposition:  $\text{tol} = 10^{-9}$ .

We show the velocity model, some rays, and wavefronts in Fig. 2, where the rays and wavefronts are obtained by solving the eikonal equation using the method of characteristics. Caustics occur when rays intersect and wavefronts fold. Note that in the current domain, caustics develop only along the  $y$  direction. The frequency-domain fast Huygens sweeping method developed in [37, 40] can also be used to compute the point-source wave field in this medium by propagating waves layer by layer along the  $y$  direction.



**Fig. 2** Sinusoidal model. **(a)** The velocity; **(b)** Rays and wavefronts with source  $x_0 = [0.5, 0.5]$ . The thick blue lines represent equal-time wavefronts (traveltime contours) with the contour interval equal to 0.1, and thin colored lines represent rays with different take-off angles

**Table 1** The relative  $L^2$  and  $L^\infty$  errors of TFT solutions for Sinusoidal model at  $t = 0.5$

$\beta$	8	16	32
Relative $L^2$ error	$3.22e-2$	$1.97e-2$	$3.77e-2$
Relative $L^\infty$ error	$3.08e-2$	$1.88e-2$	$3.64e-2$

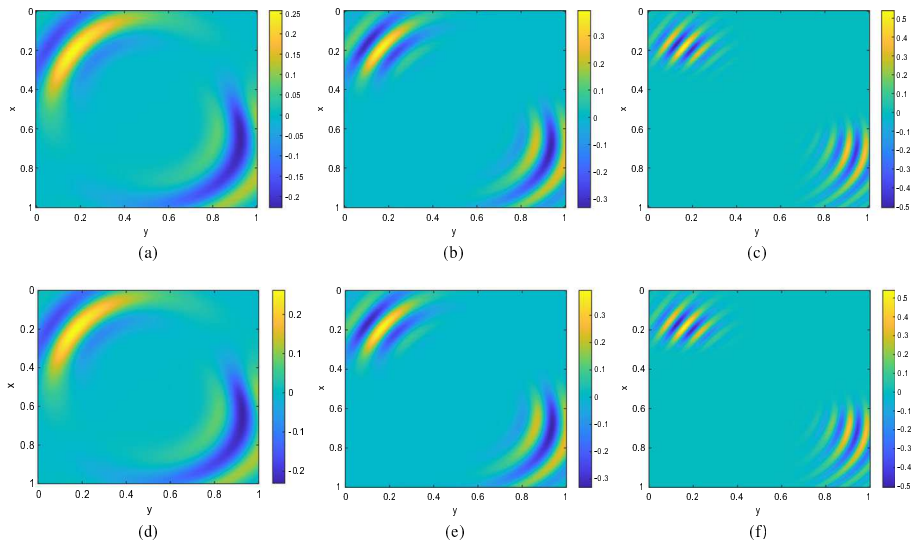
### 6.1.1 TFT Method for Case 1

For Case 1, we take the initial conditions as

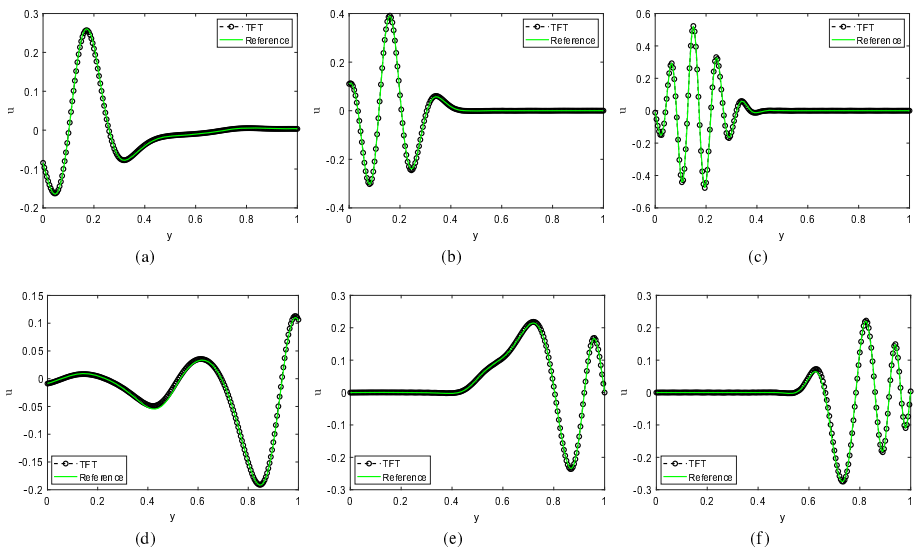
$$\begin{aligned}
 u(0, x, y) &= \sin\left(\beta\pi \frac{x+y-1}{\sqrt{2}}\right) \exp\left(-100((x-0.5)^2 + (y-0.5)^2)\right), \\
 u_t(0, x, y) &= 0,
 \end{aligned} \tag{6.1}$$

where  $\beta = 8, 16$  and  $32$ , respectively, and we utilize the Hadamard integrator to simultaneously solve these three Cauchy problems. Figure 3 shows the comparisons between the TFT solutions and the reference solutions in  $[0, 1] \times [0, 1]$  at  $t = 0.5$ . In Fig. 4, we further compare the solutions along some lines. Table 1 shows the relative  $L^2$  and  $L^\infty$  errors between the TFT solutions and the reference solutions.

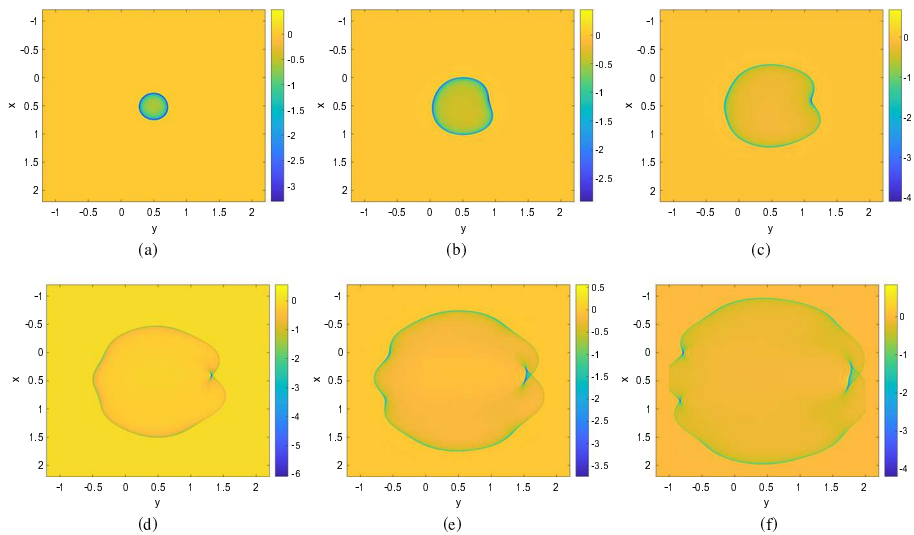
Here, we observe that compared with the case of  $\beta = 16$ , the TFT method shows larger errors for the cases of  $\beta = 8$  and  $\beta = 32$ . The reason for the case of  $\beta = 8$  is that the asymptotic error of the Hadamard integrator dominates in the low-frequency band, which is typical of the error behavior for microlocal-analysis based asymptotic methods [34]; on the other hand, as  $\beta$  gradually increases to reach or even exceed the Nyquist frequency  $B/2$ , the time-domain solution reconstructed through frequency-domain samplings will no longer be accurate, as indicated by the Shannon-Nyquist Sampling theorem. Similar error performance will be observed in subsequent numerical examples. We will further discuss the error behaviors and convergence results in detail in Sect. 6.4.



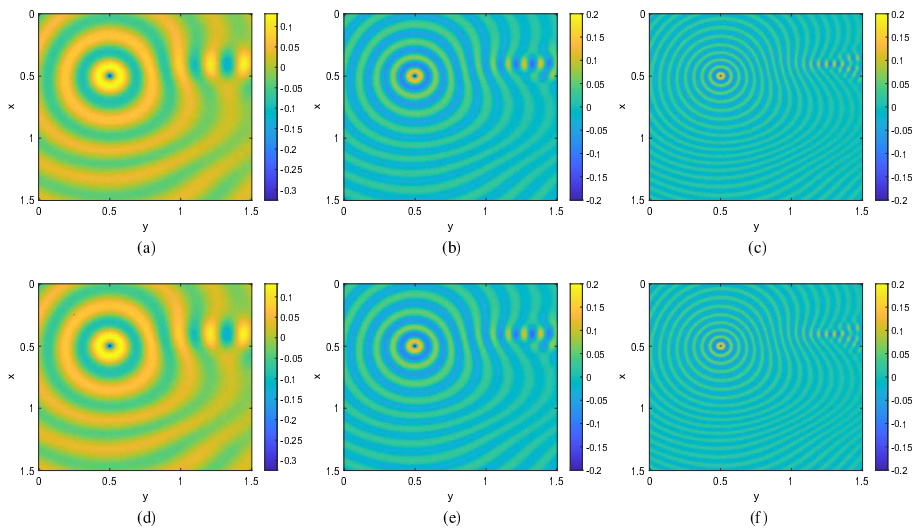
**Fig. 3** Sinusoidal model.  $u(t, x)$ ,  $t = 0.5$ : (a) TFT solution with  $\beta = 8$ ; (b) TFT solution with  $\beta = 16$ ; (c) TFT solution with  $\beta = 32$ ; (d) reference solution with  $\beta = 8$ ; (e) reference solution with  $\beta = 16$ ; (f) reference solution with  $\beta = 32$



**Fig. 4** Sinusoidal model. Slices of time-domain wave fields  $u(t, x)$  when  $t = 0.5$ : (a) a slice at  $x = 0.2$  with  $\beta = 8$ ; (b) a slice at  $x = 0.2$  with  $\beta = 16$ ; (c) a slice at  $x = 0.2$  with  $\beta = 24$ ; (d) a slice at  $y = 0.85$  with  $\beta = 8$ ; (e) a slice at  $y = 0.85$  with  $\beta = 16$ ; (f) a slice at  $y = 0.85$  with  $\beta = 24$



**Fig. 5** Sinusoidal model. Time-domain point-source wave fields  $u(t, \mathbf{x})$ . (a)  $T = 0.25$ ; (b)  $T = 0.5$ ; (c)  $T = 0.75$ ; (d)  $T = 1$ ; (e)  $T = 1.25$ ; (f)  $T = 1.5$



**Fig. 6** Sinusoidal model.  $\hat{u}(\omega, \mathbf{x})$ . (a) TFTF solution with  $\omega = 8\pi$ ; (b) TFTF solution with  $\omega = 16\pi$ ; (c) TFTF solution with  $\omega = 32\pi$ ; (d) reference solution with  $\omega = 8\pi$ ; (e) reference solution with  $\omega = 16\pi$ ; (f) reference solution with  $\omega = 32\pi$

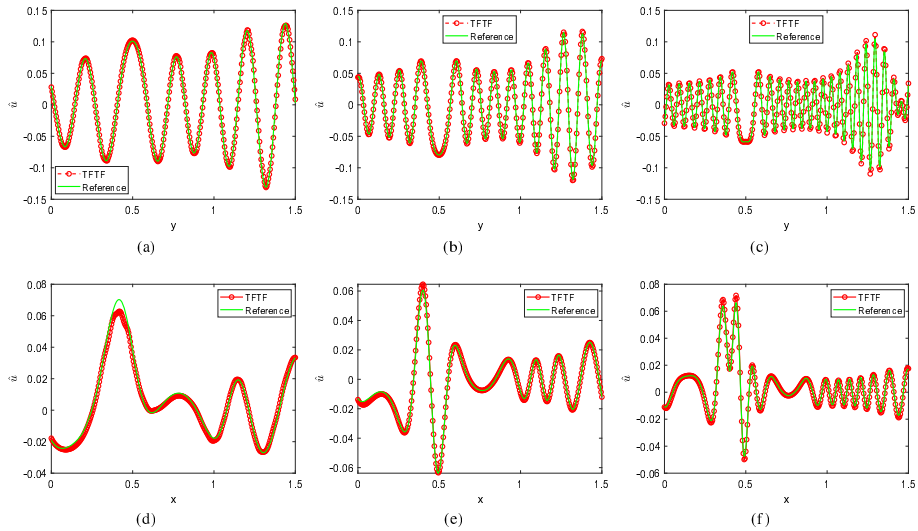
### 6.1.2 TFTF Method for Case 2

For Case 2, taking  $\mathbf{z} = [0.5, 0.5]$  and  $T_{end} = 1.5$ , we obtain the time-domain point-source wave fields  $u(t, \mathbf{x})$ . As shown in Fig. 5, we can clearly observe the occurrence of caustics.

Further applying Fourier transform in time, we obtain the frequency-domain point-source wave fields  $\hat{u}(\omega, \mathbf{x})$  with different angular frequencies  $\omega$ . In Fig. 6, we compare the TFTF solution with the reference solution for  $\omega = 8\pi$ ,  $16\pi$  and  $32\pi$ , respectively, where  $\mathbf{x} \in$

**Table 2** The  $L^2$  and  $L^\infty$  errors of TFTF solutions for Sinusoidal model

$\omega$	$8\pi$	$16\pi$	$32\pi$
Relative $L^2$ error	$3.21e-2$	$2.48e-2$	$5.88e-2$
Relative $L^\infty$ error	$3.81e-2$	$2.79e-2$	$3.84e-2$

**Fig. 7** Sinusoidal model. Slices of frequency-domain point source wave fields  $\hat{u}(\omega, \mathbf{x})$ . (a) a slice at  $x = 0.4$  with  $\omega = 8\pi$ ; (b) a slice at  $x = 0.4$  with  $\omega = 16\pi$ ; (c) a slice at  $x = 0.4$  with  $\omega = 32\pi$ ; (d) a slice at  $y = 1.4$  with  $\omega = 8\pi$ ; (e) a slice at  $y = 1.4$  with  $\omega = 16\pi$ ; (f) a slice at  $y = 1.4$  with  $\omega = 32\pi$ 

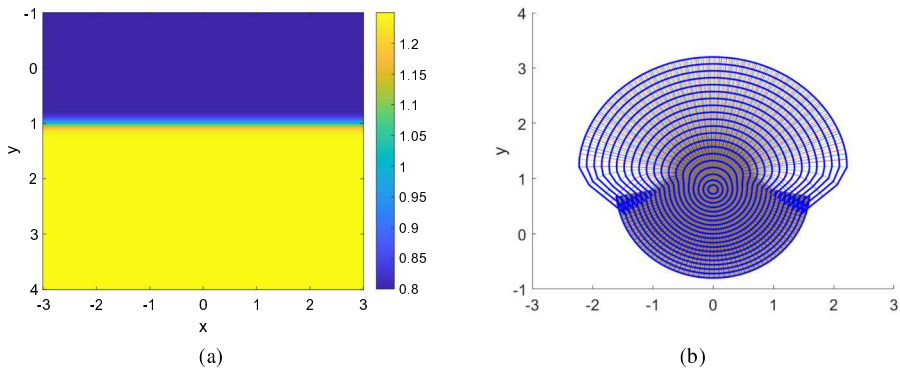
$[0, 1.5] \times [0, 1.5]$ . Table 2 shows the relative  $L^2$  and  $L^\infty$  errors between the TFTF solutions and the reference solutions.

Figure 7 shows the comparisons of the two solutions along lines that traverse through the caustic region for various  $\omega$ . Although they match well, it can be observed that the errors at  $\omega = 8\pi$  and  $32\pi$  are slightly larger than those at  $\omega = 16\pi$ , and we will explain this phenomenon in Sect. 6.4.

## 6.2 Smoothed Heaviside Model

We introduce a  $y$ -dependent velocity model  $c$ , which is analogous to a scaled, smoothed, and shifted Heaviside function.

- $\rho = \left( \frac{1 + e^{-20(y-1)}}{1.25 + 0.8e^{-20(y-1)}} \right)^2$  and  $\nu = 1$  such that  $c = 0.8 + \frac{0.45}{1 + e^{-20(y-1)}}$ .
- $B = 48\pi$ ,  $h = \frac{1}{160}$ ,  $\Delta\omega = \frac{\pi}{2}$ , and  $\Delta t = \frac{1}{144}$ .
- $\Omega = [-3, 3] \times [-1, 4]$ ; the sizes of  $\Omega_S^\ell$  and  $\Omega_R^\ell$  are  $0.2 \times 0.2$  and  $0.6 \times 0.6$ , respectively;  $\Delta T = \frac{1}{8}$ .
- Orders of Chebyshev interpolations of Hadamard ingredients in the four variables  $x_0$ ,  $y_0$ ,  $x$ , and  $y$  are 11, 11, 13, and 13, respectively.
- Tolerance used in interpolative decomposition:  $\text{tol} = 10^{-9}$ .



**Fig. 8** Heaviside model. **(a)** The velocity; **(b)** Rays and wavefronts with source  $x_0 = [0, 0.8]$ . The thick blue lines represent equal-time wavefronts (traveltime contours) with the contour interval equal to 0.1, and thin colored lines represent rays with different take-off angles

**Table 3** The relative  $L^2$  and  $L^\infty$  errors of TFT solutions for the Heaviside model at  $t = 0.5$

$\beta$	8	16	32
Relative $L^2$ error	$2.76e-2$	$2.38e-2$	$4.62e-2$
Relative $L^\infty$ error	$4.05e-2$	$2.81e-2$	$4.92e-2$

We show the velocity model, some rays, and wavefronts in Fig. 8. The rapid change in velocity near  $y = 1$  results in overturning rays and caustics. The FHS method, which propagates the wave field along a specific spatial direction, is no longer valid due to these overturning rays. However, the proposed Hadamard integrator, which propagates the wave field along the time direction, can naturally handle such caustics.

### 6.2.1 TFT Method for Case 1

For Case 1, we take the initial conditions as

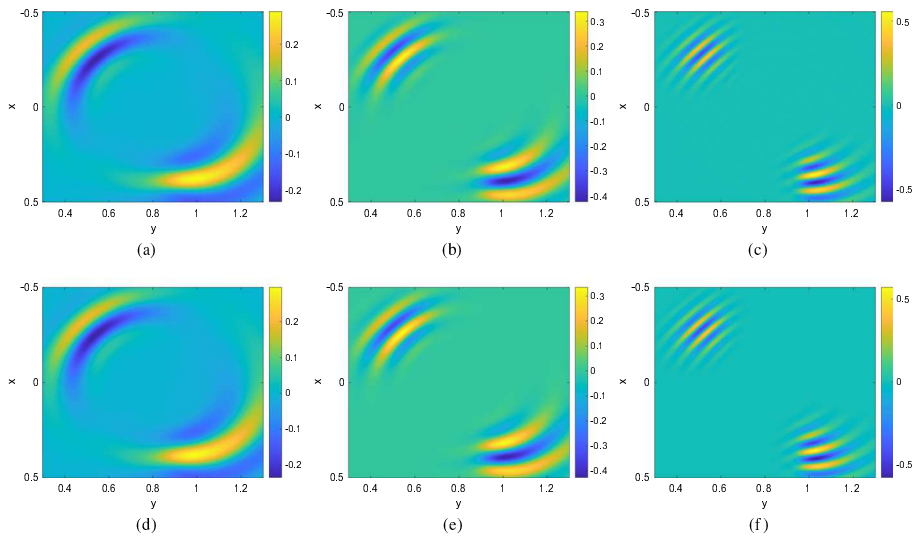
$$u(0, x, y) = \sin\left(\beta\pi \frac{x+y-1}{\sqrt{2}}\right) \exp(-100(x^2 + (y-0.8)^2)), \quad u_t(0, x, y) = 0, \quad (6.2)$$

where  $\beta = 8, 16, 24$ , respectively, and we utilize the Hadamard integrator to simultaneously solve these three Cauchy problems. Figure 9 shows the comparisons between the TFT solutions and the reference solutions at  $t = 0.5$ . Figure 10 further compares the solutions along some lines. Table 3 shows the relative  $L^2$  and  $L^\infty$  errors between the TFT solutions and the reference solutions.

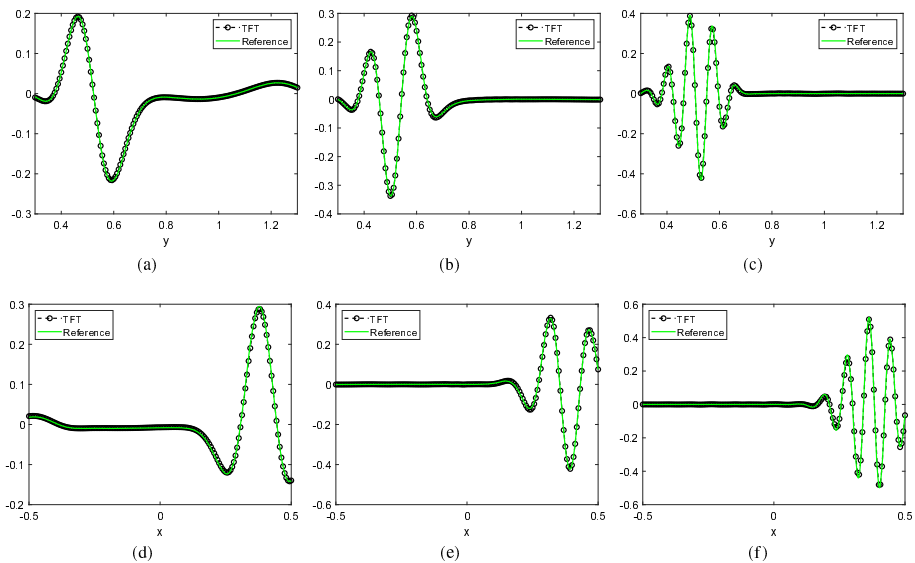
### 6.2.2 TFTF Method for Case 2

For Case 2, taking  $z = [0, 0.8]$  and  $T_{\text{end}} = 2$ , we obtain the time-domain point-source wave fields  $u(t, x)$ . As shown in Fig. 11, we can clearly observe the caustics induced by the overturning rays.

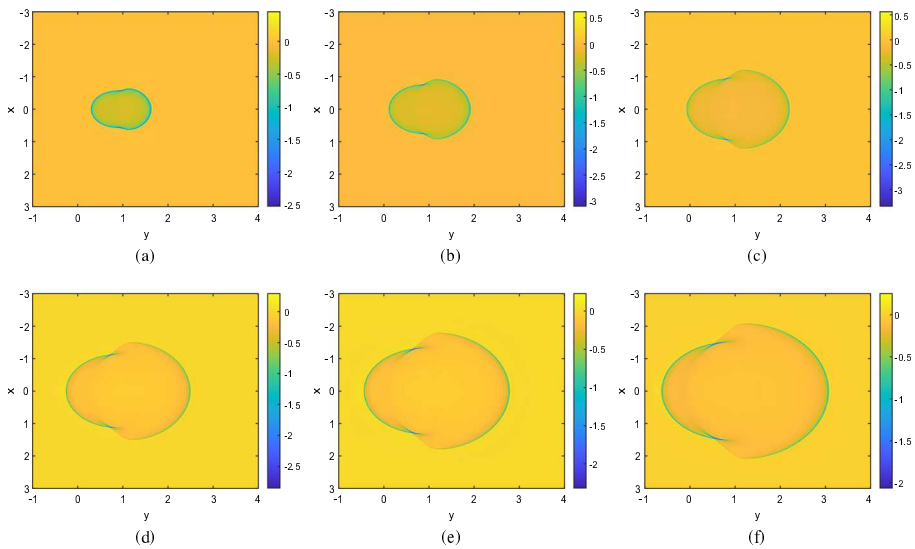
Further applying the Fourier transform, we obtain the frequency-domain point-source wave fields  $\hat{u}(\omega, x)$  with different angular frequencies  $\omega$ . In Fig. 12, we compare the TFTF solution with the reference solution for  $\omega = 8\pi, 16\pi$ , and  $24\pi$ , respectively, where  $x \in$



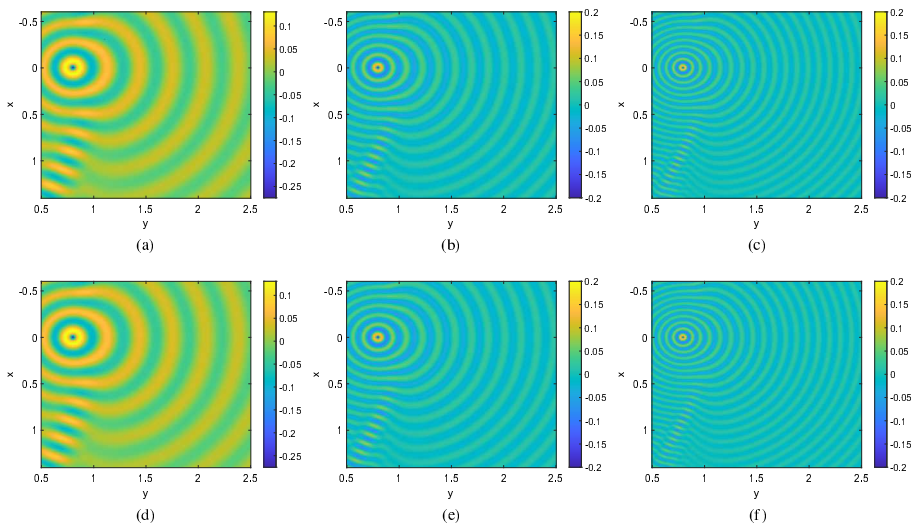
**Fig. 9** Heaviside model  $u(t, \mathbf{x})$ ,  $t = 0.5$ : (a) TFT solution with  $\beta = 8$ ; (b) TFT solution with  $\beta = 16$ ; (c) TFT solution with  $\beta = 32$ ; (d) reference solution with  $\beta = 8$ ; (e) reference solution with  $\beta = 16$ ; (f) reference solution with  $\beta = 32$



**Fig. 10** Heaviside model, Slices of time-domain wave fields  $u(t, \mathbf{x})$  when  $t = 0.5$ : (a) a slice at  $x = -0.3$  with  $\beta = 8$ ; (b) a slice at  $x = -0.3$  with  $\beta = 16$ ; (c) a slice at  $x = -0.3$  with  $\beta = 24$ ; (d) a slice at  $y = 1$  with  $\beta = 8$ ; (e) a slice at  $y = 1$  with  $\beta = 16$ ; (f) a slice at  $y = 1$  with  $\beta = 24$



**Fig. 11** Heaviside model. Time-domain point source wave fields  $u(t, \mathbf{x})$ . (a)  $T = 0.75$ ; (b)  $T = 1$ ; (c)  $T = 1.25$ ; (d)  $T = 1.5$ ; (e)  $T = 1.75$ ; (f)  $T = 2$



**Fig. 12** Heaviside model.  $\hat{u}(\omega, \mathbf{x})$ . (a) TFTF solution with  $\omega = 8\pi$ ; (b) TFTF solution with  $\omega = 16\pi$ ; (c) TFTF solution with  $\omega = 24\pi$ ; (d) reference solution with  $\omega = 8\pi$ ; (e) reference solution with  $\omega = 16\pi$ ; (f) reference solution with  $\omega = 24\pi$

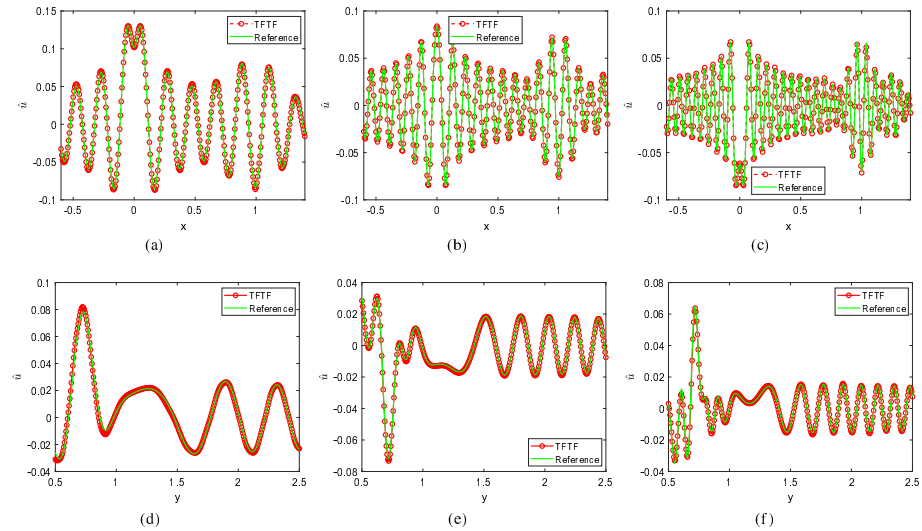
$[-0.6, 1.4] \times [0.5, 2.5]$ . We then show in Table 4 the relative  $L^2$  and  $L^\infty$  errors of the TFTF solutions compared with the reference solutions.

Figure 13 shows the comparisons of the two solutions along lines that traverse through the caustics region for various frequencies  $\omega$ . The TFTF solutions match with the reference solutions well.



**Table 4** The  $L^2$  and  $L^\infty$  errors of TFTF solutions for Heaviside model

$\omega$	$8\pi$	$16\pi$	$24\pi$
Relative $L^2$ error	$4.33e-2$	$2.98e-2$	$5.34e-2$
Relative $L^\infty$ error	$3.47e-2$	$3.07e-2$	$4.91e-2$

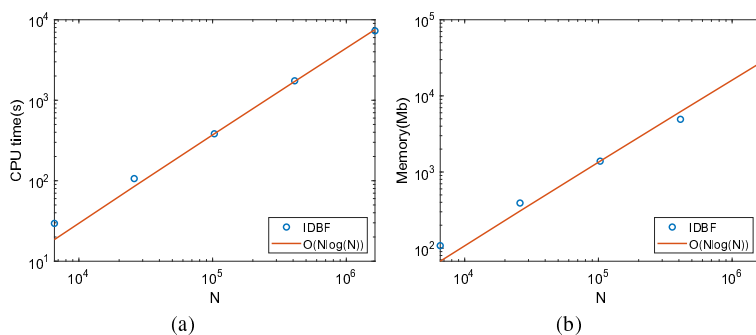
**Fig. 13** Heaviside model. Slices of frequency-domain point source wave fields  $\hat{u}(\omega, \mathbf{x})$ . (a) a slice at  $y = 0.75$  with  $\omega = 8\pi$ ; (b) a slice at  $y = 0.75$  with  $\omega = 16\pi$ ; (c) a slice at  $y = 0.75$  with  $\omega = 24\pi$ ; (d) a slice at  $x = 1.3$  with  $\omega = 8\pi$ ; (e) a slice at  $x = 1.3$  with  $\omega = 16\pi$ ; (f) a slice at  $x = 1.3$  with  $\omega = 24\pi$ 

### 6.3 Complexity Validation

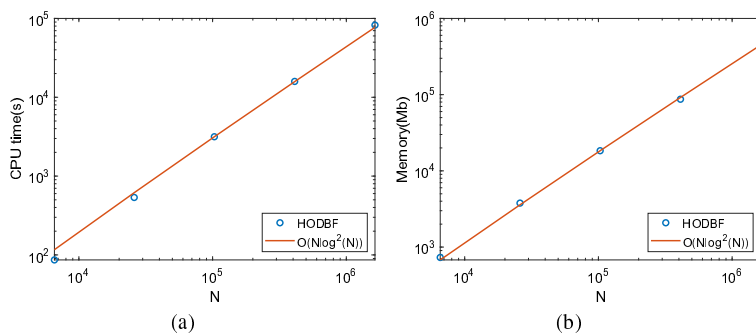
In this subsection, we validate the CPU and memory complexities of the proposed Eulerian Hadamard integrators. Here we use the Sinusoidal model, and we set the source region  $\Omega_1 = [-0.1, 0.1] \times [-0.1, 0.1]$  and one of the corresponding adjacent receiver subregions  $\Omega_2 = [-0.3, -0.1] \times [-0.1, 0.1]$ .

Since there exist singularities in our kernels, the compression error will increase as the grid becomes finer; however, we can choose proper tolerance  $\text{tol}$  and oversampling parameters  $\chi_1$  and  $\chi_2$  so that, even for an extremely fine mesh, such as  $h = \frac{1}{3200}$ , the maximum relative error is controlled to be smaller than  $10^{-4}$ , which can be neglected in comparison to other errors.

We set  $\text{tol} = 10^{-9}$  and  $\chi_1 = \chi_2 = 3$ , and we vary the frequency and cell count from  $\omega = 10\pi$  and  $N = 81^2$  to  $\omega = 160\pi$  and  $N = 1281^2$ , respectively. We sequentially construct the IDBF matrices of the kernels  $U_1^\omega(\Omega_1, \Omega_2)$ ,  $U_2^\omega(\Omega_1, \Omega_2)$ ,  $U_3^\omega(\Omega_1, \Omega_2)$  and  $U_4^\omega(\Omega_1, \Omega_2)$  and HODBF matrices of the kernels  $U_1^\omega(\Omega_1, \Omega_1)$ ,  $U_2^\omega(\Omega_1, \Omega_1)$ ,  $U_3^\omega(\Omega_1, \Omega_1)$  and  $U_4^\omega(\Omega_1, \Omega_1)$ . And we record the total CPU time and total memory usage of IDBF and HODBF, respectively. The CPU time and memory usage for constructing the IDBF and HODBF matrices are plotted, respectively, in Figs. 14 and 15, which validate our complexity estimates. Thus, we can state that for given tolerance  $\text{tol}$  and oversampling parameters  $\chi_1$  and  $\chi_2$ , the computational complexity and the CPU memory usage of the butterfly compression grow as  $O(N \log(N))$  for IDBF and  $O(N \log^2(N))$  for HODBF, respectively.



**Fig. 14** CPU time and memory usage of IDBF without parallel implementation



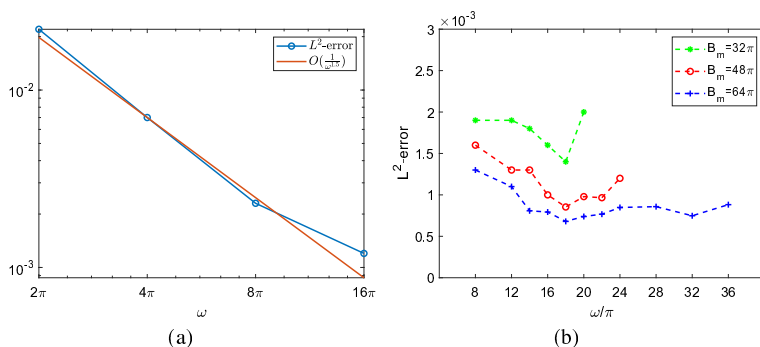
**Fig. 15** CPU time and memory usage of HODBF without parallel implementation

Due to its quasi-linear memory usage, the computations of IDBF and HODBF can be parallelized quite directly. Although there are  $O(B)$  kernels to compress, we will not embed parallelism into the algorithm so that we will only show the computational time of a sequential implementation of the algorithm. Nevertheless, simply using “parfor” with respect to  $\omega_k$  and  $t_\ell$  reduces the computational time of each kernel by roughly  $O(N_c)$  times, where  $N_c$  is the number of cores of the computer, and parallel implementation on the 56-core computer that we used here can improve the efficiency of the IDBF and HODBF in our numerical examples by roughly about 25 times.

## 6.4 Convergence Test

As mentioned earlier,  $B$  is an artificial frequency bandwidth, and  $\omega$  or  $\beta$  is the frequency parameter from the problem, both of which affect the accuracy of the Hadamard integrator.

In the TFT method for Case 1, we have assumed that the time-domain solution  $u(t, x)$  decays rapidly in the frequency domain, which means that increasing  $B$  will not improve the accuracy of the TFT solution. On the other hand, we regard  $\beta$  as a frequency parameter, but as  $\beta \rightarrow 0$ , the frequency-domain bandwidth of the solution does not tend to zero, due to the influence of components in the initial conditions ensuring compact support. Consequently, it is non-trivial to discuss the asymptotic behavior of the TFT method in terms of  $B$  or  $\beta$ , and thus we will not do that here. However, we need to choose an appropriately large  $B$  in conjunction with  $\beta$  to ensure the rapid decay beyond the bandwidth.



**Fig. 16** Convergence. (a):  $L^2$ -error with  $B = 64\pi$  and different  $\omega$ ; (b):  $L^2$ -error with different  $B$  and  $\omega$

In the TTFM method for Case 2, generating the time-domain Green's function from the Fourier summation of frequency-domain Green's functions yields an  $O(\frac{1}{B^{0.5}})$  asymptotic error according to (4.8). However, when using these time-domain solutions to generate the final frequency-domain solution, we observe an asymptotic behavior of  $O(\frac{1}{B})$ , which we believe is due to the fact that the TTFM method is somewhat analogous to reconstructing a signal using the Fourier transform. Additionally, the angular frequency also affects the accuracy. According to [34], taking the one-term Babich's ansatz  $\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x})$  as the frequency-domain Green's function introduces an  $O(\frac{1}{\omega^{1.5}})$  asymptotic error.

To verify the asymptotic behavior with respect to  $\omega$ , we take the angular frequency  $\omega = 2\pi, 4\pi, 8\pi$ , and  $16\pi$ , respectively, and evaluate the  $L^2$ -error with respect to the reference solution. As shown in Fig. 16a, when  $\omega$  is small, the asymptotic error of the one-term Babich's ansatz dominates, resulting in  $O(\frac{1}{\omega^{1.5}})$  convergence.

To verify the asymptotic behavior with respect to  $B$ , we consider the following setups in the Sinusoidal model:

1.  $B = 32\pi$ ,  $h = \frac{1}{100}$ , and  $\Delta t = \frac{1}{96}$ ;
2.  $B = 48\pi$ ,  $h = \frac{1}{160}$ , and  $\Delta t = \frac{1}{144}$ ;
3.  $B = 64\pi$ ,  $h = \frac{1}{200}$ , and  $\Delta t = \frac{1}{192}$ .

We further set  $T_{end} = 1$  and compute the frequency-domain solution in  $[0.2, 1.2] \times [0.2, 1.2]$ . The  $L^2$  errors for different  $B$  and  $\omega$  are shown in Fig. 16b.

For a fixed  $B$ , as  $\omega$  increases, the error initially decreases due to the dominance of the asymptotic error from Babich's ansatz. Subsequently, the error saturates due to the dominance of the asymptotic error caused by truncation of the frequency domain and the fixed  $B$ . Finally, after  $\omega > \frac{B}{2}$ , the error increases as the artificial bandwidth  $B$  can no longer guarantee a sufficient sampling of the frequency domain. According to the Shannon-Nyquist Sampling theorem, we set  $B > 2\omega$ . By comparing the errors for different bandwidths  $B$  in the interval where the errors do not vary much, we observe an asymptotic behavior of  $\frac{1}{B}$ . Therefore, we can conclude that when solving the Helmholtz equation, the Hadamard integrator enjoys a frequency-dependent asymptotic convergence in the form of  $O(\frac{1}{\omega^{1.5}}) + O(\frac{1}{B^{0.5}})$ .

## 7 Conclusion

Based on the Kirchhoff-Huygens representation, Hadamard's ansatz and Babich's ansatz, we developed novel butterfly-compressed Eulerian Hadamard integrators for solving the high-frequency wave equation in time and frequency domains. We derive the Eulerian formulations via the Fourier transform and utilize the butterfly algorithm to accelerate the resulting matrix-vector multiplication. The proposed integrators propagate wave fields beyond caustics implicitly and advance spatially overturning waves naturally, with quasi-linear computational complexity and memory usage. Once constructed, the integrators can simultaneously solve the time-domain wave equations with different initial conditions or the frequency-domain wave equations with different point sources in the computational domain. Numerical examples illustrate the accuracy and efficiency of the new integrators.

## Appendix A Computation of Singular Terms

### A.1 Frequency-Space Integral

Consider

$$I(t, \mathbf{x}_0; \mathbf{x}) = \int_0^{\Delta\omega} e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega, \quad (8.1)$$

which is a frequency-space integral. Since the HKH propagator is short-time valid, both  $t$  and  $\tau$  are bounded by a sufficiently small  $\Delta T$ . In the numerical examples, we take  $\Delta\omega = \frac{\pi}{2}$ ,  $\Delta T = \frac{1}{8}$ . The small arguments  $\omega t$  and  $\omega\tau$  allow us to use the Taylor expansion of  $e^{-i\omega t}$  and polynomial approximation of Bessel functions. According to [1], we have the following polynomial approximations of Bessel functions  $J_0(x)$  and  $Y_0(x)$  when  $0 < x < 3$ ,

$$J_0(x) = 1 - a_1\left(\frac{x}{3}\right)^2 + a_2\left(\frac{x}{3}\right)^4 + \epsilon, \quad |\epsilon| = O\left(\left(\frac{x}{3}\right)^6\right), \quad (8.2)$$

and

$$Y_0(x) = \frac{2}{\pi} \ln\left(\frac{x}{2}\right) J_0(x) + b_1 + b_2\left(\frac{x}{3}\right)^2 - b_3\left(\frac{x}{3}\right)^4 + \epsilon, \quad |\epsilon| = O\left(\left(\frac{x}{3}\right)^6\right), \quad (8.3)$$

where

$$a_1 = 2.2499997, \quad a_2 = 1.2656208, \quad b_1 = 0.36746691, \quad b_2 = 0.60559366, \quad b_3 = 0.74350384.$$

Using the Taylor expansion of  $e^{-i\omega t}$ , we obtain

$$\begin{aligned} ie^{-i\omega t} H_0^1(\omega\tau) &\approx i\left(1 + i\omega t - \frac{\omega^2 t^2}{2} - \frac{i\omega^3 t^3}{6} + \frac{\omega^4 t^4}{24}\right) \left[\left(1 + \frac{2i}{\pi} \ln\left(\frac{\omega\tau}{2}\right)\right) \left(1 - a_1\left(\frac{\omega\tau}{3}\right)^2\right.\right. \\ &\quad \left.\left.+ a_2\left(\frac{\omega\tau}{3}\right)^4\right) + ib_1 + ib_2\left(\frac{\omega\tau}{3}\right)^2 + ib_3\left(\frac{\omega\tau}{3}\right)^4\right]. \end{aligned}$$

Dropping fifth- and higher-order terms as well as the imaginary parts, and calculating the definite integral over  $[0, \Delta\omega]$ , we obtain

$$\int_0^{\Delta\omega} e^{-i\omega t} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) d\omega \approx \frac{\sqrt{\pi}}{2} v_0 (L_1 - L_2), \quad (8.5)$$

where

$$\begin{aligned}
 L_1 \approx & \frac{1}{(\Delta\omega)\pi} \{-2(\Delta\omega)^2 + 2 \frac{[(\Delta\omega)^4(72t^2 + 16a_1\tau^2)]}{648} - \frac{(\Delta\omega)^6(54t^4 + 72a_1t^2\tau^2 + 16a_2\tau^4)}{16200} \\
 & + (\Delta\omega)^2 \ln\left(\frac{(\Delta\omega)\tau}{2}\right) + \frac{(\Delta\omega)^6 \ln\left(\frac{(\Delta\omega)\tau}{2}\right)(54t^4 + 72a_1t^2\tau^2 + 16a_2\tau^4)}{3240} \\
 & - \frac{(\Delta\omega)^4 \ln\left(\frac{(\Delta\omega)\tau}{2}\right)(216t^2 + 48a_1\tau^2)}{648}\} - \frac{1}{(\Delta\omega)}[-b_1(\Delta\omega)^2 \\
 & + \frac{(\Delta\omega)^4(108b_1t^2 - 24b_2\tau^2)}{648} + \frac{(\Delta\omega)^6(-27b_1t^4 + 36b_2t^2\tau^2 + 8b_3\tau^4)}{3240}],
 \end{aligned} \quad (8.6)$$

and

$$\begin{aligned}
 L_2 \approx & \int_0^{\Delta\omega} \left(\omega t - \frac{\omega^3 t^3}{6} + \frac{\omega^5 t^5}{120}\right) \left(1 - a_1\left(\frac{\omega\tau}{3}\right)^2 + a_2\left(\frac{\omega\tau}{3}\right)^4\right) d\omega \\
 \approx & \int_0^{\Delta\omega} \omega t \left(1 - a_1\left(\frac{\omega\tau}{3}\right)^2 + a_2\left(\frac{\omega\tau}{3}\right)^4\right) - \frac{\omega^3 t^3}{6} \left(1 - a_1\left(\frac{\omega\tau}{3}\right)^2\right) + \frac{\omega^5 t^5}{120} d\omega \\
 = & \int_0^{\Delta\omega} \omega t - \left(a_1 \frac{t\tau^2}{9} + \frac{t^3}{6}\right) \omega^3 + \left(a_2 \frac{t\tau^4}{81} + a_1 \frac{t^3\tau^2}{54} + \frac{t^5}{120}\right) \omega^5 d\omega \\
 = & \frac{(\Delta\omega)^2 t}{2} - \left(a_1 \frac{t\tau^2}{9} + \frac{t^3}{6}\right) \frac{(\Delta\omega)^4}{4} + \left(a_2 \frac{t\tau^4}{81} + a_1 \frac{t^3\tau^2}{54} + \frac{t^5}{120}\right) \frac{(\Delta\omega)^6}{6}.
 \end{aligned} \quad (8.7)$$

## A.2 Diagonal Term of $I(t, \mathbf{x}_0; \mathbf{x})$

As shown in the appendix of [34], the integral of self-intersection term inside source cell  $c_j$

$$\begin{aligned}
 \int_{c_j} \hat{G}_0(\omega_k, \mathbf{x}_0; \mathbf{x}) d\mathbf{x} & \approx \frac{i\sqrt{\pi}}{2} v_0(\mathbf{x}_0; \mathbf{x}_0) \frac{1}{(n_0\omega_k)^2} \left[ 8 \int_0^{\frac{\pi}{4}} \frac{hn_0\omega_k}{2\cos\theta} H_1^{(1)}\left(\frac{hn_0\omega_k}{2\cos\theta}\right) d\theta + 4i \right] \\
 & = \frac{i}{4v_0(n_0\omega_k)^2} \left[ 8 \int_0^{\frac{\pi}{4}} \frac{hn_0\omega_k}{2\cos\theta} H_1^{(1)}\left(\frac{hn_0\omega_k}{2\cos\theta}\right) d\theta + 4i \right],
 \end{aligned} \quad (8.8)$$

where  $n_0 = n(\mathbf{x}_0)$ ,  $v_0 = v(\mathbf{x}_0)$ , and we use the initial condition (3.10).

Now we consider a self-intersection term of  $I(t, \mathbf{x}_0; \mathbf{x})$ . Starting from (8.8), we have

$$\begin{aligned}
 \int_{c_j} I(t, \mathbf{x}_0; \mathbf{x}) d\mathbf{x} & = \int_0^{\Delta\omega} \frac{ie^{-i\omega t}}{4v_0(n_0\omega)^2} \left[ 8 \int_0^{\frac{\pi}{4}} \frac{hn_0\omega}{2\cos\theta} H_1^{(1)}\left(\frac{hn_0\omega}{2\cos\theta}\right) d\theta + 4i \right] d\omega \\
 & = \int_0^{\frac{\pi}{4}} \int_0^{\Delta\omega} \frac{2ie^{-i\omega t}}{v_0(n_0\omega)^2} \left[ \frac{hn_0\omega}{2\cos\theta} H_1^{(1)}\left(\frac{hn_0\omega}{2\cos\theta}\right) + \frac{2i}{\pi} \right] d\omega d\theta \\
 & = \int_0^{\frac{\pi}{4}} \int_0^{\Delta\omega} \frac{ie^{-i\omega t} h^2}{2v_0\cos^2\theta} \left(\frac{2\cos\theta}{hn_0\omega}\right)^2 \left[ \frac{hn_0\omega}{2\cos\theta} H_1^{(1)}\left(\frac{hn_0\omega}{2\cos\theta}\right) + \frac{2i}{\pi} \right] d\omega d\theta \\
 & \doteq S_1.
 \end{aligned} \quad (8.9)$$

To handle  $H_1^{(1)}(s) = J_1(s) + iY_1(s)$ , we introduce the polynomial approximation of Bessel functions  $J_1$  and  $Y_1$  [1] when  $0 < s < 3$ ,

$$s^{-1}J_1(s) = \frac{1}{2} - 0.56249985\left(\frac{s}{3}\right)^2 + 0.21093573\left(\frac{s}{3}\right)^4 - 0.03954289\left(\frac{s}{3}\right)^6 + 0.00443319\left(\frac{s}{3}\right)^8 \\ - 0.00031761\left(\frac{s}{3}\right)^{10} + 0.00001109\left(\frac{s}{3}\right)^{12} + \epsilon, \quad |\epsilon| < 1.3 \times 10^{-8}, \quad (8.10)$$

and

$$sY_1(s) = \frac{2}{\pi}s \ln\left(\frac{s}{2}\right)J_1(s) - \frac{2}{\pi} + 0.2212091\left(\frac{s}{3}\right)^2 + 2.1682709\left(\frac{s}{3}\right)^4 - 1.3164827\left(\frac{s}{3}\right)^6 \\ + 0.3123951\left(\frac{s}{3}\right)^8 - 0.0400976\left(\frac{s}{3}\right)^{10} + 0.0027873\left(\frac{s}{3}\right)^{12} + \epsilon, \quad |\epsilon| < 1.1 \times 10^{-7}. \quad (8.11)$$

Thus as  $s = \frac{hn_0\omega}{2\cos\theta} \rightarrow 0$ , we obtain

$$\frac{1}{s^2} \left( sH_1^1(s) + \frac{2i}{\pi} \right) = \frac{1}{s^2} \left[ s(J_1(s) + iY_1(s)) + \frac{2i}{\pi} \right] \\ = \frac{1}{s} J_1(s) + i\left(\frac{1}{s^2}\right) \left( sY_1(s) + \frac{2}{\pi} \right) \\ \sim \frac{1}{2} + i\frac{1}{\pi} \ln\left(\frac{1}{2}s\right). \quad (8.12)$$

Dropping higher order terms, we have

$$s^{-1}J_1(s) = \frac{1}{2} - 0.56249985\left(\frac{s}{3}\right)^2 + 0.21093573\left(\frac{s}{3}\right)^4 \quad (8.13)$$

for the real part and

$$\frac{1}{s^2} \left( sY_1(s) + \frac{2}{\pi} \right) = \frac{2}{\pi} \ln\left(\frac{s}{2}\right)s^{-1}J_1(s) + \frac{1}{s^2} \left( 0.2212091\left(\frac{s}{3}\right)^2 \right. \\ \left. + 2.1682709\left(\frac{s}{3}\right)^4 - 0.03954289\left(\frac{s}{3}\right)^6 \right) \quad (8.14)$$

for the imaginary part. The singularity appearing in the first term of (8.14) can be removed in the following way,

$$S_1 = \int_0^{\frac{\pi}{4}} \frac{ih^2}{2v_0 \cos^2 \theta} \int_0^{\Delta\omega} e^{-i\omega t} \left( \frac{2\cos\theta}{hn_0\omega} \right)^2 \left[ \frac{hn_0\omega}{2\cos\theta} H_1^{(1)} \left( \frac{hn_0\omega}{2\cos\theta} \right) + \frac{2i}{\pi} \right] d\omega d\theta \\ = \int_0^{\frac{\pi}{4}} \frac{ih^2}{2v_0 \cos^2 \theta} \int_0^{\Delta\omega} e^{-i\omega t} \left( \frac{2\cos\theta}{hn_0\omega} \right)^2 \left[ \frac{hn_0\omega}{2\cos\theta} H_1^{(1)} \left( \frac{hn_0\omega}{2\cos\theta} \right) + \frac{2i}{\pi} \right] d\omega d\theta \\ - \frac{i}{\pi} \ln\left(\frac{hn_0\omega}{4\cos\theta}\right) d\omega d\theta + \int_0^{\frac{\pi}{4}} \int_0^{\Delta\omega} \frac{i^2 h^2}{2v_0 \cos^2 \theta} \frac{1}{\pi} \ln\left(\frac{hn_0\omega}{4\cos\theta}\right) d\omega d\theta, \quad (8.15)$$

where the last singular term in (8.15) can be handled as

$$\int_0^{\frac{\pi}{4}} \int_0^{\Delta\omega} \frac{i^2 h^2}{2v_0 \cos^2 \theta} \frac{1}{\pi} \ln\left(\frac{hn_0\omega}{4\cos\theta}\right) d\omega d\theta \\ = -\frac{2h^2}{\pi v_0} \int_0^{\frac{\pi}{4}} \int_0^{\Delta\omega} \frac{1}{\cos^2 \theta} \ln\left(\frac{hn_0\omega}{4\cos\theta}\right) d\omega d\theta \\ = -\frac{2h^2}{\pi v_0} \int_0^{\frac{\pi}{4}} \int_0^{\Delta\omega} \frac{1}{\cos^2 \theta} [\ln(hn_0\omega) - \ln(4\cos\theta)] d\omega d\theta \\ = -\frac{2h^2}{\pi v_0} \int_0^{\frac{\pi}{4}} \frac{1}{\cos^2 \theta} [\Delta\omega \ln\left(\frac{hn_0\Delta\omega}{4\cos\theta}\right) - \Delta\omega] d\theta. \quad (8.16)$$

### A.3 Diagonal Term of $\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\tau \partial \tau} \right) \left( \frac{u^k}{v_0} \right)$

Consider the self-intersection terms (4.19) with non-zero  $\omega$  in cell  $c_j$ , where  $n$  and non-singular ingredients are taken to be constant. We state that

$$S_2 \doteq \int_{c_j} \hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) \left( \frac{\partial}{\tau \partial \tau} \right) \left( \frac{u^k}{v_0} \right) d\mathbf{x} = 0, \quad k = 1, 2, \quad (8.17)$$

which implies that the self-intersection term is zero.

To see this, we note that, when  $n = n_0$ ,

$$\frac{\partial}{\partial \tau} = c^2 \nabla \tau \cdot \nabla = \frac{1}{n_0} (\cos \theta, \sin \theta) \cdot \nabla, \quad (8.18)$$

where  $\theta$  is the take-off angle, which is consistent with that in the polar coordinate transformation when  $n = n_0$ . And

$$\hat{G}_0(\omega, \mathbf{x}_0; \mathbf{x}) = \frac{i}{4v_0} H_0^1(\omega n_0 |\mathbf{x} - \mathbf{x}_0|), \quad (8.19)$$

which is a radial function. Then we can obtain (8.17) by symmetry. Actually, we can divide the cell  $c_j$  into four parts

$$\begin{cases} c_j^1 : \theta \in [-\frac{\pi}{4}, \frac{\pi}{4}], r \in [0, \frac{h}{2 \cos(\theta)}], \\ c_j^2 : \theta \in [\frac{\pi}{4}, \frac{3\pi}{4}], r \in [0, \frac{h}{2 \sin(\theta)}], \\ c_j^3 : \theta \in [\frac{3\pi}{4}, \frac{5\pi}{4}], r \in [0, -\frac{h}{2 \cos(\theta)}], \\ c_j^4 : \theta \in [\frac{5\pi}{4}, \frac{7\pi}{4}], r \in [0, -\frac{h}{2 \sin(\theta)}]. \end{cases} \quad (8.20)$$

Now we use the polar coordinate transformation in (8.17), yielding

$$\begin{aligned} S_2 &= \iint \frac{i}{4v_0} H_0^1(\omega n_0 |\mathbf{x} - \mathbf{x}_0|) \frac{1}{n_0} (\cos \theta, \sin \theta) \cdot \nabla \left( \frac{u^k}{v_0} \right) \frac{1}{n_0 |\mathbf{x} - \mathbf{x}_0|} |\mathbf{x} - \mathbf{x}_0| dr d\theta \\ &= \frac{i}{4\rho_j} \left( \int_{c_j^1} + \int_{c_j^2} + \int_{c_j^3} + \int_{c_j^4} \right) H_0^1(\omega n_0 |\mathbf{x} - \mathbf{x}_0|) (\cos \theta, \sin \theta) \cdot \nabla \left( \frac{u^k}{v_0} \right) dr d\theta \\ &\doteq S_{2,1} + S_{2,2} + S_{2,3} + S_{2,4}. \end{aligned} \quad (8.21)$$

Substituting  $\xi = \theta + \pi$ , it is easy to verify that

$$S_{2,1} + S_{2,3} = 0, \quad S_{2,2} + S_{2,4} = 0.$$

That is,  $S_2 = 0$ .

**Funding** J. Cheng is supported by National Key Research and Development Programs of China (No. 2023YFA1009103), NSFC 11971121, 12241103, the Sino-German Mobility Programme (M-0187) by the Sino-German Center for Research Promotion and Science and Technology Commission of Shanghai Municipality (23JC1400501). S. Leung is supported by the Hong Kong RGC under grant 16302223. Qian's research is partially supported by NSF 2012046, 2152011, and 2309534.

**Data Availability** Data sets generated during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no Conflict of interest.

## References

1. Abramowitz, M., Stegun, I.A.: Handbook of mathematical functions. Dover Publications Inc, New York (1965)
2. Avila, G.S., Keller, J.B.: The high-frequency asymptotic field of a point source in an inhomogeneous medium. *Commun. Pure Appl. Math.* **16**, 363–381 (1963)
3. Babich, V.M.: The short wave asymptotic form of the solution for the problem of a point source in an inhomogeneous medium. *USSR Comput. Math. Math. Phys.* **5**(5), 247–251 (1965)
4. Babich, V.M., Buldyrev, V.S.: Asymptotic methods in short wave diffraction problems. Nauka, Moscow (1972). ((in Russian))
5. Babuška, I.M., Sauter, S.A.: Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Rev.* **42**, 451–484 (2000)
6. Bao, G., Lai, J., Qian, J.: Fast multiscale Gaussian beam methods for wave equations in bounded domains. *J. Comput. Phys.* **261**, 36–64 (2014)
7. Bayliss, A., Goldstein, C.I., Turkel, E.: On accuracy conditions for the numerical computation of waves. *J. Comput. Phys.* **59**, 396–404 (1985)
8. Benamou, J.D.: An introduction to Eulerian geometrical optics (1992–2002). *J. Sci. Comp.* **19**, 63–93 (2003)
9. Berenger, J.P.: A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.* **114**, 185–200 (1994)
10. Candes, E., Demanet, L., Ying, L.: A fast butterfly algorithm for the computation of Fourier integral operators. *SIAM Multiscale Model. Simul.* **7**, 1727–1750 (2009)
11. Courant, R., Hilbert, D.: Methods of Mathematical Physics, vol. II. Interscience Publishers, New York (1964)
12. Engquist, B., Runborg, O.: Computational high frequency wave propagation. *Acta Numer.* **12**, 181–266 (2003)
13. Engquist, B., Ying, L.: Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *Multiscale Model. Simul.* **9**, 686–710 (2011)
14. Fomel, S., Luo, S., Zhao, H.K.: Fast sweeping method for the factored eikonal equation. *J. Comput. Phys.* **228**, 6440–6455 (2009)
15. Gelfand, I., Shilov, G.: Generalized Functions, vol. I. Academic Press, New York and London (1964)
16. Grote, M.J., Sim, I.: Efficient pml for the wave equation (2010)
17. Hadamard, J.: Lectures on Cauchy's Problem in Linear Partial Differential Equations. Yale University Press; (reprinted Dover Publications, New York 1952) (1923)
18. Ho, K., Leung, S., Qian, J.: Fast Huygens sweeping methods for a class of nonlocal Schrödinger equations. *J. Sci. Comput.* **88**, 54 (2021)
19. Jiang, G.S., Peng, D.: Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.* **21**, 2126–2143 (2000)
20. Jo, C.H., Shin, C., Suh, J.H.: An optimal 9-point, finite-difference, frequency-space, 2-d scalar wave extrapolator. *Geophysics* **61**(2), 529–537 (1996)
21. Kao, C.Y., Osher, S.J., Qian, J.: Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations. *J. Comput. Phys.* **196**, 367–391 (2004)
22. Keller, J., Lewis, R.: Asymptotic methods for partial differential equations: the reduced wave equation and Maxwell's equations. *Surv. Appl. Math.* **1**, 1–82 (1995)
23. Kwan, W., Leung, S., Wang, X.P., Qian, J.: A fast Huygens sweeping method for capturing paraxial multi-color optical self-focusing in nematic liquid crystals. *J. Comput. Phys.* **348**, 108–138 (2017)
24. Lax, P.: Asymptotic solutions of oscillatory initial value problems. *Duke Math. J.* **24**, 627–645 (1957)
25. Leung, S., Qian, J.: Eulerian Gaussian beam methods for Schrödinger equations in the semi-classical regime. *J. Comput. Phys.* **228**, 2951–2977 (2009)
26. Leung, S., Qian, J.: The backward phase flow and FBI-transform-based Eulerian Gaussian beams for the Schrödinger equation. *J. Comput. Phys.* **229**, 8888–8917 (2010)
27. Leung, S., Qian, J., Burridge, R.: Eulerian Gaussian beams for high frequency wave propagation. *Geophysics* **72**, SM61–SM76 (2007)



28. Leung, S., Qian, J., Serna, S.: Fast Huygens sweeping methods for Schrodinger equations in the semiclassical regime. *Methods Appl. Anal.* **21**, 031–066 (2014)
29. Leung, S., Zhao, H.K.: Gaussian beam summation for diffraction in inhomogeneous media based on the grid based particle method. *Comm. Comp. Phys.* **8**, 758–796 (2010)
30. Li, Y., Yang, H.: Interpolative butterfly factorization. *SIAM J. Sci. Comput.* **39**(2), A503–A531 (2017)
31. Li, Y., Yang, H., Martin, E.R., Ho, K.L., Ying, L.: Butterfly factorization. *Multiscale Model. Simul.* **13**(2), 714–732 (2015)
32. Liu, X.D., Osher, S.J., Chan, T.: Weighted Essentially NonOscillatory schemes. *J. Comput. Phys.* **115**, 200–212 (1994)
33. Liu, Y., Guo, H., Michielssen, E.: An hss matrix-inspired butterfly-based direct solver for analyzing scattering from two-dimensional objects. *IEEE Antennas Wirel. Propag. Lett.* **16**, 1179–1183 (2016)
34. Liu, Y., Song, J., Burrridge, R., Qian, J.: A fast butterfly-compressed Hadamard-Babich integrator for high-frequency Helmholtz equations in inhomogeneous media with arbitrary sources. *SIAM Multiscale Model. Simul.* **21**, 269–308 (2023)
35. Liu, Y., Xing, X., Guo, H., Michielssen, E., Ghysels, P., Li, X.S.: Butterfly factorization via randomized matrix-vector multiplications. *SIAM J. Sci. Comput.* **43**, A883–A907 (2021)
36. Lu, W., Qian, J., Burrridge, R.: Babich-like ansatz for three-dimensional point-source Maxwell's equations in an inhomogeneous medium at high frequencies. *SIAM J. Multiscale Model. Simul.* **14**(3), 1089–1122 (2016)
37. Lu, W., Qian, J., Burrridge, R.: Babich's expansion and the fast Huygens sweeping method for the Helmholtz wave equation at high frequencies. *J. Comput. Phys.* **313**, 478–510 (2016)
38. Lu, W., Qian, J., Burrridge, R.: Extending Babich's ansatz for point-source Maxwell's equations using Hadamard's method. *SIAM J. Multiscale Model. Simul.* **16**, 727–751 (2018)
39. Ludwig, D.: Uniform asymptotic expansions at a caustic. *Commun. Pure Appl. Math.* **19**, 215–250 (1966)
40. Luo, S., Qian, J., Burrridge, R.: Fast Huygens sweeping methods for Helmholtz equations in inhomogeneous media in the high frequency regime. *J. Comput. Phys.* **270**, 378–401 (2014)
41. Luo, S., Qian, J., Burrridge, R.: High-order factorization based high-order hybrid fast sweeping methods for point-source eikonal equations. *SIAM J. Numer. Anal.* **52**, 23–44 (2014)
42. Maslov, V.P., Fedoriuk, M.V.: Semi-classical approximation in quantum mechanics. D. Reidel Publishing Company (1981)
43. Michielssen, E., Boag, A.: A multilevel matrix decomposition algorithm for analyzing scattering from large structures. *IEEE Trans. Antennas Propag.* **44**(8), 1086–1093 (1996)
44. Osher, S.J., Shu, C.W.: High-order Essentially NonOscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.* **28**, 907–922 (1991)
45. Qian, J., Lu, W., Yuan, L., Luo, S., Burrridge, R.: Eulerian geometrical optics and fast Huygens sweeping methods for three-dimensional time-harmonic high-frequency Maxwell's equations in inhomogeneous media. *SIAM J. Multiscale Model. Simul.* **16**, 595–636 (2016)
46. Qian, J., Song, J.: Fast multiscale Gaussian beam method for three-dimensional elastic wave equations in bounded domains. *SIAM J. Numer. Anal.* **59**, 2536–2570 (2021)
47. Qian, J., Song, J., Lu, W., Burrridge, R.: Hadamard-Babich ansatz for point-source elastic wave equations in variable media at high frequencies. *SIAM Multiscale Model. Simul.* **19**, 46–86 (2021)
48. Qian, J., Song, J., Lu, W., Burrridge, R.: Truncated Hadamard-Babich ansatz and fast Huygens sweeping methods for time-harmonic high-frequency elastic wave equations in inhomogeneous media. *Minimax Theory Appl.* **8**, 171–212 (2023)
49. Qian, J., Symes, W.W.: An adaptive finite difference method for traveltime and amplitude. *Geophysics* **67**, 167–176 (2002)
50. Qian, J., Ying, L.: Fast Gaussian wavepacket transforms and Gaussian beams for the Schrödinger equation. *J. Comput. Phys.* **229**, 7848–7873 (2010)
51. Qian, J., Ying, L.: Fast multiscale Gaussian wavepacket transforms and multiscale Gaussian beams for the wave equation. *SIAM J. Multiscale Model. Simul.* **8**, 1803–1837 (2010)
52. Qian, J., Yuan, L., Liu, Y., Luo, S., Burrridge, R.: Babich's expansion and high-order Eulerian asymptotics for point-source Helmholtz equations. *J. Sci. Comput.* **67**, 883–908 (2016)
53. Symes, W.W., Qian, J.: A slowness matching Eulerian method for multivalued solutions of eikonal equations. *J. Sci. Comp.* **19**, 501–526 (2003)
54. Taflov, A., Hagness, S.C., Picket-May, M.: Computational electromagnetics: the finite-difference time-domain method. *Electr. Eng. Handbook* **3**(629–670), 15 (2005)
55. Tanushev, N., Engquist, B., Tsai, R.: Gaussian beam decomposition of high frequency wave fields. *J. Comput. Phys.* **228**, 8856–8871 (2009)
56. Tanushev, N., Qian, J., Ralston, J.: Mountain waves and Gaussian beams. *SIAM J. Multiscale Model. Simul.* **6**, 688–709 (2007)

- 57. Wei, Y., Cheng, J., Burrridge, R., Qian, J.: Hadamard integrator for time-dependent wave equations: Lagrangian formulation via ray tracing. *J. Comput. Phys.* **497**, 112637 (2024)
- 58. White, B.S.: The stochastic caustic. *SIAM J. Appl. Math.* **44**, 127–149 (1984)
- 59. Zhang, L., Rector, J.W., Hoversten, G.M.: Eikonal solver in the celerity domain. *Geophys. J. Internat.* **162**, 1–8 (2005)
- 60. Zhao, H.K.: Fast sweeping method for eikonal equations. *Math. Comp.* **74**, 603–627 (2005)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Yuxiao Wei<sup>1</sup> · Jin Cheng<sup>1</sup> · Shingyu Leung<sup>2</sup> · Robert Burrridge<sup>3</sup> · Jianliang Qian<sup>4</sup> 

✉ Jianliang Qian  
jqian@msu.edu

Yuxiao Wei  
19110180029@fudan.edu.cn

Jin Cheng  
jcheng@fudan.edu.cn

Shingyu Leung  
masyeung@ust.hk

Robert Burrridge  
burrigel37@gmail.com

<sup>1</sup> School of Mathematical Sciences, Fudan University, Shanghai 200433, China

<sup>2</sup> Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China

<sup>3</sup> Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131, USA

<sup>4</sup> Department of Mathematics and Department of CMSE, Michigan State University, East Lansing, MI 48824, USA