

## MIT Open Access Articles

### *Fast wave computation via Fourier integral operators*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Demanet, Laurent, and Lexing Ying. "Fast Wave Computation via Fourier Integral Operators." *Mathematics of Computation* 81.279 (2012): 1455–1486. Web. 11 Apr. 2012.

**As Published:** <http://dx.doi.org/10.1090/S0025-5718-2012-02557-9>

**Publisher:** American Mathematical Society (AMS)

**Persistent URL:** <http://hdl.handle.net/1721.1/78677>

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

**Terms of use:** Creative Commons Attribution-Noncommercial-Share Alike 3.0



# Fast Wave Computation via Fourier Integral Operators

Laurent Demanet<sup>‡</sup> and Lexing Ying<sup>§</sup>

<sup>‡</sup> Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139

<sup>§</sup> Department of Mathematics and ICES, The University of Texas at Austin, Austin, TX 78712

July 2010, Revised April 2011

## Abstract

This paper presents a numerical method for “time upscaling” wave equations, i.e., performing time steps not limited by the Courant-Friedrichs-Lewy (CFL) condition. The proposed method leverages recent work on fast algorithms for pseudodifferential and Fourier integral operators (FIO). This algorithmic approach is not asymptotic: it is shown how to construct an exact FIO propagator by 1) solving Hamilton-Jacobi equations for the phases, and 2) sampling rows and columns of low-rank matrices at random for the amplitudes. The setting of interest is that of scalar waves in two-dimensional smooth periodic media (of class  $C^\infty$  over the torus), where the bandlimit  $N$  of the waves goes to infinity. In this setting, it is demonstrated that the algorithmic complexity for solving the wave equation to fixed time  $T \simeq 1$  can be as low as  $O(N^2 \log N)$  with controlled accuracy. Numerical experiments show that the time complexity can be lower than that of a spectral method in certain situations of physical interest.

**Keywords.** Wave equations, Fourier integral operators, discrete symbol calculus, random sampling, separated approximation, multiscale computations.

**AMS subject classifications.** 65M80, 65T99

**Acknowledgments.** The authors would like to acknowledge funding from the National Science Foundation and the Sloan Foundation, as well as interesting discussions with Nicholas Maxwell.

## 1 Introduction

This paper is concerned with the rapid solution of the two dimensional wave equation with variable coefficient:

$$\begin{cases} \partial_{tt}u(x, t) - \nabla \cdot (c^2(x)\nabla u(x, t)) = 0 & t > 0, x \in [0, 1]^2 \\ u(x, 0) = u_0(x) & x \in [0, 1]^2 \\ \partial_t u(x, 0) = u_1(x) & x \in [0, 1]^2, \end{cases} \quad (1.1)$$

where the boundary conditions are taken to be periodic. We assume that  $c(x)$  is positive and smooth – essentially bandlimited.

The initial condition  $(u_0(x), u_1(x))$  is typically discretized on a uniform  $N \times N$  grid. For a fixed final time  $T \simeq 1$ , we seek an algorithm for computing the time- $T$  solution  $u(x, T)$  in almost linear time.

Standard methods for solving this system uses finite difference or spectral differentiation in the spatial domain and forward marching in the time domain. Suppose that the initial conditions  $(u_0(x), u_1(x))$  are discretized with an  $N \times N$  Cartesian grid. The CFL condition restricts the time step to be of order  $O(1/N)$ . Hence, in order to computing the solution  $u(x, T)$  at a time  $T = O(1)$ , on the order of  $N$  applications of the small-time propagator are needed, for a total complexity of  $O(N^3)$  – or more if an accuracy estimate is desired.

In this paper, we propose to lower complexity by representing the propagator as a Fourier integral operator. This approach permits *time upscaling*, namely, that much larger  $O(1)$  time steps are now allowed. As a result, evaluating the solution  $u(x, T)$  using an upscaled time step  $\tau$  takes only  $T/\tau = O(1)$  steps. Moreover, each evaluation of the Fourier integral operator representation can be performed efficiently in a time complexity as low as  $O(N^2 \log N)$ . Therefore, this new approach requires only  $O(N^2 \log N)$  steps to compute the solution  $u(x, T)$ .

Preparing the Fourier integral operator representation of the solution, however, may have complexity greater than  $O(N^2 \log N)$ . Our proposed solution to this problem has complexity  $O(N^3)$ .

## 1.1 General strategy

Our approach is based on the Fourier integral operator representation of the solution of (1.1). Since  $L := -\nabla \cdot (c^2(x)\nabla)$  is a positive semidefinite operator, we define  $P := L^{1/2}$  to be the positive semidefinite square root of  $L$ . Using this operator notation, we can rewrite (1.1) as

$$\partial_{tt}u + P^2u = 0.$$

Factorizing this operator equation gives

$$(\partial_t + iP)(\partial_t - iP)u = 0.$$

Therefore, the general solution of (1.1) is given by

$$u(x, t) = (e^{iPt}f_+)(x) + (e^{-iPt}f_-)(x),$$

where  $f_{\pm}$  are two arbitrary functions. Matching with the initial conditions gives

$$f_{\pm} = \frac{1}{2}(u_0 \pm (iP)^{-1}u_1). \quad (1.2)$$

The theory of Fourier integral operators states that for a given  $c(x) > 0$ , assumed to be of class  $C^\infty$ , there exists a time  $t^*$  that depends only on  $c(x)$  such that for any  $t < t^*$ ,  $e^{iPt}f_+$  and  $e^{-iPt}f_-$  have the following Fourier integral operator (FIO) representation:

$$(e^{\pm iPt}f_{\pm})(x) = \sum_{\xi \in \mathbb{Z}^2} e^{2\pi i\Phi_{\pm}(x, \xi, t)} a_{\pm}(x, \xi, t) \widehat{f_{\pm}}(\xi). \quad (1.3)$$

Here the Fourier transforms of  $f_{\pm}(x)$  are defined by

$$\widehat{f_{\pm}}(\xi) = \int_{[0,1]^2} e^{-2\pi i x \cdot \xi} f_{\pm}(x) dx.$$

$\Phi_{\pm}(x, \xi, t)$  are called the phase functions and they are smooth in  $x$  and  $\xi \neq 0$  with homogeneous degree one in  $\xi$ .  $a_{\pm}(x, \xi, t)$  are called the amplitude functions, and for any given

$t$ ,  $a_{\pm}(x, \xi, t)$  have a separated approximation in  $x$  vs.  $\xi$  with a small number of terms. Equation (1.3) is exact; its justification is given in Section 3.

It is important to notice that the medium  $c(x)$  needs not only to be of class  $C^{\infty}$ , but of small “numerical” bandwidth for the FIO representation to be numerically advantageous. In particular, the smoother  $c(x)$  the larger the cut-off time  $t^*$  before caustics develop in the evolution of initially plane waves.

The natural spatial discretization for smooth functions on the torus involves sampling functions on the  $N \times N$  Cartesian grid

$$X := \left\{ \left( \frac{n_1}{N}, \frac{n_2}{N} \right) : n_1, n_2 \in \mathbb{Z}, 0 \leq n_1, n_2 < N \right\}.$$

For the discretization in time, a large time step  $\tau < t^*$  is chosen. For  $x \in X$ , the solution at time  $\tau$  is approximated as

$$u(x, \tau) \approx \sum_{\xi \in \Omega} e^{2\pi i \Phi_+(x, \xi, \tau)} a_+(x, \xi, \tau) \widehat{f}_+(\xi) + \sum_{\xi \in \Omega} e^{2\pi i \Phi_-(x, \xi, \tau)} a_-(x, \xi, \tau) \widehat{f}_-(\xi).$$

Here  $\widehat{f}_{\pm}(\xi) = \frac{1}{N^2} \sum_{x \in X} e^{-2\pi i x \cdot \xi} f_{\pm}(x)$  are now discrete Fourier transforms of  $\{f_{\pm}(x), x \in X\}$  and both sums are taken over

$$\Omega := \{(\xi_1, \xi_2) : \xi_1, \xi_2 \in \mathbb{Z}, -\beta N/2 < \xi_1, \xi_2 < \beta N/2\},$$

where  $0 < \beta < 1$ . In the case of constant coefficient  $c(x)$ , we can choose  $\beta = 1$ . However, for variable coefficient  $c(x)$ ,  $\beta$  is chosen to be adequately bounded away from 1 to avoid an unwanted aliasing effect.

For  $t = n\tau$  an integer multiple of  $\tau$ , the solution at  $t = (n-1)\tau$  is taken as an initial condition and the procedure is repeated to obtain an approximation of  $u(x, n\tau)$ . In order to carry out this procedure algorithmically, we need to address the following questions:

- How to construct the square root operator  $P$  and its inverse  $P^{-1}$ , and how to apply them efficiently to functions? Discrete symbol calculus (DSC) is a natural answer to this question [13]. With the help of DSC, constructing  $P$  and  $P^{-1}$  takes only a number of steps that is polylogarithmic in  $N$ . Applying  $P$  to any vector defined on  $X$  can also be performed efficiently in only  $O(N^2 \log N)$  steps.
- How to compute the phase functions  $\Phi_{\pm}(x, \xi, \tau)$ ? It is well known that they satisfy the Hamilton-Jacobi equations:

$$\begin{cases} \partial_t \Phi_{\pm}(x, \xi, t) \mp c(x) |\nabla_x \Phi_{\pm}(x, \xi, t)| = 0 \\ \Phi_{\pm}(x, \xi, 0) = x \cdot \xi. \end{cases} \quad (1.4)$$

Since  $c(x)$  is a  $C^{\infty}$  function whose (numerical) bandwidth is small compared to  $N$ , we can solve  $\Phi_{\pm}(x, \xi, \tau)$  up to a very high accuracy on a much smaller grid with spectral differentiation in space and an accurate time stepping scheme in time (such as Runge-Kutta).

- How to compute the amplitude functions  $a_{\pm}(x, \xi, \tau)$ ? When viewed as a matrix,  $(a_{\pm}(x, \xi, \tau))_{x \in X, \xi \in \Omega}$  is numerically low-rank. The construction of a low-rank separated approximation of  $a_{\pm}(x, \xi, \tau)$  is the main contribution of the current paper. The method adopted here is based on the randomized sampling algorithm for constructing factorizations for low-rank matrices. This procedure only requires sampling a

constant number of rows and columns of the matrix randomly. In our setting, sampling the rows and columns of  $a_{\pm}(x, \xi, \tau)$  reduces to solving (1.1) with special initial conditions that correspond to plane waves and Dirac deltas. We solve these special initial value problems with the standard spectral differentiation and time stepping algorithm. As a result, the construction of  $a_{\pm}(x, \xi, \tau)$  is treated as a precomputation that takes about  $O(N^3)$  steps.

- Finally, how to evaluate the Fourier integral operator (1.3) for given  $f_{\pm}(x, \xi, \tau)$ ,  $a_{\pm}(x, \xi, \tau)$  and  $\Phi_{\pm}(x, \xi, \tau)$ ? Along with E. Candès, the authors have already developed two efficient algorithms for this problem, one with a small constant and complexity  $O(N^{2.5} \log N)$  [7] and the other with a slightly larger constant and the optimal complexity  $O(N^2 \log N)$  [8]. Here, we simply resort to these algorithms.

## 1.2 Applications

The main application for fast wave computation in variable media might be seismic imaging, where wave speed in the subsurface (among other parameters, possibly) is inferred from recorded wave echos [32]. The modern, more accurate imaging algorithms all involve wave propagation on a very large computational scale. Slow execution and large memory imprints limit the size of problems that seismologists can currently consider, and even for small problems, limit the number of iterations that can be done in an inversion loop. Time-stepping, (or depth-stepping, or looping over all frequencies,) is the main culprit.

The assumption of a two-dimensional smooth periodic medium for the numerical experiments presented in this paper should be put in perspective.

- Nothing prevents the application of the algorithms to three spatial dimensions. This is in contrast with some of our previous work on wave atoms [12], where the separation technique was intrinsically two-dimensional.
- Absorbing layers such as the perfectly matched layer (PML) are compatible with periodic boundary conditions: it suffices to let the absorption coefficient be maximal at the edges where the periodic stitching is done. If in addition this coefficient is smooth, it should not affect the smoothness and separability properties of the amplitude in the FIO representation. If a seismologist is willing to surround all sides of his or her domain with PML, then periodic boundary conditions are not an issue. If some water-air or rock-air interface demands a Neumann boundary condition, it is not inconceivable that a FIO approach may work, but new ideas such as Fourier continuation [4] would be needed.
- Smooth background media are not physical in the Earth, but they are the simplified model that arises from inversion processes such a traveltime tomography. It is a great “discovery” of seismologists that the high-frequency singularities that produce scattering are well treated by linearization and need not be part of the model velocity, in which waves are simulated. In fact, they should *not* be part of the model velocity: it is well documented that the output least-squares objective of full-waveform inversion becomes quite nonconvex in the presence of oscillatory or singular model velocities [31].

Other applications may include certain ultrasound techniques in medical imaging, where the goal is to image contrasts in the shear stiffness of different organs. Transient elastography may be one such technique [19]. Current methodologies mostly deal with traveltimes

rather than waveforms, but if the progress in seismology is any indication, a full modeling of the background shear wave speed may one day prove useful.

### 1.3 Previous work

The first attempt to solve the one-dimensional wave equation accurately using large time steps is probably the work of Engquist, Osher and Zhong [16]. They constructed the wavelet representation of the solution operator, a sparse matrix that can then be applied to the initial condition in the wavelet domain. Although no estimate of complexity and accuracy was given, their algorithm runs in near linear-time complexity.

Their work was generalized to two spatial dimensions by Candès and the authors, with complexity and accuracy estimates. In [5, 6] it was shown that curvelets provide a suitably sparse representation of the wave propagator. In [12], the agenda of fast wave computation was operationalized with wave atoms, another transform that achieved a greater flexibility at obtaining a sparse representation. Genuine time upscaling and advantageous time complexity were obtained in [12], but at the expense of a complex code that only handled certain special smooth media well.

Other approaches have been proposed for realizing time upscaling of wave equations. This includes the work of Beylkin and Sandberg, where an economical representation of the propagator is obtained via low-rank separation in the prolate spheroidal wavefunction domain [3]. Stolk proposed to couple the geometrical optics asymptotic formulas with wavelets in one spatial dimension [30].

Concerning fast computation of general pseudodifferential and Fourier integral operators (FIO), Candès and the authors have reported on different ideas in [13, 7, 8]. Many of the algorithmic tools used in those papers are present in different forms or different contexts; let us for instance mention work on angular decompositions of the symbol of pseudodifferential operators [1], work on the butterfly algorithm and its applications in [21, 23, 38, 37, 35], work on fast “beamforming” methods for filtered backprojection for Radon and generalized Radon transforms (a problem similar to wave propagation) in [36, 22, 2], work on the plane-wave time-domain fast multipole method summarized in Chapters 18 and 19 of [9], and work on “phase-screen” methods in geophysics [11].

Note that the idea of using FIO for solving the wave equation is a pillar of microlocal analysis. It dates back to at least Lax in 1957 [20], and ultimately to geometrical optics and the WKB expansion schemes in 1930s quantum mechanics. The modern, careful study of propagation of singularities with FIO is due to Hormander and Duistermaat [18, 14]. Important analytical estimates on FIO and wave equations are due to Cordoba and Fefferman [10], Stein et al. [25, 29], and more recently Smith [27].

### 1.4 Contents

The rest of this paper is organized as follows. In Section 2, we briefly review the numerical tools that are used in these papers. They include discrete symbol calculus, randomized sampling method for low-rank factorizations, and fast algorithms for applying Fourier integral operators. Section 3 includes the proofs that justify our approach theoretically. In Section 4, we describe our algorithm in detail. Several numerical experiments are provided in Section 5.

## 2 Background

### 2.1 Discrete symbol calculus

This section is a summary of [13]. We use the discrete symbol calculus (DSC) framework to represent the operators  $P = L^{1/2}$  and  $P^{-1} = L^{-1/2}$ . In short, discrete symbol calculus is concerned with efficiently representing, manipulating, and applying pseudodifferential operators

$$(Af)(x) = \sum_{\xi \in \mathbb{Z}^2} e^{2\pi i x \cdot \xi} a(x, \xi) \hat{f}(\xi).$$

Here we assume that the symbol  $a(x, \xi)$  belongs to the standard symbol class  $S^d$ , i.e.,

$$\partial_{\xi}^{\alpha} \partial_x^{\beta} a(x, \xi) \leq C_{\alpha\beta} \langle \xi \rangle^{d-|\alpha|}, \quad \langle \xi \rangle := (1 + |\xi|^2)^{1/2}.$$

The degree of  $a(x, \xi)$  is defined to be the minimum  $d$  such that  $a(x, \xi) \in S^d$ . In what follows, we use  $\Lambda_K$  to denote the grid  $\{(n_1, n_2) : n_1, n_2 \in \mathbb{Z}, -K < n_1, n_2 < K\}$ .

**Representations** Let  $a(x, \xi)$  be a symbol of class  $S^{d_a}$ , where  $d_a$  is the degree of  $a(x, \xi)$ . Expanding  $a(x, \xi)$  in the  $x$  variable using Fourier basis  $e_{\lambda}(x) := e^{2\pi i \lambda \cdot x}$  gives

$$a(x, \xi) = \sum_{\lambda} e_{\lambda}(x) a_{\lambda}(\xi),$$

where  $a_{\lambda}(\xi)$  are the Fourier coefficients (in the  $x$  variable) of  $a(x, \xi)$ . If  $a(x, \xi)$  is essentially bandlimited in  $\lambda \in \Lambda_B$  for a constant  $B$  in the  $x$  variable, one only needs to keep the terms associated with these  $\lambda$  values in the sum. The  $x$ -Fourier coefficients  $a_{\lambda}(\xi)$  can be computed from the samples of  $a(x, \xi)$  on a  $2B \times 2B$  Cartesian grid in  $x$  variable for each  $\xi$ .

The representation of the symbol in the  $\xi$  variable is slightly more complicated. The  $x$ -Fourier coefficients  $a_{\lambda}(\xi)$  inherits the same smoothness property in the  $\xi$  variable from  $a(x, \xi)$ , i.e., the normalized term  $a_{\lambda}(\xi) \langle \xi \rangle^{-d_a}$  gains smoothness as  $\xi$  goes to infinity. Due to this consideration, it is natural to approximate  $a_{\lambda}(\xi) \langle \xi \rangle^{-d_a}$  with basis functions  $g_{\mu}(\xi)$  that exhibit the same behavior:

$$a_{\lambda}(\xi) \langle \xi \rangle^{-d_a} \approx \sum_{\mu} a_{\lambda\mu} g_{\mu}(\xi).$$

Two choices of  $g_{\mu}(\xi)$  give good theoretical and numerical results:

- Rational Chebyshev interpolation. This approach starts by studying  $a_{\lambda}(\xi) \langle \xi \rangle^{-d_a}$  in polar coordinates  $\xi = (r, \theta)$ . We first map the half line  $r \in [0, \infty)$  to the interval  $s \in [-1, 1)$  with rational functions

$$s(r) = \frac{r + L}{r - L}, \quad r(s) = L \frac{1 + s}{1 - s},$$

where  $L$  is a fixed constant. Within the  $(s, \theta)$  coordinate,  $a_{\lambda}(\xi) \langle \xi \rangle^{-d_a}$  becomes  $a_{\lambda}(s, \theta) \left\langle L \frac{1+s}{1-s} \right\rangle^{-d_a}$ . We interpolate it with basis functions that are tensor products of Fourier bases in  $\theta$  and Chebyshev functions in  $s$ . This corresponds to choosing

$$g_{\mu}(\xi) = g_{(m,n)}(s, \theta) = e^{im\theta} T_n(s)$$

where  $T_n(s)$  are Chebyshev functions. Due to its smoothness property, only a small number of basis functions are required to interpolate  $a_\lambda(s, \theta) \left\langle L \frac{1+s}{1-s} \right\rangle^{-d_a}$  accurately.

The interpolation coefficients  $a_{\lambda\mu}$  can be computed from sampling  $a_\lambda(s, \theta) \left\langle L \frac{1+s}{1-s} \right\rangle^{-d_a}$  on the tensor product grid with equal spacing in  $\theta$  and a one-dimensional Chebyshev grid in  $s \in [-1, 1]$ . It has been shown that, for a fixed accuracy  $\varepsilon$ , the number of grid points required is of order 1 (see Figure 1(a)).

- **Hierarchical spline interpolation.** This approach starts by partitioning the  $\xi$  domain into a hierarchy of squares. The size of a square grows linearly with respect to its distance from the origin. Within each box, we embed a local Cartesian grid and the function  $a_\lambda(\xi) \langle \xi \rangle^{-d_a}$  restricted to this box is approximated using a cubic spline. The basis functions  $g_\mu(\xi)$  are then the union of the spline basis functions over all boxes. It is clear from the spline construction that the computation of  $a_{\lambda\mu}$  only requires the values of  $a_\lambda(\xi) \langle \xi \rangle^{-d_a}$  at the local Cartesian grid within each box. It has been shown that, for a fixed  $\varepsilon$ , one needs  $O(\log N)$  hierarchical spline basis functions to approximate  $a_\lambda(\xi)$  on the domain  $\xi \in [-N/2, N/2]^2$  (see Figure 1(b)).

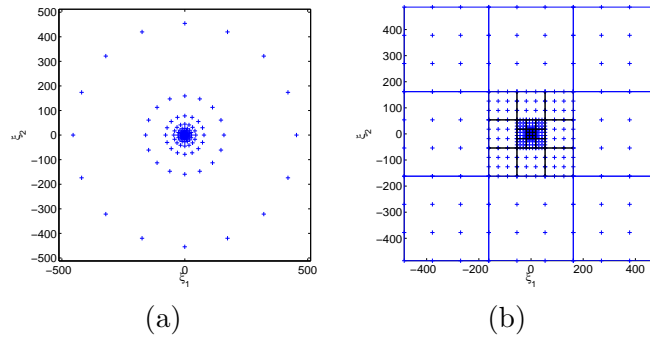


Figure 1: (a) Sampling points for the rational Chebyshev interpolation. (b) Sampling points for the hierarchical spline interpolation, where each box is equipped with a local Cartesian grid.

To summarize, in both approaches, we can approximate  $a(x, \xi)$  at  $x \in X$  and  $\xi \in \Omega$  with

$$a(x, \xi) \approx \sum_{\lambda \in \Lambda_B} e_\lambda(x) \left( \sum_{\mu} a_{\lambda\mu} g_\mu(\xi) \langle \xi \rangle^{d_a} \right)$$

and the computation of the coefficients  $a_{\lambda\mu}$  can be reduced to sampling  $a_\lambda(\xi) \langle \xi \rangle^{-d_a}$  (or equivalently  $a_\lambda(\xi)$ ) at at most  $O(\log N)$  locations in  $\xi$  for each  $\lambda \in \Lambda_B$ .

**Operations** We can carry out the standard operations of the pseudodifferential operators in the DSC framework. From the above discussion, we know that approximating an operator with symbol  $a(x, \xi)$  reduces to sampling  $a_\lambda(\xi)$  at a few locations in  $\xi$  for each  $\lambda \in \Lambda_B$ . Therefore, we focus on this task in the following discussion.

The symbol  $c(x, \xi)$  of the sum of two operators with symbols  $a(x, \xi)$  and  $b(x, \xi)$ , respectively, is given by

$$c(x, \xi) = a(x, \xi) + b(x, \xi).$$



To approximate  $c(x, \xi)$ , we set  $d_c = \max(d_a, d_b)$  and

$$c_\lambda(\xi) = a_\lambda(\xi) + b_\lambda(\xi).$$

The symbol  $c(x, \xi)$  of the product of two operators with symbols  $a(x, \xi)$  and  $b(x, \xi)$ , respectively, is equal to

$$c(x, \xi) = a(x, \xi) \# b(x, \xi) := \sum_{\eta} \int e^{-2\pi i(x-y)(\xi-\eta)} a(x, \eta) b(y, \xi) dy.$$

In order to approximate  $c(x, \xi)$ , we let  $d_c = d_a + d_b$  and

$$c_\lambda(\xi) = \sum_{\kappa, \ell \in \Lambda_B: \kappa + \ell = \lambda} a_\kappa(\xi + \ell) b_\ell(\xi).$$

Once we know how to multiply two operators, computing the inverse of an operator  $A$  can be done via a simple Schulz iteration.

- 1: Pick  $\alpha$  so that  $\|\alpha A\| < 1$ .
- 2: Set  $X_0 = I$ .
- 3: For  $k = 0, 1, \dots$  until convergence, set  $X_{k+1} = 2X_k - X_k(\alpha A)X_k$ .
- 4: The inverse is set to be  $\alpha X_k$ .

The square root and inverse square root of an operator  $A$  are computed using the following Schulz-Higham iteration [17].

- 1: Pick  $\alpha$  so that  $\|\alpha A\| < 1$ .
- 2: Set  $Y_0 = \alpha A$  and  $Z_0 = I$ .
- 3: For  $k = 0, 1, \dots$  until convergence, set

$$Y_{k+1} = \frac{1}{2}Y_k(3I - Z_k Y_k), \quad Z_{k+1} = \frac{1}{2}(3I - Z_k Y_k)Z_k.$$

- 4:  $\alpha^{-1/2}Y_k$  is the square root and  $\alpha^{1/2}Z_k$  is the inverse square root.

In these two algorithms,  $\|\alpha A\| < 1$  makes sense since all operators become bounded after discretization. For all operators  $A$  discussed in this paper,  $\alpha$  is of order  $O(1/N^{O(1)})$  and both algorithms converge in  $O(\log N)$  iterations.

**Applying the operator** Once the DSC representation of an operator with symbol  $a(x, \xi)$  is ready, applying it to an arbitrary function  $f(x)$  consists of the following steps:

$$\begin{aligned} \sum_{\xi \in \Omega} a(x, \xi) e^{2\pi i x \cdot \xi} \hat{f}(x) &= \sum_{\xi \in \Omega} \sum_{\lambda \in \Lambda_B} e_{\lambda}(x) a_{\lambda}(\xi) e^{2\pi i x \cdot \xi} \hat{f}(\xi) \\ &= \sum_{\lambda \in \Lambda_B} e_{\lambda}(x) \left( \sum_{\xi \in \Omega} e^{2\pi i x \cdot \xi} (a_{\lambda}(\xi) \hat{f}(\xi)) \right). \end{aligned}$$

As the formula in the parentheses is a Fourier transform and the cardinality of  $\Lambda_B$  is a constant, this computation can be performed in  $O(N^2 \log N)$  steps.

## 2.2 Randomized algorithm for low-rank factorization

The  $\varepsilon$ -rank of an  $m \times n$  matrix  $M$ , denoted by  $r_\varepsilon(M)$  or just  $r_\varepsilon$  if  $M$  is fixed, is the number of singular values of  $M$  that are greater than or equal to  $\varepsilon$ . We say  $M$  to be numerically low-rank if  $r_\varepsilon$  is much smaller than  $n$  even for  $\varepsilon$  very small. As we pointed out earlier, the symbols  $a_\pm(x, \xi, \tau)$  viewed as a matrix indexed by  $x \in X$  and  $\xi \in \Omega$  is numerically low-rank when  $\tau < t^*$ . The algorithm described below aims to construct a separated approximation of form

$$M \approx CDR$$

with accuracy  $O(\varepsilon)$ , where the number of columns in  $C$  and the number of rows in  $R$  are roughly  $r_\varepsilon$ . Here, we adopt the standard notation for submatrix: given a row index set  $I$  and a column index set  $J$ ,  $M(I, J)$  is the submatrix with entries from rows in  $I$  and columns in  $J$ .

1. Sample randomly a set of  $\beta r_\varepsilon$  rows and denote the index set by  $S = (s_i)$ . Here  $\beta$  is the oversampling factor. Perform pivoted QR decomposition to the matrix  $M(S, :)$  and obtain

$$M(S, P) = QR,$$

where  $P = (p_i)$  is the resulting permutation vector of the columns and  $R = (r_{ij})$  is upper triangular. Let  $k$  be the largest index such that  $r_{kk} \geq \varepsilon$ . Define the index set  $S_c$  to be  $\{p_1, \dots, p_k\}$ .

2. Sample randomly a set of  $\beta r_\varepsilon$  columns and denote the index set by  $S = (s_i)$ . Perform pivoted LQ decomposition on the rows of  $M(:, S)$ :

$$M(P, S) = LQ,$$

where  $P$  is the resulting permutation vector of the rows and  $L = (\ell_{ij})$  is lower triangular. Let  $k$  be the largest index such that  $\ell_{kk} \geq \varepsilon$ . Define the index set  $S_r$  to be  $\{p_1, \dots, p_k\}$ .

3. Perform pivoted QR decomposition on the columns of  $M(:, S_c)$  and pivoted LQ decomposition the rows of  $M(S_r, :)$  respectively:

$$M(:, S_c) \cdot P_c = Q_c R_c, \quad P_r \cdot M(S_r, :) = L_r Q_r,$$

where  $P_c$  and  $P_r$  are the resulting permutation matrices that reorder the columns of  $M(:, S_c)$  and the rows of  $M(S_r, :)$ , respectively.

4. We seek for a factorization of form  $M \approx Q_c \cdot D \cdot Q_r$ . In order to do that efficiently, we restrict ourselves to the rows in  $S_r$  and columns in  $S_c$  and solve the following problem:

$$\min_D \|M(S_r, S_c) - Q_c(S_r, :) \cdot D \cdot Q_r(:, S_c)\|_F.$$

A simple linear square solution gives  $D = (Q_c(S_r, :))^+ M(S_r, S_c) (Q_r(:, S_c))^+$ , where  $(\cdot)^+$  stands for the pseudoinverse. Therefore, the resulting factorization is

$$M \approx Q_c \cdot ((Q_c(S_r, :))^+ \cdot M(S_r, S_c) \cdot (Q_r(:, S_c))^+) \cdot Q_r.$$

Clearly this process provides us with a separated approximation  $M \approx CDR$  with

$$C = Q_c, \quad D = (Q_c(S_r, :))^+ \cdot M(S_r, S_c) \cdot (Q_r(:, S_c))^+, \quad R = Q_r.$$

In practice, setting oversampling factor  $\beta$  to 5 is sufficient for an accurate approximation. Notice that the most computationally intensive steps of this algorithm are pivoted QR decompositions on matrices of size  $m \times O(r_\varepsilon)$  and pivoted LQ decompositions on matrices of size  $O(r_\varepsilon) \times n$ . When  $\varepsilon$  is fixed and  $r_\varepsilon$  can be treated as a constant, the cost of this algorithm is only linear in  $\max(m, n)$ .

For matrices  $(a_\pm(x, \xi, \tau))_{x \in X, \xi \in \Omega}$  of this paper, one may replace the random samplings in the first two steps with deterministic sampling in  $X$  and  $\Omega$  because of the smoothness of  $a_\pm(x, \xi, \tau)$  both in  $x$  and in  $\xi$ . However, we prefer the randomly sampling approach since it works for more general scenarios where the rows and columns of the matrices are not smooth and the deterministic alternatives might require careful adaptive sampling such as the divide-and-conquer strategy.

### 2.3 Fast algorithms for applying FIOs

Given a function  $f(x)$  defined on a Cartesian grid  $X = \{(\frac{n_1}{N}, \frac{n_2}{N}) : n_1, n_2 \in \mathbb{Z}, 0 \leq n_1, n_2 < N\}$ , the discrete Fourier transform  $\hat{f}(\xi)$  is defined by

$$\hat{f}(\xi) = \frac{1}{N^2} \sum_{x \in X} e^{-2\pi i x \cdot \xi} f(x)$$

for  $\xi \in \Omega = \{(\xi_1, \xi_2) : \xi_1, \xi_2 \in \mathbb{Z}, -\frac{N}{2} \leq \xi_1, \xi_2 < \frac{N}{2}\}$ . The discrete Fourier integral operator with phase function  $\Phi(x, \xi)$  and amplitude  $a(x, \xi)$  is defined by

$$u(x) = (Af)(x) := \sum_{\xi \in \Omega} a(x, \xi) e^{2\pi i \Phi(x, \xi)} \hat{f}(\xi)$$

for  $x \in X$ . We assume that  $\Phi(x, \xi)$  is smooth in  $x$  and in  $\xi$  away from the origin, and  $a(x, \xi)$  has a low-rank separated approximation. Along with E. Candès, we have developed two efficient algorithms for computing  $u(x)$  in [7] and later in [8].

**Approach 1: Angular partitioning of the frequency domain** This approach [7] is based on a parabolic angular partitioning of the frequency domain. Let  $\arg(\xi)$  be the angle between  $\xi$  and the horizontal vector  $(1, 0)$ . Assuming that  $\sqrt{N}$  is an integer, we partition the frequency domain into a family of angular wedges  $\{W_\ell\}$  defined by

$$W_\ell = \{\xi : (2\ell - 1)\pi/\sqrt{N} \leq \arg(\xi) < (2\ell + 1)\pi/\sqrt{N}\}$$

for  $0 \leq \ell < \sqrt{N}$  (see Figure 2).

For each wedge  $W_\ell$ , we define  $\chi_\ell(\xi)$  to be the indicator function of  $W_\ell$  and  $\hat{\xi}_\ell$  to be the (unit) center direction of  $W_\ell$ . For a fixed  $\ell$ , the phase function can be written as the sum of two parts:

$$\Phi(x, \xi) = \nabla_\xi \Phi(x, \hat{\xi}_\ell) \cdot \xi + \Phi_\ell(x, \xi),$$

where the term  $\Phi_\ell(x, \xi)$  is called the residual phase of  $W_\ell$ . Using this decomposition, we can write

$$a(x, \xi) e^{2\pi i \Phi(x, \xi)} = \left( e^{2\pi i \nabla_\xi \Phi(x, \hat{\xi}_\ell) \cdot \xi} \right) \cdot \left( a(x, \xi) e^{2\pi i \Phi_\ell(x, \xi)} \right).$$

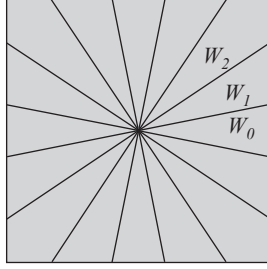


Figure 2: The frequency domain  $[-N/2, N/2]^2$  is partitioned into  $\sqrt{N}$  equiangular wedges  $\{W_\ell\}$ .

When  $\xi \in W_\ell$ , it turns out that  $\Phi_\ell(x, \xi)$  is of order  $O(1)$  and thus  $e^{2\pi i \Phi_\ell(x, \xi)}$  is non-oscillatory. Since  $a(x, \xi)$  is also assumed to have a separated approximation, one can show that the term  $a(x, \xi)e^{2\pi i \Phi_\ell(x, \xi)}$  also has a separated approximation:

$$a(x, \xi)e^{2\pi i \Phi_\ell(x, \xi)} \approx \sum_t \alpha_{\ell t}(x) \beta_{\ell t}(\xi). \quad (2.1)$$

This approximation further implies that

$$\begin{aligned} (Af)(x) &\approx \sum_\ell \sum_\xi e^{2\pi i \Phi(x, \xi)} a(x, \xi) \chi_\ell(\xi) \hat{f}(\xi) \\ &= \sum_\ell \sum_\xi e^{2\pi i \nabla_\xi \Phi(x, \hat{\xi}_\ell) \cdot \xi} a(x, \xi) e^{2\pi i \Phi_\ell(x, \xi)} \chi_\ell(\xi) \hat{f}(\xi) \\ &= \sum_\ell \sum_\xi e^{2\pi i \nabla_\xi \Phi(x, \hat{\xi}_\ell) \cdot \xi} \sum_t \alpha_{\ell t}(x) \beta_{\ell t}(\xi) \chi_\ell(\xi) \hat{f}(\xi) \\ &= \sum_\ell \left( \sum_t \alpha_{\ell t}(x) \left( \sum_\xi e^{2\pi i \nabla_\xi \Phi(x, \hat{\xi}_\ell) \cdot \xi} \left( \beta_{\ell t}(\xi) \chi_\ell(\xi) \hat{f}(\xi) \right) \right) \right). \end{aligned}$$

In the last formula, the sum over  $\xi$  is in fact a Fourier transform of  $(\beta_{\ell t}(\xi) \chi_\ell(\xi) \hat{f}(\xi))$  at locations  $\nabla_\xi \Phi(x, \hat{\xi}_\ell)$  for  $x \in X$ . This can be computed easily using the non-uniform fast Fourier transform [15] in  $O(N^2 \log N)$  steps. The sum over  $t$  involves only a constant number of terms since the rank of the separated approximation is constant. Finally, since there are only  $N^{1/2}$  wedges  $W_\ell$ , the overall complexity of the algorithm is  $O(N^{2.5} \log N)$ .

**Approach 2: Butterfly strategy** This approach [8] starts with parameterization  $\xi = (\xi_1, \xi_2)$  with polar coordinates  $p = (p_1, p_2)$ :  $(\xi_1, \xi_2) = \frac{\sqrt{2}}{2} N p_1 e^{2\pi i p_2}$ . Using this transformation, we introduce a new phase function  $\Psi(x, p)$  in the polar coordinates:

$$\Psi(x, p) := \frac{1}{N} \Phi(x, \xi) = \frac{\sqrt{2}}{2} \Phi(x, e^{2\pi i p_2}) p_1.$$

Denote by  $P \subset [0, 1]^2$  the set of all possible points  $p$  generated by  $\xi \in \Omega$ . The Fourier integral operator then takes the following equivalent form

$$u(x) = \sum_{p \in P} a(x, \xi(p)) e^{2\pi i N \Psi(x, p)} \hat{f}(\xi(p)).$$

The main observation is that the kernel  $e^{2\pi i N \Psi(x,p)}$  is approximately low-rank when restricted to appropriate spatial and frequency regions. More precisely, suppose that  $A$  and  $B$  are two squares in  $x$  and  $p$ , respectively, with widths  $w^A$  and  $w^B$  and centers  $c^A$  and  $c^B$ . We assume that  $w^A w^B \leq 1/N$  and define

$$R^{AB}(x,p) = \Psi(x,p) - \Psi(c^A,p) - \Psi(x,c^B) + \Psi(c^A,c^B).$$

Using the smoothness of  $\Psi(x,p)$  in both  $x$  and  $p$ , one can show that  $R^{AB}(x,p) = O(1/N)$  for  $x \in A$  and  $p \in B$ . As a result,  $e^{2\pi i R^{AB}(x,p)}$  is not oscillatory and has a low-rank separated approximation in  $x$  and  $p$ . Similarly, using the identity

$$e^{2\pi i \Psi(x,p)} = e^{2\pi i \Psi(c^A,p)} e^{2\pi i \Psi(x,c^B)} e^{-2\pi i \Psi(c^A,c^B)} e^{2\pi i R^{AB}(x,p)},$$

we see that the kernel  $e^{2\pi i \Psi(x,p)}$  also enjoys a similar low-rank separated approximation in  $x$  vs.  $p$ . The actual separated approximation is implemented based on the idea of oscillatory Chebyshev interpolations. For a fixed  $\varepsilon$ , we define  $\{x_t^A\}$  and  $\{p_t^B\}$  to be the Chebyshev grids adapted to  $A$  and  $B$ , respectively. We also denote the standard Lagrange interpolants of these grids by  $\{L_t^A(x)\}$  and  $\{L_t^B(p)\}$ , respectively. It is proved in [8] that the size of the Chebyshev grids grows at most polylogarithmically in  $O(1/\varepsilon)$ . Consider an admissible pair of squares  $(A,B)$  with  $w^A w^B \leq 1/N$ . When  $w^B \leq 1/\sqrt{N}$ , we have

$$|e^{2\pi i N \Psi(x,p)} - \sum_t e^{2\pi i N \Psi(x,p_t^B)} S_t^{AB}(p)| \leq \varepsilon,$$

where  $S_t^{AB}(p)$  relate to the standard Chebyshev interpolant  $L_t^B(p)$  on the grid  $p_t^B$  via

$$S_t^{AB}(p) = e^{-2\pi i N \Psi(c^A,p_t^B)} L_t^B(p) e^{2\pi i N \Psi(c^A,p)}.$$

Similarly, when  $w^A \leq 1/\sqrt{N}$ , we have

$$|e^{2\pi i N \Psi(x,\xi)} - \sum_t T_t^{AB}(x) e^{2\pi i N \Psi(x_t^A,p)}| \leq \varepsilon,$$

where  $T_t^{AB}(x)$  relate to the standard Chebyshev interpolant  $L_t^A(p)$  via

$$T_t^{AB}(x) = e^{2\pi i N \Psi(x,c^B)} L_t^A(x) e^{-2\pi i N \Psi(x_t^A,c^B)}.$$

Let us define  $u^B(x) = \sum_{p \in B} e^{2\pi i N \Psi(x,p)} \hat{f}(\xi(p))$  to be the partial sum with  $p$  restricted to  $B$ . The separated approximations imply that there exists a compact representation for  $\{u^B(x), x \in A\}$ : when  $w^B \leq 1/\sqrt{N}$ ,

$$u^B(x) \approx \sum_t e^{2\pi i N \Psi(x,p_t^B)} \delta_t^{AB} \quad \text{if} \quad \delta_t^{AB} \approx \sum_{p \in B} S_t^{AB}(p) \hat{f}(\xi(p)) \quad (2.2)$$

and when  $w^A \leq 1/\sqrt{N}$ ,

$$u^B(x) \approx \sum_t T_t^{AB}(x) \delta_t^{AB} \quad \text{if} \quad \delta_t^{AB} \approx \sum_{p \in B} e^{2\pi i N \Psi(x_t^A,p)} \hat{f}(\xi(p)). \quad (2.3)$$

It is not difficult to see that  $\delta_t^{AB}$  serve as equivalent sources in the first case and approximates  $u^B(x_t^A)$  in the second one.

Combining these observations with the structure of the Butterfly algorithm [21, 23], we have the following algorithm for applying the Fourier integral operators. Essentially, this computes  $\{\delta_t^{AB}\}$  for all admissible pairs  $(A,B)$  efficiently in a recursive fashion.

1. Construct quadtrees  $\mathcal{T}_X$  and  $\mathcal{T}_P$  for  $X$  and  $P$  respectively. Both of them has  $[0, 1]^2$  as the root box at level 0 and have leaf boxes of width  $1/N$ . Let  $L$  denote the total number of levels.

2. Let  $A$  be the root of  $\mathcal{T}_X$ . For each leaf box  $B$  of  $\mathcal{T}_P$ , construct

$$\delta_t^{AB} = \sum_{p \in B} S_t^{AB}(p) \hat{f}(\xi(p)).$$

3. For level  $\ell = 1, 2, \dots, L/2$ , construct  $\delta_t^{AB}$  for all pairs  $(A, B)$  where  $A$  at level  $\ell$  and  $B$  is at level  $L - \ell$ . Denote by  $A_p$  the parent of  $A$  and by  $\{B_c\}$  the set of the children of  $B$ :

$$\delta_t^{AB} = \sum_c \sum_{t'} S_t^{AB}(p_{t'}^{B_c}) \delta_{t'}^{A_p B_c}.$$

4. For each pair  $(A, B)$  with where  $A$  at level  $L/2$  and  $B$  is at level  $L/2$ , compute

$$\delta_t^{AB} = \sum_{t'} a(x_t^A, p_{t'}^B) e^{2\pi i \Psi(x_t^A, p_{t'}^B)} \delta_{t'}^{A_p B_c}.$$

(This is the only place where the amplitude enters. Note that we had assumed  $a = 1$  in [8].)

5. For level  $\ell = L/2 + 1, \dots, L$ , construct  $\delta_t^{AB}$  for all pairs  $(A, B)$  where  $A$  at level  $\ell$  and  $B$  is at level  $L - \ell$ .

$$\delta_t^{AB} = \sum_c \sum_{t'} T_{t'}^{A_p B_c}(x_t^A) \delta_{t'}^{A_p B_c}.$$

6. Let  $B$  be the root of  $\mathcal{T}_P$ . For each leaf box  $A$  and each  $x \in A$ , set

$$u(x) = \sum_t T_t^{AB}(x) \delta_t^{AB}.$$

For each level  $\ell$ , there are  $N^2$  pairs of admissible boxes  $(A, B)$  with  $w^A w^B = 1/N$ . Since we perform  $O(1)$  steps for constructing each set of coefficients  $\{\delta_t^{AB}\}$ , the number of steps used for each level  $\ell$  is  $O(N^2)$ . Since there are at most  $O(\log N)$  levels, this algorithm takes at most  $O(N^2 \log N)$  steps.

### 3 Theory

In this section we present the justification of the FIO formula (1.3). Such oscillatory integrals are very well-known and go back at least to Lax [20]. However, they are usually introduced as *parametrices*, i.e., asymptotic high-frequency approximations to the solution operator. The same is true for the polarizers  $\pm P$  that form the one-way components  $f_{\pm}(x)$  in equation (1.2). In Section 3.1, we study the operator  $P^{-1}$  and show that it is exactly pseudodifferential. In Section 3.2 we make the transition to an exact oscillatory integral representation of the propagators  $e^{\pm itP}$  in free space. In Section 3.3, we extend both characterizations to the case of the torus, where Fourier series replace the Fourier transform, and where differentiations in the wave number domain become finite differences.

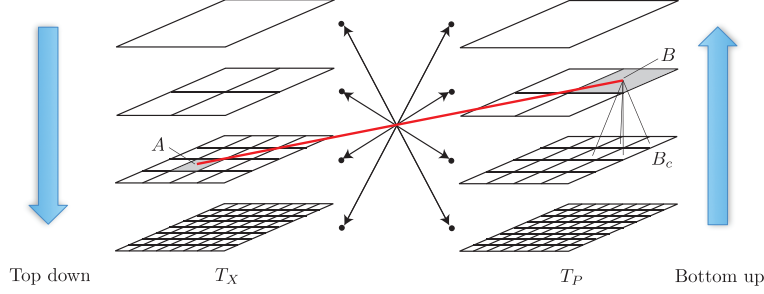


Figure 3: Schematic illustration of the butterfly algorithm in 2D with 4 levels ( $L = 3$ ). The levels are paired as indicated so that the product of the sidelengths remains constant. The algorithm starts at the root of  $\mathcal{T}_X$  and at the bottom of  $\mathcal{T}_P$ . It then traverses  $\mathcal{T}_X$  top down and  $\mathcal{T}_P$  bottom up, and terminates when the last level (the bottom of  $\mathcal{T}_X$ ) is reached. The figure also represents the four children of any box  $B$ .

The wave equation  $u_{tt} = \nabla \cdot c^2(x)\nabla u$  can be written as the system

$$\frac{\partial}{\partial t} \begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{L} \begin{pmatrix} u \\ v \end{pmatrix}, \quad (3.1)$$

where  $\mathcal{L}$  is the matrix of operators

$$\mathcal{L} = \begin{pmatrix} 0 & I \\ \nabla \cdot c^2(x)\nabla & 0 \end{pmatrix}.$$

By letting  $P$  be the positive semidefinite square root of  $L = -\nabla \cdot c^2(x)\nabla$ , we obtain the formal spectral factorization

$$\mathcal{L} = \begin{pmatrix} 0 & I \\ (iP)^2 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} I & I \\ iP & -iP \end{pmatrix} \begin{pmatrix} iP & 0 \\ 0 & -iP \end{pmatrix} \begin{pmatrix} I & (iP)^{-1} \\ I & -(iP)^{-1} \end{pmatrix}.$$

As a result, the propagator obeys

$$e^{t\mathcal{L}} = \frac{1}{2} \begin{pmatrix} I & I \\ iP & -iP \end{pmatrix} \begin{pmatrix} e^{itP} & 0 \\ 0 & e^{-itP} \end{pmatrix} \begin{pmatrix} I & (iP)^{-1} \\ I & -(iP)^{-1} \end{pmatrix}.$$

This equation is another expression of the decoupling into one-way components as in Section 1: the matrix of operators on the right forms the “polarized” components, the middle matrix evolves them, and the left matrix recomposes them.

### 3.1 The polarizers are pseudodifferential operators

In this section we define  $P$  and  $P^{-1}$ , and show that both belong to a class of operators with pseudodifferential symbols of so-called classical type. Throughout this paper, the pseudodifferential representation of almost any linear operator  $P$  is

$$Pu(x) = \int e^{ix \cdot \xi} p(x, \xi) \hat{u}(\xi) d\xi,$$

where  $p(x, \xi)$  is called the symbol. This formula always makes sense for  $x \in \mathbb{R}^2$  and in the scope of the Schwartz theorem  $p(x, \xi)$  always exists as a distributional kernel. When  $x \in$

$[0, 1]^2$  however, the dual Fourier variable  $\xi$  is continuous and does *not* result from Fourier series transformation. Instead, it is the usual Fourier transform used in patches, after taking a partition of unity  $\phi_j$  on the torus. The operator  $P$  is then itself partitioned as  $\phi_i P \phi_j$ : the pieces are studied independently by means of pseudodifferential symbols, and later recombined by summation over  $i$  and  $j$ . We will thus feel free to let  $x \in [0, 1]^2$  in this section and ignore this partitioning in the notations. See [33, 18, 28] for complete background on the definition of pseudodifferential operators on manifolds. Note in passing that nothing in what follows depends on the fact that we only consider two spatial dimensions.

Symbols generally have a lot of structure.

**Definition 3.1.** (*Smoothing*) A linear operator  $R$  acting inside the space of tempered distributions in  $[0, 1]^2$  is called *smoothing* when it has a bounded pseudodifferential symbol  $r(x, \xi)$  such that, for all integer  $M > 0$  (arbitrarily large), and for all  $r > 0$  (arbitrarily small),

$$|\partial_\xi^\alpha \partial_x^\beta r(x, \xi)| \leq C_{Mr} (1 + |\xi|^2)^{-M/2}, \quad |\xi| > r.$$

Such a symbol  $r(x, \xi)$  decays faster than any negative power of  $|\xi|$  as  $|\xi| \rightarrow \infty$ ; its corresponding operator can be shown to map tempered distributions to  $C^\infty$  functions.

**Definition 3.2.** (*Classical symbol, [33, 18]*) A symbol  $p(x, \xi)$  is called *classical of order  $m$* , denoted  $p \in \Psi_{cl}^m$ , when

$$p(x, \xi) \sim \sum_{j \geq 0} p_{m-j}(x, \xi),$$

where  $p_n(x, \xi)$  is *positive-homogeneous of order  $n$  in  $\xi$* , i.e.,

$$p_n(x, \lambda \xi) = \lambda^n p_n(x, \xi), \quad \lambda > 0.$$

In addition,  $p_n(x, \xi)$  is required to be  $C^\infty$  in  $x$  and in the angular  $\xi$  variables, with uniform smoothness constants. The notation  $\sim$  means that there exists a  $C^\infty$  cutoff function  $\chi(x, \xi)$  such that  $\chi = 0$  in a neighborhood of  $\xi = 0$ , and  $\chi = 1$  outside of a larger neighborhood of  $\xi = 0$ , and there exist numbers  $\epsilon_j > 0$  (possibly tending to zero as  $j \rightarrow \infty$ ) such that

$$r(x, \xi) = p(x, \xi) - \sum_{j \geq 0} \chi(x, \epsilon_j \xi) p_{m-j}(x, \xi)$$

is the symbol of a smoothing operator in the sense of the previous definition.

A classical symbol needs not be  $C^\infty$  at the origin in  $\xi$ . It is well-known that the classical condition above implies the smoothness condition

$$|\partial_\xi^\alpha \partial_x^\beta p(x, \xi)| \leq C_{\alpha\beta r} (1 + |\xi|^2)^{m/2 - |\alpha|/2}, \quad |\xi| > r. \quad (3.2)$$

Functions of operators, when they make sense, can be defined by means of the spectral theorem. All the operators of interest in this paper have discrete spectra<sup>1</sup>.

**Definition 3.3.** Let  $L$  have the spectral decomposition

$$L = \sum_j \lambda_j E_j,$$

---

<sup>1</sup>The Rellich embedding theorem ensures that any invertible operators on a compact manifold like the torus have discrete spectra.



where the  $E_j$  are projection operators onto the eigenspaces with eigenvalues  $\lambda_j$ . Then, for any function  $f$  whose domain includes the spectrum of  $L$ ,

$$f(L) = \sum_j f(\lambda_j) E_j,$$

(with strong  $L^2$ -operator convergence on functions  $g$  such that  $\sum_j f^2(\lambda_j) \|E_j g\|_2^2 < \infty$ .)

For instance, the operator of interest in this paper,  $L = -\nabla \cdot c^2(x) \nabla$  on the torus, is positive semi-definite. It has a nullspace associated with the constant eigenfunction, and an otherwise discrete set of positive eigenvalues.

The square-root and the inverse square-root of a positive-definite operator are well-defined operations. The following classical result is due to Seeley [26], and its proof can also be found in [28], p.110. For background, see also [33], Vol.3, pages leading up to p.43.

**Theorem 3.4.** *Consider a positive-definite operator  $T \in \Psi_{cl}^2$ . Then  $T^{1/2}$  and  $T^{-1/2}$ , as defined by the spectral theorem, have classical symbol of orders 1 and  $-1$  respectively.*

The operator  $L = -\nabla \cdot c^2(x) \nabla$  is not positive definite because it has a nullspace. Instead, the theorem should be applied to  $T = \mathbb{P}_0 + L$ , where  $\mathbb{P}_0$  is the orthogonal projector onto constants,

$$\mathbb{P}_0 g(x) = \int_{[0,1]^2} g(x) dx.$$

This modification of  $L$  only changes the symbol at the origin, hence does not change the classical character of the symbol. Once  $T^{1/2}$  and  $T^{-1/2}$  are obtained,

- $L^{1/2} = T^{1/2} - \mathbb{P}_0$  also has a symbol in the class  $\Psi_{cl}^1$ ;
- $L^{-1/2} = T^{-1/2}$  as long as the function  $g$  to which it is applied has a zero mean, i.e.,  $\mathbb{P}_0 g = 0$ . This is why care has been taken to assume that  $\hat{u}_1(0) = \mathbb{P}_0 u_1 = 0$ . The initial condition  $u_1$  is the only function to which  $P^{-1}$  is applied. So when restricted to functions of mean zero,  $L^{-1/2}$  makes sense and has a symbol in  $\Psi_{cl}^{-1}$ .

As a conclusion,  $P$  and  $P^{-1}$  are pseudodifferential operators with classical symbols, with the proviso that  $P^{-1}$  is only applied to mean zero functions. The remaining question of characterizing the symbols of  $P$  and  $P^{-1}$  in the native Fourier series variables on the torus will be addressed in Section 3.3.

## 3.2 The propagators are of Fourier integral type

In this section we show that  $e^{\pm itP}$ , the solution operators to the pseudodifferential equations  $u_t = \pm iPu$ , are Fourier integral operators.

The following classical result hinges on the property that  $p(x, \xi)$ , the symbol of  $P$ , is in the class  $\Psi_{cl}^1$ . It was perhaps first formulated by Lax [20]. More modern formulations with precise estimates on the remainder such as the one below require some microlocal analysis that can be found at least in [18, 28, 33].

**Theorem 3.5.** *(FIO parametrix) Assume  $P$  has a symbol in the class  $\Psi_{cl}^1$ . There exists  $T > 0$  such that for all  $0 \leq t < T$ , there exist scalar phase functions  $\Phi_{\pm}$ , and scalar amplitude functions  $a_{\pm}$  such that*

$$(e^{\pm itP} f)(x) = \int_{\mathbb{R}^n} e^{i\Phi_{\pm}(x, \xi; t)} a_{\pm}(x, \xi; t) \hat{f}(\xi) d\xi + (R(t)f)(x). \quad (3.3)$$

Moreover, as long as  $t \in [0, T)$ ,  $\Phi_{\pm}$  obey the Hamilton-Jacobi equations

$$\frac{\partial \Phi_{\pm}}{\partial t}(x, \xi; t) = \lambda_{\pm}(x, \nabla_x \Phi_{\pm}(x, \xi; t)), \quad \Phi_{\pm}(x, \xi; 0) = x \cdot \xi,$$

and  $a_{\pm}$  obey the smoothness estimate

$$|\partial_{\xi}^{\alpha} \partial_x^{\beta} a_{\pm}(x, \xi; t)| \leq C_{\alpha\beta r} (1 + |\xi|^2)^{-|\alpha|/2}, \quad |\xi| \geq r. \quad (3.4)$$

For each  $0 \leq t < T$ ,  $R(t)$  is a smoothing operator in the sense of Definition 3.1.

This result is local in time:  $T$  cannot be taken larger than the time at which caustics appear from planar wavefronts. This geometrical situation occurs when, in time  $T$ , two different points initially on the same straight line are mapped to a single point by the Hamiltonian flow generated by either  $\lambda_{\pm}$ . The smoother the medium  $c^2(x)$  the larger  $T$ .

Mathematicians usually specify the amplitudes  $a_{\pm}$  in the above result as asymptotic expansions for  $|\xi| \rightarrow \infty$ , or equivalence classes of highest-order symbols. Such characterizations may be sufficient as proof techniques, but they are inadequate as numerical expansion schemes.

The operators  $R(t)$  are smoothing, but they are not small and cannot be ignored numerically. The discussion in this section deals with amplitudes that are not necessarily  $C^{\infty}$  near the origin; it is this precise point which allows to absorb the remainder  $R(t)$  in the amplitude of the oscillatory integral.

**Corollary 3.6.** *In Theorem 3.5,  $R(t)$  can be taken to be zero.*

*Proof.* Let  $R(t)$  be the original smoothing remainder obtained from Theorem 3.5,

$$(R(t)) f(x) = \int e^{ix \cdot \xi} r(x, \xi; t) \hat{f}(\xi) d\xi,$$

with

$$|\partial_{\xi}^{\alpha} \partial_x^{\beta} r(x, \xi; t)| \leq C_{Mr} (1 + |\xi|^2)^{-M/2}, \quad |\xi| > r, \quad \forall M > 0.$$

This gives rise to the composite expression

$$(e^{\pm itP} f)(x) = \int_{\mathbb{R}^n} e^{-i\Phi_{\pm}(x, \xi; t)} b_{\pm}(x, \xi; t) \hat{f}(\xi) d\xi,$$

with

$$b_{\pm}(x, \xi; t) = a_{\pm}(x, \xi; t) + r(x, \xi; t) e^{i(x \cdot \xi - \Phi_{\pm}(x, \xi; t))}.$$

It suffices to show that the second term in the right-hand side is a smoothing amplitude.

This term is to be differentiated in  $x$  and in  $\xi$ . In the Leibniz formula, every term will have either  $r(x, \xi; t)$  or one of its derivatives as a factor. A contribution  $(1 + |\xi|^2)^{-M/2}$  for all  $M > 0$  results in the bound – in short,  $(1 + |\xi|^2)^{-\infty}$ . The other factors are derivatives of the exponential: they can only grow polynomially in  $\xi$ . This growth will not undo the super-algebraic decay of  $(1 + |\xi|^2)^{-M/2}$ , only the value of the overall constant  $C_M$ .  $\square$

### 3.3 Analysis on the torus

The analysis in the previous sections, and in particular equations (3.2), (3.3), and (3.4), assumed that continuous Fourier transforms were taken in coordinate charts for the torus. In this section it is shown how these various results extend to the more natural setting of discrete  $\xi \in \mathbb{Z}^2$  coming from taking Fourier series. It is also the approach followed in the algorithms.

Without loss of generality the coordinate charts can be taken to be isometric maps resulting from the canonical embedding  $[0, 1]^2 \subset \mathbb{R}^2$ : in this scenario the symbol  $x$  is overloaded as a coordinate in both  $[0, 1]^2$  and  $\mathbb{R}^2$ . Periodization is one way to relate the solution of the wave equation on  $[0, 1]^2$ , with periodic boundary conditions, to that of the wave equation on  $\mathbb{R}^2$ .

The periodic extension of a wavefield from  $[0, 1]^2$  to  $\mathbb{R}^2$  is

$$u(x, t) = u(x \bmod [0, 1]^2, t), \quad x \in \mathbb{R}^2.$$

The medium parameter is similarly periodized as

$$c^2(x) = c^2(x \bmod [0, 1]^2).$$

A window function  $\rho(x), x \in \mathbb{R}^2$ , is then chosen such that its Fourier transform  $\hat{\rho}$  obeys

- $\hat{\rho} \in C^\infty(\mathbb{R}^2)$ ;
- $\hat{\rho}(0) = 1$ ;
- $\text{supp } \hat{\rho} \subset B_0(1/2)$ , the ball centered at the origin with radius 1/2.

Owing to the property that  $\hat{\rho}(n) = \delta_{0n}, n \in \mathbb{Z}^2$ , it follows that the integer translates of  $\rho$  form a partition of unity:

$$\sum_{n \in \mathbb{Z}^2} \rho(x - n) = 1.$$

Let us now consider the wavefield  $w(x, t)$  solution of

$$w_{tt} - \nabla \cdot c^2(x) \nabla w = 0, \quad x \in \mathbb{R}^2,$$

with initial conditions

$$w(x, 0) = \rho(x)u_0(x), \quad w_t(x, 0) = \rho(x)u_1(x).$$

By linearity,

$$u(x, t) = \sum_{n \in \mathbb{Z}^2} w(x - n, t), \tag{3.5}$$

with pointwise convergence.

The important property that  $\int_{[0,1]^2} u_1(x) dx = 0$  is preserved at the level of  $w$  in the sense that

$$\widehat{\rho u_1}(\xi) = 0, \quad \xi \in B_0(1/2).$$

We now address the two related questions of how to express pseudodifferential and Fourier integral operators in the native Fourier variable  $\xi \in \mathbb{Z}^2$  on the torus. If  $g(x)$  is

a  $[0, 1]^2$ -periodic function, it makes sense to apply a pseudodifferential operator  $T$  to the fast-decaying function  $h_n(x) = \rho(x - n)g(x)$  of  $x \in \mathbb{R}^2$ , yielding

$$Th_n(x) = \int_{\mathbb{R}^2} e^{2\pi i x \cdot \xi} \sigma(x, \xi) \hat{h}_n(\xi) d\xi, \quad x \in \mathbb{R}^2,$$

where  $\sigma(x, \xi)$  is the amplitude of  $T$ , and “hat” is the Fourier transform. The action of  $T$  on  $g$  itself can then be defined as

$$Tg(x) = \lim_{\varepsilon \rightarrow 0} \sum_{n \in \mathbb{Z}^2} e^{-\varepsilon n^2} (Th_n)(x), \quad x \in [0, 1]^2.$$

The problem is to find an amplitude  $a(x, \xi)$ ,  $\xi \in \mathbb{Z}^2$ , such that

$$Tg(x) = \sum_{\xi \in \mathbb{Z}^2} e^{2\pi i x \cdot \xi} \tau(x, \xi) \hat{g}(\xi) d\xi,$$

where the hat now denotes the Fourier series,

$$\hat{g}(\xi) = \int_{[0, 1]^2} e^{-2\pi i x \cdot \xi} g(x) dx, \quad \xi \in \mathbb{Z}^2.$$

Elementary manipulations show that  $\tau(x, \xi)$  are nothing but the samples of  $\sigma(x, \xi)$ :

$$\tau(x, \xi) = \sigma(x, \xi), \quad x \in [0, 1]^2, \quad \xi \in \mathbb{Z}^2.$$

The smoothness properties of  $\sigma$  are therefore transferred at the discrete level: *discrete symbols are samples of their very smooth counterparts*. This is chiefly true of the polarizer  $T = (iP)^{-1}$ , as far as our application is concerned.

Similarly, we seek to write the propagators as *Fourier series operators*:

$$(e^{\pm itP} f_{\pm})(x) = \sum_{\xi \in \mathbb{Z}^2} e^{i\Phi_{\pm}(x, \xi; t)} b_{\pm}(x, \xi; t) \hat{\alpha}_{\pm}(\xi).$$

The phase functions  $\Phi_{\pm}$  are the same as previously, and the polarized components are

$$f_{\pm}(x) = \frac{1}{2} [u_0(x) \pm (iP)^{-1} u_1(x)], \quad x \in [0, 1]^2,$$

where we have just made sense of  $P^{-1}$ . An analysis very similar to the pseudodifferential case shows that the amplitudes  $b_{\pm}(x, \xi; t)$  are also samples of the amplitudes for the propagator in  $\mathbb{R}^2$ :

$$b_{\pm}(x, \xi; t) = a_{\pm}(x, \xi; t), \quad x \in [0, 1]^2, \quad \xi \in \mathbb{Z}^2.$$

It was the subject of research by Turunen et al. over the past few years to formulate a smoothness criterion native to the torus for symbols such as  $\tau(x, \xi)$  and  $b_{\pm}(x, \xi; t)$ . In [34, 24] the authors propose to replace partial derivatives by finite differences in  $\xi$ . The forward difference operator  $\Delta_{\xi}^{\alpha}$  with multi-index  $\alpha$  is defined as

$$\begin{aligned} \Delta_{\xi}^{\alpha} &= \Delta_{\xi_1}^{f_1} \Delta_{\xi_2}^{f_2}, \\ (\Delta_{\xi_j} f)(\xi) &= f(\xi + \delta_j) - f(\xi), \quad j = 1, 2, \end{aligned}$$

with obvious modifications in several spatial dimensions. Here  $\delta_1 = (1, 0)$  and  $\delta_2 = (0, 1)$  are Kronecker deltas. Is it explained how to formulate a notion of calculus (via boundedness in certain Sobolev scales) based on these finite differences. In the scope of this discussion it will suffice to observe that the following simple result holds.

**Proposition 3.7.** *Let  $p(x, \xi)$  be a function of  $x \in [0, 1]^2$  and  $\xi \in \mathbb{R}^2$ . If there exists  $m > 0$ , and for all  $r > 0$  and multi-indices  $\alpha, \beta$  there exists  $C_{\alpha\beta r} > 0$  such that*

$$|\partial_\xi^\alpha \partial_x^\beta p(x, \xi)| \leq C_{\alpha\beta r} (1 + |\xi|^2)^{m/2 - |\alpha|/2}, \quad |\xi| > r, \quad (3.6)$$

and such that  $\partial_x^\beta p(x, \xi)$  is bounded on any compact set, then it also holds that

$$|\Delta_\xi^\alpha \partial_x^\beta p(x, \xi)| \leq C'_{\alpha\beta r} (1 + |\xi|^2)^{m/2 - |\alpha|/2}, \quad |\xi| > r, \quad \xi \in \mathbb{Z}^2,$$

for a possibly different constant  $C'_{\alpha\beta r}$ .

*Proof.* Fix  $\alpha$  and  $\beta$ . Let  $f(\xi) = \partial_x^\beta p(x, \xi)$  for simplicity of notation. Then

$$\Delta_{\xi_j}^{f_j} f(\xi) = \int_{[0,1]^{f_j}} \frac{\partial^{f_j} f}{\partial \xi_j^{f_j}}(\xi + |s| \delta_j) ds,$$

with  $|s| = \sum_{i=1}^{f_j} s_i$  and  $ds = ds_1 \dots ds_{f_j}$ . As a result,

$$|\Delta_\xi^f f(\xi)| = |\Delta_{\xi_1}^{f_1} \Delta_{\xi_2}^{f_2} f(\xi)| \leq \max_{|\eta| \leq |\alpha|} \left| \frac{\partial^{|\alpha|} f}{\partial \xi_1^{f_1} \partial \xi_2^{f_2}}(\xi + \eta) \right|, \quad (3.7)$$

with  $|\alpha| = f_1 + f_2$ . Two regimes must be contrasted:

- Near the origin, consider  $\xi$  such that  $|\xi| \leq |\alpha| + 1$ . Let  $D_\beta$  be the maximum of  $\partial_x^\beta p(x, \xi)$  over the larger ball  $|\xi| \leq 2|\alpha| + 1$ . We may simply take all the terms in absolute value in the finite difference expansion and obtain

$$|\Delta_\xi^f f(\xi)| \leq 2^{|\alpha|} D_\beta, \quad |\xi| \leq |\alpha| + 1.$$

- Consider  $|\xi| > |\alpha| + 1$ . Then  $|\xi + \eta| > 1$  when  $|\eta| \leq |\alpha|$ , and we may invoke equation (3.6) with  $r = 1$ . (W.l.o.g.  $C_{\alpha\beta r}$  decreases as a function of increasing  $r$ .) The right-hand side in (3.7) is bounded by

$$C_{\alpha\beta 1} \max_{|\eta| \leq |\alpha|} (1 + |\xi + \eta|^2)^{-|\alpha|/2}.$$

Peetre's inequality allows to conclude that

$$\begin{aligned} (1 + |\xi + \eta|^2)^{-|\alpha|/2} &\leq 2^{|\alpha|/2} (1 + |\eta|^2)^{|\alpha|/2} (1 + |\xi|^2)^{-|\alpha|/2}; \\ &\leq 2^{|\alpha|/2} (1 + |\alpha|^2)^{|\alpha|/2} (1 + |\xi|^2)^{-|\alpha|/2}, \end{aligned}$$

which is what we sought to establish. □

## 4 Numerical Algorithm

In Section 1, we list the four questions to be addressed and briefly outline the solutions. In this section, we describe the solutions in detail.

## 4.1 Computation of $P$ and $P^{-1}$

To construct  $P = L^{1/2}$  and  $P^{-1}$ , we use discrete symbol calculus. There is one small problem:  $L = -\nabla \cdot (c^2(x)\nabla)$  is only a positive semidefinite operator with one zero eigenvalue. Hence, the square root algorithm in Section 2.1 cannot be applied directly. However, since  $L$  is of divergence form, we know that the corresponding eigenfunction is the constant function. Therefore, we can fix this problem easily, and in the same manner as we did in the theory earlier, by removing the average from the initial condition  $u_0(x)$  and addressing it separately. For all initial value function  $u_0(x)$  with zero mean, we can redefine  $L$  to be

$$Lu = -\nabla \cdot (c^2(x)\nabla u) + \int u(x)dx.$$

Now this  $L$  is a positive definite operator, and we can use the algorithm in Section 2.1 to compute  $P = L^{1/2}$  and  $P^{-1} = L^{-1/2}$ .

## 4.2 Computation of $\Phi_{\pm}(x, \xi, \tau)$

The next task is to compute  $\Phi_{\pm}(x, \xi, t)$  for  $x \in X = \{(\frac{n_1}{N}, \frac{n_2}{N}) : n_1, n_2 \in \mathbb{Z}, 0 \leq n_1, n_2 < N\}$  and  $\xi \in \Omega = \{(\xi_1, \xi_2) : \xi_1, \xi_2 \in \mathbb{Z}, -\beta\frac{N}{2} \leq \xi_1, \xi_2 < \beta\frac{N}{2}\}$ .  $\Phi_{\pm}(x, \xi, t)$  are functions of homogeneous degree 1 in  $\xi$  and satisfy

$$\begin{cases} \partial_t \Phi_{\pm}(x, \xi, t) \mp c(x)|\nabla_x \Phi_{\pm}(x, \xi, t)| = 0 \\ \Phi_{\pm}(x, \xi, 0) = x \cdot \xi. \end{cases} \quad (4.1)$$

We can simplify the computation using two simple observations:

- Since  $\Phi_{\pm}(x, \xi, t)$  is homogeneous of degree 1 in  $\xi$ , we only need to solve the problem  $\Phi_{\pm}(x, \hat{\xi}, t)$  for  $\hat{\xi}$  on unit circle.
- Even though  $\Phi_{\pm}(x, \xi, t)$  is not periodic in  $x$ , its difference with the initial condition  $\Phi_{\pm}(x, \xi, t) - x \cdot \xi$  is periodic and is amenable to high accuracy spectral differentiation.

Therefore, we solve the following problem for  $\Psi_{\pm}(x, \xi, t) := \Phi_{\pm}(x, \xi, t) - x \cdot \xi$  instead: for each  $\hat{\xi} \in S^1$ ,

$$\begin{cases} \partial_t \Psi_{\pm}(x, \hat{\xi}, t) \mp c(x)|\nabla_x \Psi_{\pm}(x, \hat{\xi}, t) + \hat{\xi}| = 0 \\ \Psi_{\pm}(x, \hat{\xi}, 0) = 0. \end{cases} \quad (4.2)$$

Notice that the solution of this equation only depends on the coefficient  $c(x)$ . When  $c(x)$  is a bandlimited function,  $\Psi_{\pm}(x, \hat{\xi}, t)$  is a smooth function in  $x$  when  $t$  is significantly smaller than the time  $t^*$  when the caustics appear. This allows us to solve  $\Psi_{\pm}(x, \hat{\xi}, t)$  on a grid much coarser than the  $N \times N$  grid (where the function  $f(x)$  is defined) and still obtain a high accuracy. Moreover, it is easy to check that  $\Psi_{\pm}(x, \hat{\xi}, t)$  is also smooth in  $\hat{\xi}$ .

As a result, we discretize  $S^1$  with equispaced grid  $\hat{\xi}_k, 1 \leq k \leq M_{\xi}$ . For each  $\hat{\xi}_k$ , we solve  $\Psi_{\pm}(x, \hat{\xi}_k, t)$  on a  $M_x \times M_x$  Cartesian grid in the  $x$  variable. The  $x$  derivative  $\nabla_x \Psi_{\pm}(x, \hat{\xi}_k, t)$  is approximated using spectral differentiation, which can be easily done using the Fast Fourier Transform (FFT) since the function  $\Psi_{\pm}(x, \hat{\xi}_k, t)$  is periodic in  $x$ . We use a high order Runge-Kutta method for timestepping. Notice that since the solution grid is of size  $M_x \times M_x$ , the time step can be taken to be of order  $1/M_x$ .

Once the  $\Psi_{\pm}(x, \hat{\xi}_k, t)$  is obtained for each  $\hat{\xi}_k$  on the  $M_x \times M_x$  spatial grid, evaluating  $\Psi_{\pm}(x, \hat{\xi}, t)$  for any  $x$  and  $\hat{\xi}$  is a simple Fourier interpolation problem. When  $M_{\xi}$  is small, evaluating the Fourier series direction is sufficiently efficient. When  $M_{\xi}$  gets sufficiently larger, we can use tools such as nonuniform FFT [15].

### 4.3 Computation of $a_{\pm}(x, \xi, \tau)$

For  $\tau < t^*$ , the matrices  $(a_{\pm}(x, \xi, \tau))_{x \in X, \xi \in \Omega}$  are numerically low-rank. The algorithm described in Section 2.2 is used to construct their low-rank separated approximation. The main issue here is how to sample the rows and columns of  $(a_{\pm}(x, \xi, t))_{x \in X, \xi \in \Omega}$ .

Before we start, we first introduce some matrix notation that will simplify the presentation. Given two matrices  $A = (a_{x\xi})_{x \in X, \xi \in \Omega}$  and  $B = (b_{x\xi})_{x \in X, \xi \in \Omega}$  of the same size,  $A \odot B$  denotes their pointwise product, i.e., the matrix with entries  $(a_{x\xi}b_{x\xi})$ . We define the Fourier transform matrix  $F$  by

$$F = \left( \frac{1}{N^2} e^{-2\pi i x \cdot \xi} \right)_{\xi \in \Omega, x \in X}.$$

Accordingly the inverse Fourier transform matrix is given by

$$F^{-1} = (e^{2\pi i x \cdot \xi})_{x \in X, \xi \in \Omega}.$$

Recall that the main components of the solution formula are

$$(e^{\pm i P \tau} f)(x) = \sum_{\xi \in \Omega} a_{\pm}(x, \xi, \tau) e^{2\pi i \Phi_{\pm}(x, \xi, \tau)} \hat{f}(\xi) \quad (4.3)$$

for  $x \in X$ . If we define the matrices

$$A_{\pm} = (a_{\pm}(x, \xi, \tau))_{x, \xi}, \quad E_{\pm} = (e^{2\pi i \Phi_{\pm}(x, \xi, \tau)})_{x, \xi},$$

we can then rewrite (4.3) as

$$e^{\pm i P \tau} = (A_{\pm} \odot E_{\pm}) \cdot F.$$

As we now show, extracting rows and columns of  $A_{\pm}$  reduces to applying  $e^{\pm i P t}$  to some spectral functions. For any  $\tilde{\xi}$ , we define vector  $e_{\tilde{\xi}} = (e^{2\pi i x \cdot \tilde{\xi}})_x$ . Clearly  $e_{\tilde{\xi}}$  corresponds to a plane wave with wave number  $\tilde{\xi}$ . We use  $\delta_{\tilde{x}}$  to denote the vector with one at index  $\tilde{x}$  and zero everywhere else. Similarly,  $\delta_{\tilde{\xi}}$  is the vector with one at index  $\tilde{\xi}$  and zero everywhere else.

When we apply  $e^{\pm i P \tau}$  to  $e_{\tilde{\xi}}$ , we get

$$e^{\pm i P \tau} e_{\tilde{\xi}} = (A_{\pm} \odot E_{\pm}) F e_{\tilde{\xi}} = (A_{\pm} \odot E_{\pm}) \delta_{\tilde{\xi}}.$$

However, the last term is exactly equal to the  $\tilde{\xi}$ -th column of  $(A_{\pm} \odot E_{\pm})$ . Now, we can easily evaluate the  $\tilde{\xi}$ -th column of  $E_{\pm}$ , which is exactly equal to  $(e^{2\pi i \Phi_{\pm}(x, \tilde{\xi}, t)})_x$ . Dividing them term by term gives the  $\tilde{\xi}$ -th column of  $A_{\pm} = (a_{\pm}(x, \xi, \tau))_{x, \xi}$ . Now the only question that remains is how to compute  $e^{\pm i P \tau} e_{\tilde{\xi}}$ . It is easy to see that this is in fact the solution of the following problem at  $t = \tau$ :

$$\begin{cases} \partial_{tt} u(x, t) - \nabla \cdot (c^2(x) \nabla u(x, t)) = 0 & t > 0, x \in [0, 1]^2 \\ u(x, 0) = e_{\tilde{\xi}}(x) & x \in [0, 1]^2 \\ \partial_t u(x, 0) = (\pm i P e_{\tilde{\xi}})(x) & x \in [0, 1]^2, \end{cases} \quad (4.4)$$

which is solved with spectral differentiation in  $x$  and Runge-Kutta time stepping in  $t$ . Since for each  $\tilde{\xi}$  the solution takes  $O(N^3)$  steps, sampling each column of  $A_{\pm} = (a_{\pm}(x, \xi, \tau))_{x, \xi}$  costs at most  $O(N^3)$  steps.

In order to sample the  $\tilde{x}$ -th row of  $A_{\pm}$ , we use the fact

$$\delta_{\tilde{x}}^* e^{\pm iP\tau} F^{-1} = \delta_{\tilde{x}}^*(A_{\pm} \odot E_{\pm}).$$

Transposing the left side and using the fact  $(e^{\pm iP\tau})^* = e^{\mp iP\tau}$ , we get

$$(F^{-*} e^{\mp iP\tau} \delta_{\tilde{x}})^* = \delta_{\tilde{x}}^*(A_{\pm} \odot E_{\pm})$$

Therefore, in order to sample the  $\tilde{x}$ -th row of  $A_{\pm}$ , we compute  $e^{\mp iP\tau} \delta_{\tilde{x}}$ , apply  $(F^{-1})^* = N^2 F$  to it, transpose the result, and finally divide entry-wise the  $\tilde{x}$ -th row of  $E_{\pm}$ . Similar to the column sampling case,  $e^{-iP\tau} \delta_{\tilde{x}}$  is the solution of the following problem at time  $t = \tau$ :

$$\begin{cases} \partial_{tt}u(x, t) - \nabla \cdot (c^2(x)\nabla u(x, t)) = 0 & t > 0, x \in [0, 1]^2 \\ u(x, 0) = \delta_{\tilde{x}}(x) & x \in [0, 1]^2 \\ \partial_t u(x, 0) = (\mp iP\delta_{\tilde{x}})(x) & x \in [0, 1]^2 \end{cases} \quad (4.5)$$

Again, by using spectral differentiation in  $x$  and time stepping in  $t$ , we can sample each row of  $A_{\pm} = (a_{\pm}(x, \xi, \tau))_{x, \xi}$  in  $O(N^3)$  steps.

Since the algorithm in Section 2.2 for constructing the separated approximation of  $A_{\pm}$  requires sampling only a *constant* number of rows and columns (where the constant depends on the  $\varepsilon$ -rank of  $A_{\pm}$ ), the total cost of computing these separated approximations is still  $O(N^3)$ . We would to emphasize that the construction of these approximations is a precomputation step and is performed only once for a fixed  $c(x)$ . Once these approximations are ready, one can reuse them in the solutions for arbitrary initial conditions.

#### 4.4 Computation of $u(x, \tau)$

So far, we have discussed the details of constructing the Fourier integral representation

$$(e^{\pm iP\tau} f_{\pm})(x) = \sum_{\xi} e^{2\pi i \Phi_{\pm}(x, \xi, \tau)} a_{\pm}(x, \xi, \tau) \widehat{f_{\pm}}(\xi).$$

For bandlimited  $c(x)$ , the computing of  $P$  and  $P^{-1}$  takes  $O(\log N)$  steps. Computing  $\Phi_{\pm}(x, \xi, \tau)$  requires only  $O(1)$  steps. Factorizing  $(a_{\pm}(x, \xi, \tau))_{x, \xi}$  uses at most  $O(N^3)$  steps in total. Therefore, the total preprocessing time is of order  $O(N^3)$ .

Now let us briefly summarize the computing of  $u(x, \tau)$  given  $u_0(x)$  and  $u_1(x)$ .

1. Compute  $f_{\pm} = \frac{1}{2}(u_0 \pm (iP)^{-1}u_1)$ . The application of  $P^{-1}$  uses the algorithm in Section 2.1.
2. Compute

$$(e^{\pm iP\tau} f_{\pm})(x) = \sum_{\xi \in \Omega} a_{\pm}(x, \xi, \tau) e^{2\pi i \Phi_{\pm}(x, \xi, \tau)} \widehat{f_{\pm}}(\xi)$$

for  $x \in X$  using the Fourier integral operator algorithms described in Section 2.3. This takes  $O(N^2 \log N)$  or, depending on the algorithmic variant chosen,  $O(N^{2.5} \log N)$  steps.

3. Finally, set  $u(x, \tau) = (e^{iP\tau} f_{+})(x) + (e^{-iP\tau} f_{-})(x)$  for  $x \in X$ .



## 5 Numerical Results

In this section, we present several numerical examples to illustrate the algorithm described in Section 4. We implement the algorithms in Matlab and all numerical results are obtained on a workstation with a 3.0GHz CPU. We set an overall error threshold of order  $10^{-4}$  for all components of the algorithm. The time  $\tau$  for which the FIO representation is constructed is taken to be  $1/8$ , which is less than  $t^*$  in all examples.

**Example 1.** The coefficient  $c(x)$  is given by a two-dimensional sine wave (see Figure 4).

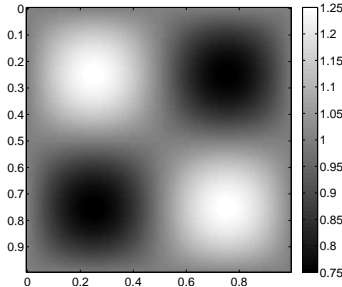


Figure 4: Example 1. Coefficient  $c(x)$ .

As we mentioned in Section 4, the precomputation step includes four steps: (1) the construction of  $P$  and  $P^{-1}$  using discrete symbol calculus, (2) the computation of the phase functions  $\Psi_{\pm}(x, \xi, t)$ , (3) the computation of the amplitudes  $a_{\pm}(x, \xi, t)$ , and (4) finally the precomputation of the Fourier integral operators with symbol  $a_{\pm}(x, \xi, t)e^{2\pi i\Phi_{\pm}(x, \xi, t)}$ .

The representation of the symbols of  $P = L^{1/2}$  and  $P^{-1} = L^{-1/2}$  uses  $9 \times 9$  Fourier modes  $e_{\lambda}(x)$  in the  $x$  domain (i.e.,  $B = 5$ ). In the  $\xi$  domain, the symbols are representation with the hierarchical spline approximation, with a total number of 667 samples for  $N \leq 1024$ . The construction of  $P$  and  $P^{-1}$  through the Schulz-Higham iteration converges in 35 iterations and costs  $1.37e+02$  seconds in total.

The phase functions  $\Psi_{\pm}(x, \hat{\xi}, \tau)$  are computed with a uniform  $32 \times 32$  Cartesian grid in the  $x$  domain and a uniform 128 grid in the angular component of  $\xi$ . We solve the Hamilton-Jacobi equation in time using the standard 4th-order Runge-Kutta method, with a time step equal to  $1/512$ . The resulting phase function  $\Psi_{\pm}(x, \hat{\xi}, \tau)$  has an accuracy of order  $10^{-8}$ , which is sufficiently accurate for the overall computation. The evaluation of  $\Psi_{\pm}(x, \hat{\xi}, \tau)$  at locations off-grid is done with the standard Fourier interpolation, so that the high order accuracy is preserved. The computation of  $\Psi_{\pm}(x, \hat{\xi}, \tau)$  takes only  $1.65e+01$  seconds.

Most of the precomputation time is spent on the the construction of the amplitudes  $a_{\pm}(x, \xi, \tau)$  and the precomputation of the FIOs with symbol  $a_{\pm}(x, \xi, \tau)e^{2\pi i\Phi_{\pm}(x, \xi, \tau)}$ . These two components depend on the problem size  $N$ . In the following experiments,  $N = 128, 256, 512, \text{ and } 1024$ . For the precomputation of the FIOs, we use the version based on angular decomposition (Approach 1 of Section 2.3). Even though this version has a relatively high asymptotic complexity, its relatively smaller prefactor constant makes it more efficient for the problem sizes addressed here (i.e.,  $N = 128, \dots, 1024$ ). The results of the precomputation are summarized in Table 1.  $T_{\text{AMP}}$  is the time for constructing the separated factorizations of  $(a_{\pm}(x, \xi, \tau))_{x, \xi}$  in seconds,  $r_{\text{AMP}}$  is the  $\varepsilon$ -rank of resulting factorization with  $\varepsilon = 10^{-4}$ ,  $T_{\text{FIO}}$  is the precomputation time of the FIOs in seconds,  $W_{\text{FIO}}$  is the total

number of wedges in the angular partitioning of the frequency domain, and  $r_{\text{FIO}}$  is the (average) separation rank in (2.1).

$N$	$T_{\text{AMP}}$	$r_{\text{AMP}}$	$T_{\text{FIO}}$	$W_{\text{FIO}}$	$r_{\text{FIO}}$
128	5.29e+02	19	1.32e+02	16	9
256	3.34e+03	18	3.27e+02	24	8
512	2.55e+04	17	1.31e+03	32	7
1024	1.38e+05	17	5.58e+03	46	7

Table 1: Example 1. Results of the precomputation of the amplitudes  $a_{\pm}(x, \xi, \tau)$  and the FIOs.  $T_{\text{AMP}}$  is the time for constructing the factorizations of  $(a_{\pm}(x, \xi, \tau))_{x, \xi}$  in seconds,  $r_{\text{AMP}}$  is the  $\varepsilon$ -rank of resulting factorization with  $\varepsilon = 10^{-4}$ ,  $T_{\text{FIO}}$  is the precomputation time of the FIOs in seconds,  $W_{\text{FIO}}$  is the total number of wedges in the angular partitioning of the frequency domain, and  $r_{\text{FIO}}$  is the (average) separation rank in (2.1).

For each  $N$ , we apply the resulting FIO representation to compute the wave solution  $u(x, \tau)$  at  $\tau = 1/8$  for the following three initial conditions.

1. Harmonic wave

$$u_0(x) = e^{2\pi i(\alpha_1 N x_1 + \alpha_2 N x_2)}, \quad u_1(x) = -2\pi i \sqrt{\alpha_1^2 + \alpha_2^2} N e^{2\pi i(\alpha_1 N x_1 + \alpha_2 N x_2)},$$

with  $(\alpha_1, \alpha_2) = (5/32, 3/32)$ . As  $N$  grows, the initial condition becomes more and more oscillatory.

2. Plane wave

$$u_0(x) = \exp(-(N/4)^2(x_1 - 1/2)^2), \quad u_1(x) = 0.$$

This initial condition is concentrated along the line  $x_1 = 1/2$ , and it becomes more and more singular as  $N$  grows.

3. Gaussian bump

$$u_0(x) = \exp(-(N/4)^2((x_1 - 1/2)^2 + (x_2 - 1/2)^2)), \quad u_1(x) = 0.$$

This Gaussian bump is localized near  $(x_1, x_2) = (1/2, 1/2)$ .

Table 2 summarizes the running time and relative  $L^2$  error of computing  $u(x, \tau)$  using the constructed FIO representations for these initial conditions and for different values of  $N$ . Here the relative error is estimated by comparing our result with the solution computed by an accurate time-stepping scheme with spectral differentiation in space and the 4th-order Runge-Kutta method in time (with sufficiently small time step). The error of this time-stepping scheme is of order  $10^{-7}$  for  $t \in [0, \tau]$ , so that we can effectively treat its result as the exact solution.

From Table 2, we observe that the error is consistently of order  $10^{-4}$ , which shows that our algorithm has the desired accuracy. Each time  $N$  doubles, the number of unknowns grows by a factor of four. For the first two initial conditions, the running time of the evaluation algorithm seems to scale linearly with respect to the unknowns. The reason behind this is that the first two initial conditions are both well-localized in the frequency domain and therefore, for the angular decomposition based FIO algorithm, one only needs to visit a small number of wedges. On the other hand, the last initial condition has a support

covering the whole frequency domain, and hence the FIO algorithm visits all wedges. One clearly sees that in this case the running time of our evaluation algorithm grows by a factor of 5 to 6 when the number of unknowns quadruples, which is consistent with the theoretical  $O(N^{2.5} \log N)$  estimate.

$N$	Init. cond. 1		Init. cond. 2		Init. cond. 3	
	Time	Error	Time	Error	Time	Error
128	7.90e-01	9.24e-05	4.65e-01	9.86e-05	3.57e+00	9.13e-05
256	1.77e+00	1.56e-04	1.76e+00	1.72e-04	2.07e+01	1.30e-04
512	6.85e+00	1.10e-04	6.67e+00	2.04e-04	1.03e+02	1.25e-04
1024	2.73e+01	1.82e-04	2.70e+01	2.28e-04	6.21e+02	1.65e-04

Table 2: Example 1. Running time and relative  $L^2$  error of computing  $u(x, \tau)$  for different initial conditions and different values of  $N$ .

We plot in Figure 5 the running time of the FIO evaluation algorithm in comparison with the full time-stepping algorithm with spectral differentiation in space, for the three initial conditions. It is clear that the algorithm based on the FIO representation is much more efficient than the time-stepping algorithm for the first two initial conditions again due to the frequency localization of the initial data. For the last initial condition, even though the FIO based algorithm has a relatively higher absolute running time, the curve suggests that asymptotically it is eventually more efficient than the full time-stepping algorithm. (Note that finite differences are at least as costly as a spectral method at the desired level of accuracy.)

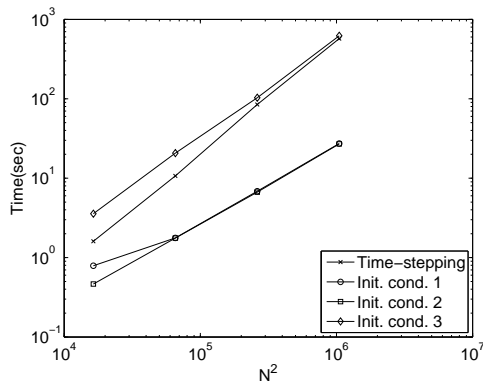


Figure 5: Comparison of the running time of our evaluation algorithm with the running time of the time-stepping algorithm.

Figure 6 shows the solutions for three initial conditions at time  $\tau$ ,  $2\tau$ ,  $3\tau$ , and  $4\tau$  with  $\tau = 1/8$  for  $N = 512$ . It is clear from these plots that, although we take a time step much larger than the CFL limit, the numerical dispersion effect is not an issue due to the spectral nature of our approach.

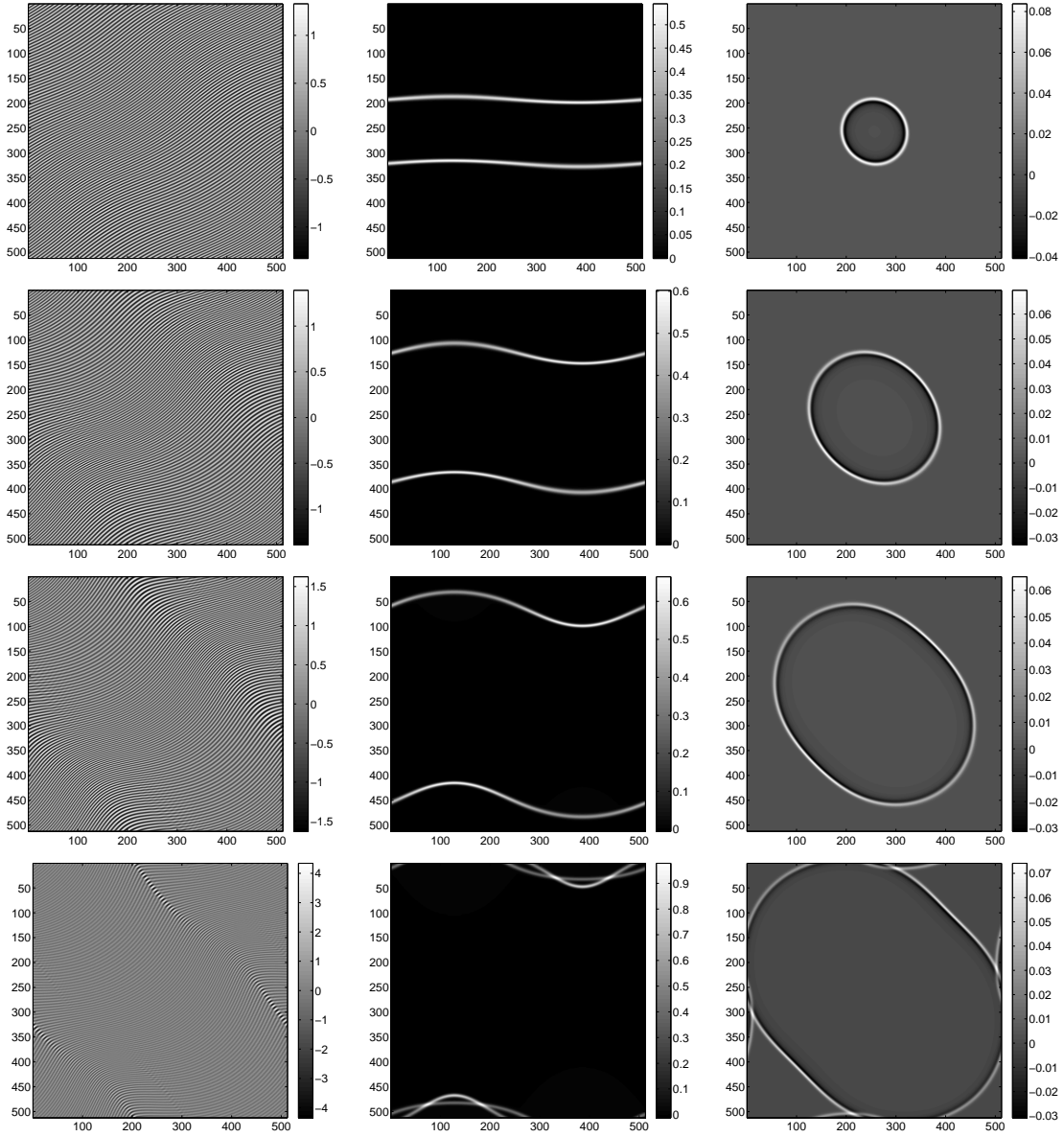


Figure 6: Example 1. The solution  $u(x,t)$  at  $t = 1/8, 1/4, 3/8,$  and  $1/2$  (from top to bottom) for three different initial conditions with  $N = 512$ .

**Example 2.** We consider a wave guide problem with coefficient  $c(x)$  given in Figure 7.

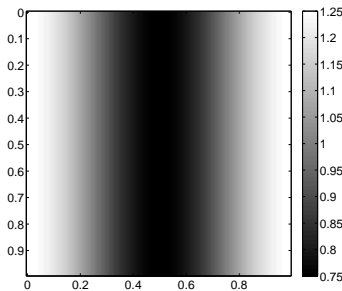


Figure 7: Example 2. Coefficient  $c(x)$ .

The results for constructing  $P$  and  $\Psi_{\pm}(x, \hat{\xi}, t)$  are omitted since they are similar to the first example. We first report the results of the construction of amplitudes  $a_{\pm}(x, \xi, \tau)$  and the precomputation of FIOs  $a_{\pm}(x, \xi, \tau)e^{2\pi i\Phi_{\pm}(x, \xi, \tau)}$  for  $N = 128, 256, 512$ , and  $1024$  in Table 3. Since the coefficient  $c(x)$  only depends on  $x_2$ , the amplitude approximations and the FIO representations are simpler compared to the first example. The separation ranks  $r_{\text{AMP}}$  and  $r_{\text{FIO}}$  are significantly smaller.

$N$	$T_{\text{AMP}}$	$r_{\text{AMP}}$	$T_{\text{FIO}}$	$W_{\text{FIO}}$	$r_{\text{FIO}}$
128	4.22e+02	7	1.17e+02	16	7
256	2.72e+03	7	3.07e+02	24	6
512	2.16e+04	7	1.28e+03	32	6
1024	1.12e+05	7	5.42e+03	46	5

Table 3: Example 2. Results of the precomputation of the amplitudes  $a_{\pm}(x, \xi, \tau)$  and the FIOs.

For each value of  $N$ , we apply the constructed FIOs to three initial conditions considered in the first example. The results for different initial conditions are summarized in Table 4. As a direct result of the smaller separation rank  $r_{\text{FIO}}$ , the running times are much lower compared to the ones in the first example. However, the scaling behavior is very similar.

$N$	Init. cond. 1		Init. cond. 2		Init. cond. 3	
	Time	Error	Time	Error	Time	Error
128	4.45e-01	3.70e-05	3.53e-01	2.32e-05	2.82e+00	4.33e-05
256	1.51e+00	4.64e-05	1.39e+00	4.03e-05	1.73e+01	3.45e-05
512	5.87e+00	4.21e-05	5.51e+00	7.26e-05	8.89e+01	5.52e-05
1024	2.37e+01	5.37e-05	2.31e+01	7.71e-05	5.21e+02	6.45e-05

Table 4: Example 2. Running time and relative  $L^2$  error of  $u(x, \tau)$  for different initial conditions and different values of  $N$ .

Figure 8 shows the solutions for the three initial conditions at time  $\tau, 2\tau, 3\tau$ , and  $4\tau$  with  $\tau = 1/8$  for  $N = 512$ .

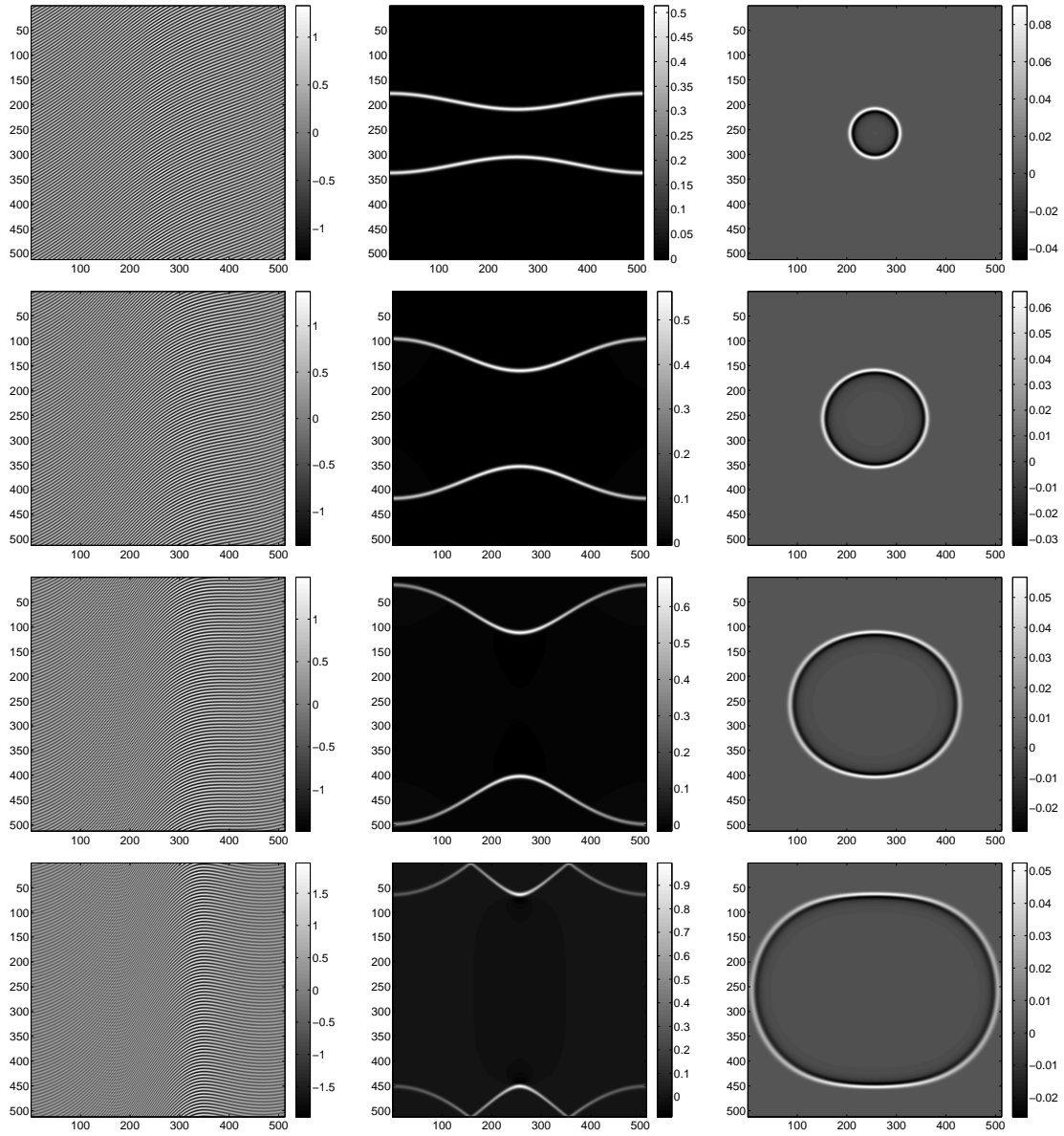


Figure 8: Example 2. The solution  $u(x,t)$  at  $t = 1/8, 1/4, 3/8,$  and  $1/2$  (from top to bottom) for three different initial conditions with  $N = 512$ .

**Example 3.** Finally, we consider a randomly generated  $c(x)$  given in Figure 9.

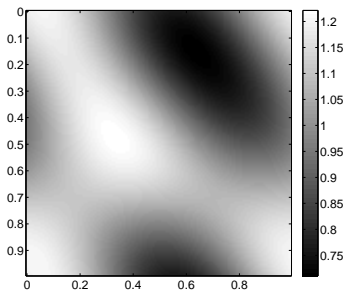


Figure 9: Example 3. Coefficient  $c(x)$ .

The results of the precomputation of the amplitudes and the FIOs for different values of  $N$  are reported in Table 5.

$N$	$T_{\text{AMP}}$	$r_{\text{AMP}}$	$T_{\text{FIO}}$	$W_{\text{FIO}}$	$r_{\text{FIO}}$
128	5.69e+02	20	1.31e+02	16	10
256	3.52e+03	19	3.41e+02	24	8
512	2.54e+04	18	1.33e+03	32	8
1024	1.42e+05	16	5.66e+03	46	7

Table 5: Example 3. Results of the precomputation of the amplitudes  $a_{\pm}(x, \xi, \tau)$  and the FIOs.

For each value of  $N$ , the FIOs is again applied to three initial conditions and these results are summarized in Table 6.

$N$	Init. cond. 1		Init. cond. 2		Init. cond. 3	
	Time	Error	Time	Error	Time	Error
128	7.43e-01	1.68e-04	4.89e-01	1.38e-04	3.65e+00	1.23e-04
256	1.85e+00	1.44e-04	1.79e+00	1.72e-04	2.13e+01	1.37e-04
512	7.08e+00	2.88e-04	6.93e+00	2.01e-04	1.07e+02	1.73e-04
1024	2.87e+01	2.39e-04	2.83e+01	2.49e-04	6.27e+02	1.74e-04

Table 6: Example 3. Running time and relative  $L^2$  error of  $u(x, \tau)$  for different initial conditions and different values of  $N$ .

Figure 10 shows the solutions for three initial conditions at time  $\tau$ ,  $2\tau$ ,  $3\tau$ , and  $4\tau$  with  $\tau = 1/8$  for  $N = 512$ .

## References

- [1] G. Bao and W. W. Symes. Computation of pseudo-differential operators. *SIAM J. Sci. Comput.*, 17(2):416–429, 1996.
- [2] S. Basu and Y. Bresler.  $O(N^2 \log N)$  filtered backprojection algorithm for tomography. *IEEE Trans. Im. Proc.*, 9(10):1760–1773, 2000.

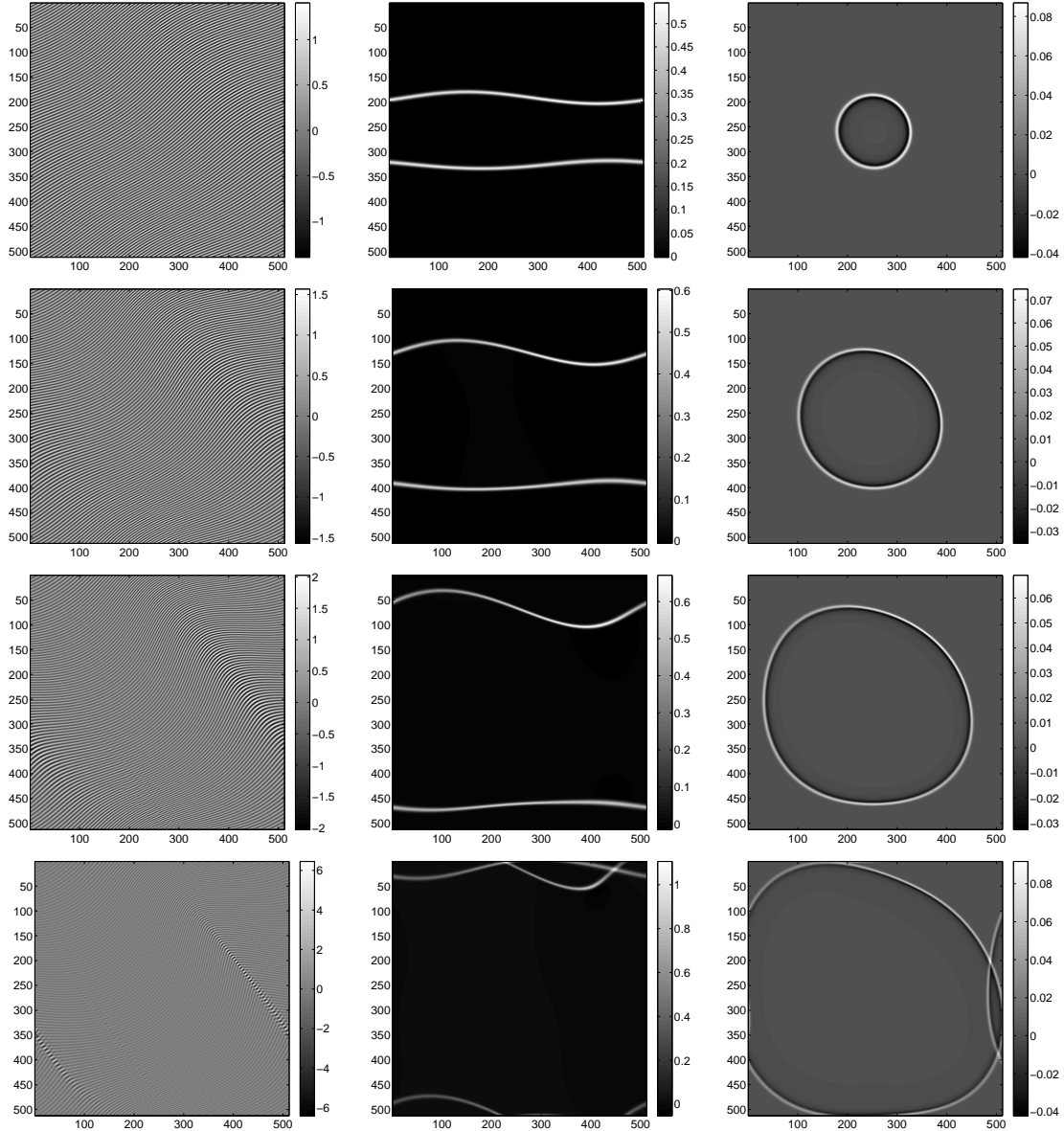


Figure 10: Example 3. The solution  $u(x,t)$  at  $t = 1/8, 1/4, 3/8,$  and  $1/2$  (from top to bottom) for three different initial conditions with  $N = 512$ .

- [3] G. Beylkin and K. Sandberg. Wave propagation using bases for bandlimited functions. *Wave Motion*, 41(3):263–291, 2005.
- [4] O. Bruno and M. Lyon. High-order unconditionally stable FC-AD solvers for general smooth domains I. basic elements. *J. Comput. Phys.*, 229(9):3358–3381, 2010.
- [5] E. Candès and L. Demanet. Curvelets and Fourier integral operators. *C. R. Math. Acad. Sci. Paris*, 336(5):395–398, 2003.
- [6] E. Candès and L. Demanet. The curvelet representation of wave propagators is optimally sparse. *Comm. Pure Appl. Math.*, 58(11):1472–1528, 2005.
- [7] E. Candès, L. Demanet, and L. Ying. Fast computation of Fourier integral operators. *SIAM Journal on Scientific Computing*, 29(6):2464–2493, 2007.



- [8] E. Candès, L. Demanet, and L. Ying. A fast butterfly algorithm for the computation of fourier integral operators. *SIAM Multiscale Model. Simul.*, 7(4):1727–1750, 2009.
- [9] W. Chew, E. Michielssen, J. M. Song, and J. M. Jin, editors. *Fast and efficient algorithms in computational electromagnetics*. Artech House, Inc., Norwood, MA, USA, 2001.
- [10] A. Córdoba and C. Fefferman. Wave packets and Fourier integral operators. *Comm. PDE*, 3(11):979–1005, 1978.
- [11] M. V. de Hoop, J. le Rousseau, and R. Wu. Generalization of the phase-screen approximation for the scattering of acoustic waves. *Wave Motion*, 31(1):43–70, 2000.
- [12] L. Demanet and L. Ying. Wave atoms and time upscaling of wave equations. *Numer. Math.*, 113(1):1–71, 2009.
- [13] L. Demanet and L. Ying. Discrete symbol calculus. *SIAM Review*, 53:71–104, 2011.
- [14] J. Duistermaat. *Fourier integral operators*. Birkhauser, Boston, 1996.
- [15] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comput.*, 14(6):1368–1393, 1993.
- [16] B. Engquist, S. Osher, and S. Zhong. Fast wavelet based algorithms for linear evolution equations. *SIAM J. Sci. Comput.*, 15(4):755–775, 1994.
- [17] N. J. Higham. Stable iterations for the matrix square root. *Numer. Algorithms*, 15(2):227–242, 1997.
- [18] L. Hörmander. *The Analysis of Linear Partial Differential Operators, 4 volumes*. Springer, 1985.
- [19] L. Ji, J. R. McLaughlin, D. Renzi, and J.-R. Yoon. Interior elastodynamics inverse problems: shear wave speed reconstruction in transient elastography. *Inverse Problems*, 19:S1–S29, 2003.
- [20] P. Lax. Asymptotic solutions of oscillatory initial value problems. *Duke Math J.*, 24:627–646, 1957.
- [21] E. Michielssen and A. Boag. A multilevel matrix decomposition algorithm for analyzing scattering from large structures. *IEEE Transactions on Antennas and Propagation*, 44(8):1086–1093, 1996.
- [22] S. Nilsson and L. Andersson. Application of fast backprojection techniques for some inverse problems of synthetic aperture radar. *Proc. SPIE*, 3370:62–72, 1998.
- [23] M. O’Neil, F. Woolfe, and V. Rokhlin. An algorithm for the rapid evaluation of special function transforms. *Appl. Comput. Harmon. Anal.*, 28(2):203–226, 2010.
- [24] M. Ruzhansky and V. Turunen. On the Fourier analysis of operators on the torus. *Operator Theory: Advances and Applications*, 172:87–105, 2007.
- [25] A. Seeger, C. Sogge, and E. Stein. Regularity properties of Fourier integral operators. *Annals of Math.*, 134:231–251, 1991.
- [26] R. T. Seeley. Complex powers of an elliptic operator. *Proc. Symp. Pure Math*, 10:288–307, 1967.
- [27] H. Smith. A parametrix construction for wave equations with  $C^{1,1}$  coefficients. *Ann. Inst. Fourier (Grenoble)*, 48:797–835, 1998.
- [28] C. Sogge. *Fourier Integrals in Classical Analysis*. Cambridge University Press, 1993.
- [29] E. M. Stein. *Harmonic analysis: real-variable methods, orthogonality, and oscillatory integrals*, volume 43 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1993. With the assistance of Timothy S. Murphy, Monographs in Harmonic Analysis, III.

- [30] C. C. Stolk. A fast method for linear waves based on geometrical optics. *SIAM J. Num. Anal.*, 47(2):1168–1194, 2009.
- [31] W. W. Symes. Velocity inversion: a case study in infinite-dimensional optimization. *Mathematical Programming: Series A and B*, 48(1):71–102, 1990.
- [32] W. W. Symes. Mathematical foundations of reflection seismology. Technical report, Rice University, 1998.
- [33] F. Trèves. *Introduction to pseudodifferential and Fourier integral operators*. Plenum Press, New York, 1980.
- [34] V. Turunen. Commutator characterization of periodic pseudodifferential operators. *Z. Anal. Anw.*, 19:95–108, 2000.
- [35] M. Tygert. Fast algorithms for spherical harmonic expansions, III. Technical report, New York University, 2009.
- [36] L. Ulander, H. Hellsten, and G. Stenstrom. Synthetic-aperture radar processing using fast factorized back-projection. *IEEE Trans. Aero. Elect. Systems*, 39(3):760–776, 2003.
- [37] L. Ying. Sparse Fourier transform via butterfly algorithm. *SIAM J. Sci. Comput.*, 31(3):1678–1694, 2009.
- [38] L. Ying and S. Fomel. Fast computation of partial Fourier transforms. *SIAM Multiscale Model. Simul.*, 8:110–124, 2009.