Journal of Theoretical and Computational Acoustics
2050021 (26 pages)

DOI: 10.1142/S2591728520500218

Pseudospectral Time-Domain (PSTD) Methods for the Wave Equation: Realizing Boundary Conditions with Discrete Sine and Cosine Transforms

Elliott S. Wise
Department of Medical Physics and Biomedical Engineering
University College London, UK

Jiri Jaros
Centre of Excellence IT4Innovations
Faculty of Information Technology
Brno University of Technology, Czech Republic

Ben T. Cox and Bradley E. Treeby*
Department of Medical Physics and Biomedical Engineering
University College London, UK
**b.treeby@ucl.ac.uk*

Received 1 May 2020
Accepted 14 September 2020
Published

Pseudospectral time domain (PSTD) methods are widely used in many branches of acoustics for the numerical solution of the wave equation, including biomedical ultrasound and seismology. The use of the Fourier collocation spectral method in particular has many computational advantages. However, the use of a discrete Fourier basis is also inherently restricted to solving problems with periodic boundary conditions. Here, a family of spectral collocation methods based on the use of a sine or cosine basis is described. These retain the computational advantages of the Fourier collocation method but instead allow homogeneous Dirichlet (sound-soft) and Neumann (sound-hard) boundary conditions to be imposed. The basis function weights are computed numerically using the discrete sine and cosine transforms, which can be implemented using $O(N \log N)$ operations analogous to the fast Fourier transform. Practical details of how to implement spectral methods using discrete sine and cosine transforms are provided. The technique is then illustrated through the solution of the wave equation in a rectangular domain subject to different combinations of boundary conditions. The extension to boundaries with arbitrary real reflection coefficients or boundaries that are nonreflecting is also demonstrated using the weighted summation of the solutions with Dirichlet and Neumann boundary conditions.

Keywords: Boundary conditions; discrete cosine transform; discrete sine transform; pseudospectral.

1. Introduction

Simulating the propagation of acoustic waves requires the spatial and temporal derivative operators in the wave equation to be replaced with discrete formulas that can be implemented using a computer. A widely-used approach is the pseudospectral time domain (PSTD) method.[1,2] This uses the Fourier collocation spectral method to calculate spatial gradients, and a finite-difference method to integrate forwards in time.[3] For smooth wave fields, the error in the Fourier collocation spectral method decays exponentially with the number of grid nodes.[4,5] This means only a small number of grid points per wavelength are required to reduce the effects of numerical dispersion, often close to the Nyquist limit of two points per wavelength (PPW).[6,7] In comparison, the finite-difference time-domain (FDTD) method typically requires 10 to 20 PPW to minimize dispersion, and this number increases with the domain size due to the accumulation of numerical errors.[8] The difference in the number of PPW is critical for the tractability of many large-scale problems in which waves propagate across domains that are hundreds of wavelengths in each spatial dimension.[9,10]

While the PSTD method has many advantages, the use of a discrete Fourier basis to compute spatial gradients is also inherently restricted to solving problems with periodic boundary conditions. This is because the discrete Fourier transform of a finite sequence assumes this sequence is periodically repeated outside its finite support. Practically, this means when the PSTD method is used to solve the wave equation, waves exiting the computational domain on one side will reappear on the other side. This can be understood in terms of image sources, where the pressure field is periodically (and infinitely) repeated as shown in Fig. 1. In this example, as the wave within the computational domain exits to the right, the wavefield from the periodic image source appears within the domain on the left. In one-dimension (1D) where there is no decay of the wavefield with distance, this wave wrapping will continue *ad infinitum*.

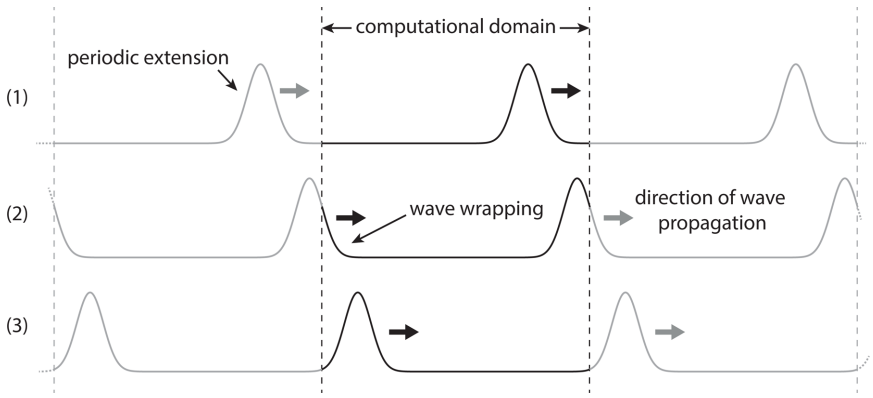

Fig. 1. Wave wrapping due to the periodicity assumed by the discrete Fourier transform. Three snapshots in time are shown. As the wave exits the computational domain on the right, the wave from the periodically repeated image source appears on the left.

In most circumstances in acoustics, a periodic boundary condition is neither physical nor desirable. To model nonreflecting boundaries (where the outgoing waves propagate as though in free-space), the periodicity is generally counteracted through the application of a perfectly matched layer (PML).[3,11] This is designed to absorb outgoing waves at the edge of the computational domain. A large body of literature now exists on the optimization and implementation of PMLs and they can be highly effective.[12] However, in some cases, other boundary conditions are required, and these are not straightforward to directly implement when using PSTD methods.[13,14]

To model reflecting boundaries, image sources can be used (assuming the wave propagation is linear). In this case, the true source and an image source are placed on opposites sides of the intended boundary (this idea was introduced by Lord Kelvin in the study of electrostatics[15]). In the simplest case, the image source is placed equidistant from the intended boundary and is either positive (in-phase) or negative (out-of-phase) relative to the true source, but otherwise identical. As the waves propagate, they pass each other at the (virtual) boundary and the summation of the two wave fields gives rise to the correct boundary and scattering behavior.

An illustrative example of using image sources to impose boundary conditions in 1D is given in Fig. 2. A positive image source corresponds to a sound-hard or Neumann boundary condition, where the gradient of the pressure field is always zero on the boundary, and the reflected wave is in phase with the incident wave. Considering the propagation of a wave between two acoustic half-spaces, this is equivalent to a boundary where the second medium has a much higher acoustic impedance. Similarly, a negative image source corresponds to a sound-soft or Dirichlet boundary condition, where the pressure is always zero on the boundary (this is sometimes called a pressure-release boundary[16]), and the reflected wave is out-of-phase with the incident wave. Considering two acoustic half-spaces, this is equivalent to a boundary where the second medium has a much lower acoustic impedance.

Using the idea of image sources, boundary conditions can be imposed within a PSTD wave model by increasing the size of the computational domain and directly including the image source terms. For example, this approach has been used to model wave propagation within reverberant cavities with reflecting walls.[17] However, the significant drawback is that the domain size must be increased in each Cartesian direction to allow the image sources to be defined. This can increase the computational load considerably, particularly in 3D. An alternative approach is to use another basis to compute the spatial gradients that already imposes the required boundary conditions. In this paper, the Fourier basis normally used in the PSTD method is replaced with a sine or cosine basis. These implicitly enforce homogeneous Dirichlet or Neumann boundary conditions while still retaining the same computational benefits as a Fourier collocation method.[18]

Using sine and cosine transforms to compute spectral gradients as part of the PSTD method has previously been discussed by several authors. In acoustics, Kosloff and Kosloff used a cosine basis to solve a 1D wave equation with a Neumann (or sound-hard) boundary condition at each end of the domain.[19] Sine and cosine transforms have also been used

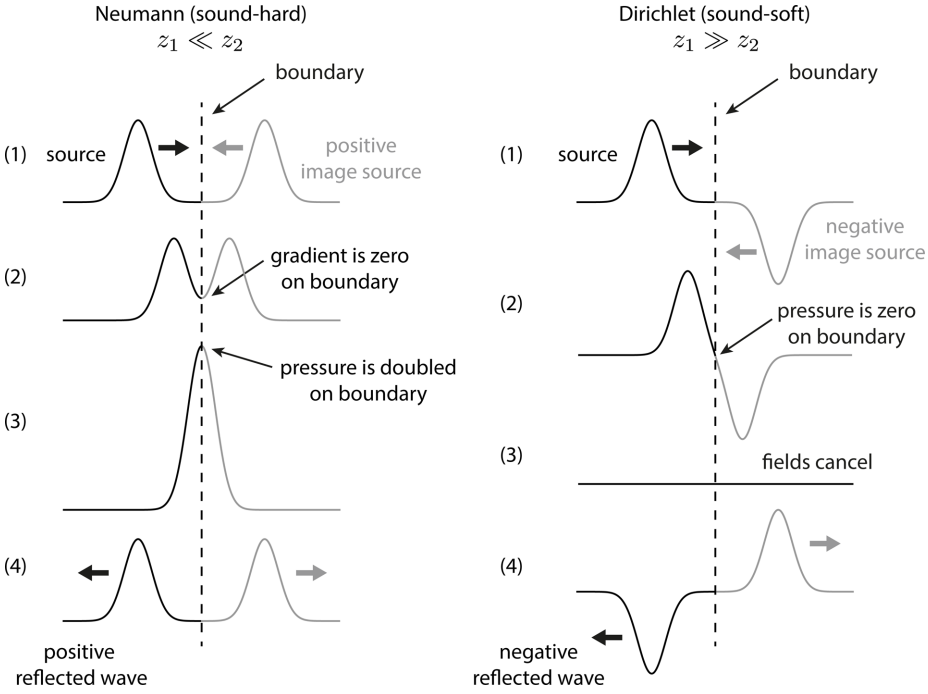

Fig. 2. Neumann (sound-hard) and Dirichlet (sound-soft) boundary conditions realized using image sources. The Neumann case uses a positive (or in-phase) image source, which results in a pressure doubling and zero pressure gradient on the boundary. The Dirichlet case uses a negative (or out-of-phase) image source, which results in the pressure on the boundary always equal to zero. Acoustically, the Neumann case corresponds to the propagation of a wave from a low to high impedance medium (i.e. $z_1 \ll z_2$), where the reflected wave is in phase with the incident wave. The Dirichlet case corresponds to the propagation of a wave from a high to a low impedance medium (i.e. $z_1 \gg z_2$), where the reflected wave is out of phase with the incident wave. The numbers (1) to (4) correspond to four snapshots in time showing the interaction of the incident wave with the boundary.

in elastodynamics for modeling applied forces that are symmetric or antisymmetric in the plane,[20] and in electromagnetics to ensure field components vanish at the boundary, e.g. when modeling wave guides.[21,22] Outside of wave problems, sine and cosine transforms have also been used to impose Dirichlet and Neumann boundary conditions for reaction–diffusion problems,[23] to impose slip boundary conditions within fluid simulations,[24] to impose symmetric boundary conditions when solving the Elder problem in hydrology,[25] to impose Dirichlet boundary conditions when solving the Klein–Gordon–Zakharov (KGZ) system,[26] and to calculate spectral gradients through symmetric extensions.[27]

In the continuous case, the use of a cosine or sine basis to enforce Dirichlet or Neumann boundary conditions is straightforward to conceptualize—a sine wave is zero at the origin and thus satisfies a Dirichlet condition, while the gradient of a cosine wave is zero at the origin and thus satisfies a Neumann condition.[18] However, in the discrete case, the different combinations of discrete symmetry leads to some complexity, with 16 different discrete trigonometric transforms (DTTs)[28,29] that each correspond to different boundary conditions

and spatial grid definitions. To form a spectral collocation method, it is also important to understand how the discrete symmetry changes under differentiation and with the use of staggered computational grids.

In this paper, we review how the 16 discrete trigonometric transforms fall into four distinct transform classes (see e.g. Ref. 29), and then discuss how each of these can be used to spectrally compute derivatives assuming different combinations of discrete symmetry. In Sec. 2, the conventional Fourier PSTD method is reviewed for the wave equation. In Sec. 3, the discrete cosine and sine transforms are reviewed, and the differentiation and interpolation properties of these transforms are discussed. An example of a DTT-based PSTD method is then given in Sec. 4. In Sec. 5, an extension to nonreflecting boundaries and arbitrary reflection coefficients is given. Discussion and summary are provided in Sec. 6.

2. The Fourier PSTD Method for the Wave Equation

Before discussing the machinery of using sine and cosine transforms to impose Dirichlet and Neumann boundary conditions, it is constructive to revisit the conventional Fourier PSTD method applied to the solution of the wave equation. The background to much of what follows can be found in several numerical methods texts, e.g. Refs. 30, 2, 4 and 5. Consider the following coupled system of equations which describes linear acoustic wave propagation in a homogeneous medium in one-dimension

$$\begin{aligned} \frac{\partial u}{\partial t} &= -\frac{1}{\rho_0}\frac{\partial p}{\partial x}, \\ \frac{\partial p}{\partial t} &= -\rho_0 c_0^2 \frac{\partial u}{\partial x}. \end{aligned} \tag{1}$$

Here u is the acoustic particle velocity, p is the acoustic pressure, ρ_0 is the mass density, c_0 is the sound speed, and x and t represent the space and time variables, respectively. This system of coupled first-order equations is commonly used in computational acoustics instead of the second-order wave equation as it allows velocity inputs and outputs to be used, and a split-field perfectly matched layer to be applied.[31]

Now consider the numerical solution of Eq. (1) using the Fourier PSTD method where u is calculated on a temporally and spatially staggered grid relative to p, where the staggered grid is shifted by half the temporal and spatial grid spacing (the use of a staggered grid improves the accuracy and stability of PSTD methods[32,33]). Three ingredients are required: (1) discretization of the spatial gradients using the Fourier collocation spectral method, (2) transformation of the gradient values to and from the spatially staggered grid, and (3) discretization of the temporal gradients using a finite difference scheme.

Starting with the spatial gradient calculation, in the Fourier collocation spectral method, the pressure and velocity fields are decomposed into a sum of complex exponential basis functions using a Fourier transform. In the continuous case, the forward and inverse Fourier transforms over one spatial dimension can be written as (other conventions can equally be

used)

$$\begin{aligned}\mathcal{F}\{f(x)\} \equiv g(k) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx}\, dx, \\ \mathcal{F}^{-1}\{g(k)\} \equiv f(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(k) e^{ikx} dk,\end{aligned} \tag{2}$$

where $i = \sqrt{-1}$, k is the spatial frequency or wavenumber, and $g(k)$ represents the complex basis function weights, i.e. the amplitude and phase of each frequency component needed to make up the original function $f(x)$. Using these transformations, it is straightforward to write an expression for the Fourier derivative of a bounded function, where[2]

$$\begin{aligned}\frac{d}{dx} f(x) &= \frac{d}{dx} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(k) e^{ikx}\, dk \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (ik) g(k) e^{ikx}\, dk \\ &= \mathcal{F}^{-1}\left\{ik \mathcal{F}\left\{f(x)\right\}\right\}.\end{aligned} \tag{3}$$

In the discrete case, $f(x)$ is given by a finite periodic sequence, and the Fourier transforms are calculated using the discrete Fourier transform, generally using the fast Fourier transform (FFT). The discrete wavenumber vector k is given by

$$k_n = \frac{2\pi}{N\Delta x} n \quad \text{where } n = \begin{cases} -\dfrac{N-1}{2}, -\dfrac{N-1}{2} + 1, \ldots, \dfrac{N-1}{2} & \text{if } N \text{ is odd} \\ -\dfrac{N}{2}, -\dfrac{N}{2} + 1, \ldots, \dfrac{N}{2} - 1 & \text{if } N \text{ is even.} \end{cases} \tag{4}$$

Here N is the length of $f(x)$ (i.e. the number of samples in the sequence), and Δx is the spacing between the grid points assuming the spatial samples are uniformly spaced.

Next, to transform the calculated gradient values to and from the staggered grid, the shift property of the Fourier transform is used, where

$$\mathcal{F}\left\{f(x+a)\right\} = e^{ika} \mathcal{F}\left\{f(x)\right\}. \tag{5}$$

This follows from Eq. (2). Here a is a scalar shift variable which for a staggered grid is set to half the grid spacing, i.e. $\Delta x/2$.

Finally, to integrate forwards in time, an explicit first-order-accurate forward difference is used,[34] where

$$\frac{\partial f}{\partial t} \approx \frac{f^{n+1} - f^n}{\Delta t}. \tag{6}$$

Here f^n represents the discrete function values at the nth time point, where $t = n\Delta t$ and $n = 0, 1, 2, \ldots$. Note, higher-order time integration schemes are also possible (e.g. Refs. 35, 31, 36, 37 and 38). However, these are not discussed here as they do not influence the

implementation of Neumann and Dirichlet boundary conditions using DTTs, which is the focus of this paper.

Combining the three ingredients then gives an explicit time-stepping solution to Eq. (1) using the Fourier PSTD method on a staggered grid (solutions in higher dimensions follow analogously)

$$\begin{aligned} u^{n+\frac{1}{2}} &= u^{n-\frac{1}{2}} - \frac{\Delta t}{\rho_0}\mathcal{F}^{-1}\{ike^{ik\Delta x/2}\mathcal{F}\{p^n\}\}, \\ p^{n+1} &= p^n - \Delta t\rho_0 c_0^2\mathcal{F}^{-1}\{ike^{-ik\Delta x/2}\mathcal{F}\{u^{n+\frac{1}{2}}\}\}. \end{aligned} \tag{7}$$

To run the model, initial values for p and u are required. Time-dependent source terms can also be added,[39] but they are not discussed here. Note, the temporal staggering between the pressure and particle velocity arises because the update steps are interleaved with the spatial gradient calculations. As mentioned in Sec. 1, the use of the discrete Fourier transform inherently assumes the wave field is periodically repeated, which leads to wave wrapping when the waves reach the edge of the computational domain (see Fig. 1). In Sec. 3, the Fourier transforms are replaced with sine and cosine transforms, and in Sec. 4 these are applied to allow Dirichlet and Neumann boundary conditions to be imposed in the solution of the wave equation.

3. Gradient Calculations Using Discrete Trigonometric Transforms

3.1. *The Fourier sine and cosine transforms*

Putting the numerical solution of the wave equation and the imposition of different boundary conditions to one side for a moment, consider now the Fourier sine and cosine transforms and their application to the calculation of spectral gradients. These transforms arise naturally as an extension of the Fourier transform (see e.g. Ref. 40). This can be seen by rewriting the complex exponential basis functions of Eq. (2) in terms of sines and cosines

$$\mathcal{F}\{f(x)\} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)(\cos(kx) - i\sin(kx))dx. \tag{8}$$

If the function $f_o(x)$ is odd such that $f_o(-x) = -f_o(x)$, the cosine terms in Eq. (8) will integrate to zero, which leads to the definition of the forward and inverse Fourier sine transforms

$$\begin{aligned} \mathcal{S}\{f_o(x)\} \equiv g_o(k) &= \sqrt{\frac{2}{\pi}}\int_0^{\infty} f_o(x)\sin(kx)dx, \\ \mathcal{S}^{-1}\{g_o(k)\} \equiv f_o(x) &= \sqrt{\frac{2}{\pi}}\int_0^{\infty} g_o(k)\sin(kx)dk, \quad x \geq 0. \end{aligned} \tag{9}$$

If the input function $f_e(x)$ is instead even such that $f_e(-x) = f_e(x)$, the sine terms in Eq. (8) will integrate to zero, which leads to the definition of the forward and inverse Fourier cosine

transforms

$$\begin{aligned}\mathcal{C}\{f_e(x)\} \equiv g_e(k) &= \sqrt{\frac{2}{\pi}}\int_0^\infty f_e(x)\cos(kx)dx,\\ \mathcal{C}^{-1}\{g_e(k)\} \equiv f_e(x) &= \sqrt{\frac{2}{\pi}}\int_0^\infty g_e(k)\cos(kx)dk, \quad x \geq 0.\end{aligned} \tag{10}$$

Using these definitions, it is possible to show the derivative properties for the sine and cosine transforms. For an odd function, this is

$$\begin{aligned}\frac{d}{dx}f_o(x) &= \frac{d}{dx}\left(\sqrt{\frac{2}{\pi}}\int_0^\infty g_o(k)\sin(kx)dk\right),\\ &= \sqrt{\frac{2}{\pi}}\int_0^\infty k g_o(k)\cos(kx)dk,\\ &= \mathcal{C}^{-1}\{k\mathcal{S}\{f_o(x)\}\}.\end{aligned} \tag{11}$$

Similarly for an even function

$$\frac{d}{dx}f(x) = \mathcal{S}^{-1}\{-k\mathcal{C}\{f_e(x)\}\}. \tag{12}$$

Comparing Eqs. (11) and (12) with Eq. (3), it can be seen that the differentiation properties for the cosine and sine transforms are very similar to those of the Fourier transform. The primary difference is that the function is multiplied by either k or $-k$ after transformation to k-space rather than by $-ik$, and the inverse transform differs from the forward transform. The latter reflects the fact that after differentiation, an odd function becomes even and vice versa. The cosine and sine transforms also produce real-valued outputs if the input function is real-valued, unlike the Fourier transform which produces a complex output.

3.2. *The discrete trigonometric transforms*

In the continuous case, there are two possible symmetries a function may exhibit about $x = 0$, even and odd. This leads to the definition of the two trigonometric transforms described in the previous section. However, in the discrete case, the number of possible symmetries increases to four.[28,29] These symmetries are shown in Fig. 3(a), and correspond to

- whole-sample symmetry (WS),
- whole-sample antisymmetry (WA),
- half-sample symmetry (HS), and
- half-sample antisymmetry (HA).

The terms whole-sample (W) and half-sample (H) describe whether the point of symmetry lies on a grid point, or halfway between two grid points, respectively. The terms symmetric

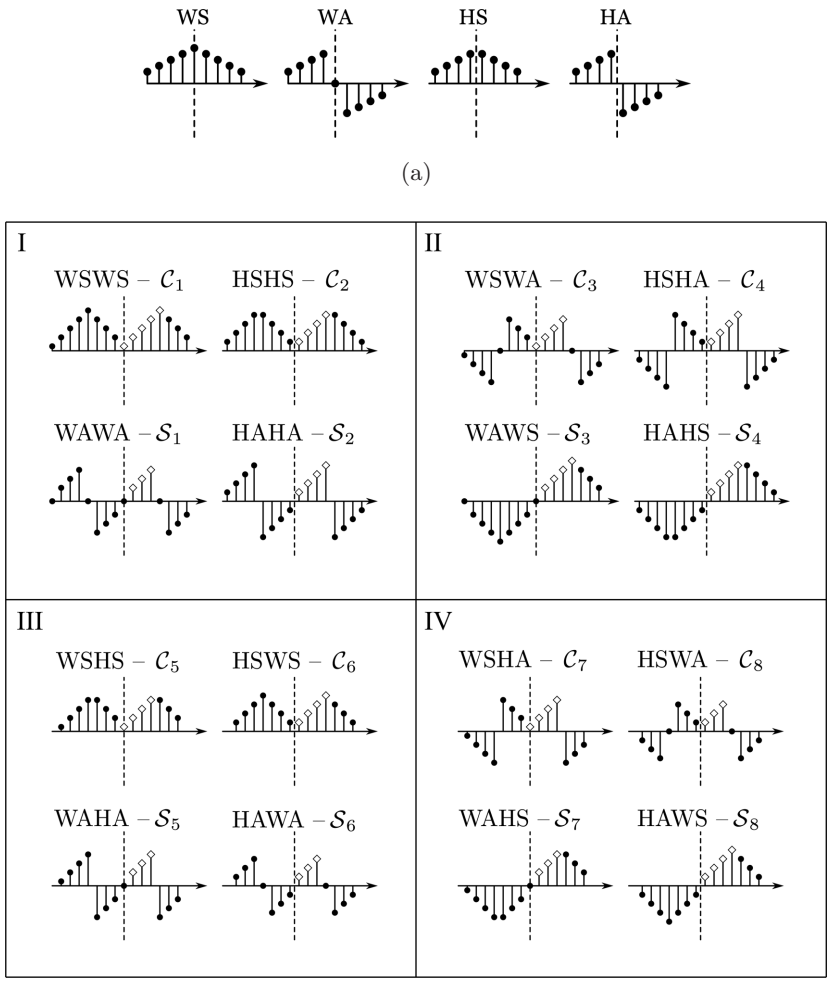

Fig. 3. Discrete trigonometric transforms (DTTs) and symmetry [29]. (a) The four types of discrete symmetry. The point of symmetry is indicated by the dashed line. (b) Examples of the 16 types of symmetric periodic sequence (SPS), and their corresponding DTTs. The white diamonds indicate the representative samples, and the black circles give the implied symmetric-periodic extension. Each SPS is either periodic or antiperiodic with period M. For the example shown, $M = 8$ for groups I and II, and $M = 7$ for groups III and IV.

(S) and antisymmetric (A) are analogous to even and odd in the continuous case. Together, the four types of symmetry may be used to define 16 types of symmetric periodic sequence (SPS) by periodically extending each end of a discrete signal of finite length (called the representative sample) using one of the four symmetry types.[29] The 16 possible SPSs are shown in Fig. 3(b). The naming convention is based on the symmetric extension used at each end. For instance, a WSHA-type sequence has a WS extension on its left end and a HA extension on its right. The extension is repeated indefinitely, such that the symmetry of a WSHA-type sequence is . . .-WS-HA-WS-HA-. . . and so on.

Each of the 16 possible SPSs correspond to a unique discrete trigonometric transform (DTT) as given in Fig. 3(b), where DST-I is denoted as $\mathcal{S}_1$ and DCT-I as $\mathcal{C}_1$, etc. The eight SPSs that have symmetric extensions on the left correspond to the eight types of discrete cosine transform (DCT), while the eight SPSs with antisymmetric extensions on the left correspond to the eight types of discrete sine transform (DST). The discrete equations for each of these transforms are defined in several places, including Refs. 29, 40, 41, and are not repeated here.

The set of 16 DTTs (eight DCTs and eight DSTs) can be divided into four groups based on whether the corresponding SPS becomes periodic or antiperiodic after it has been symmetrically extended at one end, and whether this period M is an even or odd number. Note here that a sequence is antiperiodic with period M if $f(x) = -f(x + nM)$ for $n = 1, 3, 5, \ldots$. Considering the first factor, groups I and III correspond to the SPSs that are periodic with period M, while groups II and IV are antiperiodic with period M. Considering the second factor, the implied period is always an even number for groups I and II, while the implied period is always an odd number for groups III and IV. The properties of these four groups are summarized in Table 1 and Fig. 3(b).

When using DTTs to transform to and from k-space, attention must be given to the discrete wavenumber vector k, as this varies depending on the DTT used as shown in Table 1. The different wavenumber ranges can be understood by considering the assumed symmetry of each sequence. For example, a sequence of WAWA-type is implicitly assumed to have a zero at either end of the representative sample (see Fig. 3). Because of this, both the DC and Nyquist components of the signal must be zero. This is reflected in the range

Table 1. The 16 DTTs and their corresponding symmetries grouped by wavenumber. The four groups of transforms correspond to those shown in Fig. 3. Here N is the length of the representative sample, M is the corresponding implied period of the symmetric periodic sequence, Δx is the grid spacing, and k_n is the discrete wavenumber.

Periodic				Antiperiodic			
$k_n = \frac{2\pi}{M\Delta x} n$				$k_n = \frac{2\pi}{M\Delta x}\left(n + \frac{1}{2}\right)$			
Symmetry	DTT	$n = \ldots$	$M = \ldots$	Symmetry	DTT	$n = \ldots$	$M = \ldots$
WSWS	$\mathcal{C}_1$	$0, 1, \ldots, \frac{M}{2}$	$2(N-1)$	WSWA	$\mathcal{C}_3$	$0, 1, \ldots, \frac{M}{2} - 1$	$2N$
HSHS	$\mathcal{C}_2$	$0, 1, \ldots, \frac{M}{2} - 1$	$2N$	HSHA	$\mathcal{C}_4$	$0, 1, \ldots, \frac{M}{2} - 1$	$2N$
WAWA	$\mathcal{S}_1$	$1, 2, \ldots, \frac{M}{2} - 1$	$2(N+1)$	WAWS	$\mathcal{S}_3$	$0, 1, \ldots, \frac{M}{2} - 1$	$2N$
HAHA	$\mathcal{S}_2$	$1, 2, \ldots, \frac{M}{2}$	$2N$	HAHS	$\mathcal{S}_4$	$0, 1, \ldots, \frac{M}{2} - 1$	$2N$
WSHS	$\mathcal{C}_5$	$0, 1, \ldots, \frac{M-1}{2}$	$2N-1$	WSHA	$\mathcal{C}_7$	$0, 1, \ldots, \frac{M-1}{2}$	$2N-1$
HSWS	$\mathcal{C}_6$	$0, 1, \ldots, \frac{M-1}{2}$	$2N-1$	HSWA	$\mathcal{C}_8$	$0, 1, \ldots, \frac{M-1}{2}$	$2N+1$
WAHA	$\mathcal{S}_5$	$1, 2, \ldots, \frac{M+1}{2}$	$2N+1$	WAHS	$\mathcal{S}_7$	$0, 1, \ldots, \frac{M-1}{2}$	$2N+1$
HAWA	$\mathcal{S}_6$	$1, 2, \ldots, \frac{M+1}{2}$	$2N+1$	HAWS	$\mathcal{S}_8$	$0, 1, \ldots, \frac{M-1}{2}$	$2N-1$

of wavenumbers for which the discrete trigonometric transform is defined, and thus the appropriate frequency components must be appended or removed when using combinations of transforms for which the wavenumber range varies. The different assumed symmetries between transforms also means the length of the sequence before and after differentiation may change, as the representative samples may be defined at different points in space.

Each of the DTTs is unitary (the basis vectors are orthogonal) and thus the inverse transformations can be defined in terms of the forward transforms as described in Refs. 28, 40. Letting $\mathcal{T}_j$ represent either $\mathcal{C}_j$ or $\mathcal{S}_j$, where $j = 1, 2, \ldots, 8$ is the DCT or DST type, the inverse transform $\mathcal{T}_j^{-1}$ is given by

$$\mathcal{T}_j^{-1} = \begin{cases} \frac{1}{M}\mathcal{T}_{j-1} & \text{if } j = 3, 7 \\ \frac{1}{M}\mathcal{T}_j & \text{if } j = 1, 4, 5, 8 \\ \frac{1}{M}\mathcal{T}_{j+1} & \text{if } j = 2, 6. \end{cases} \tag{13}$$

The $1/M$ scaling factor comes from the discrete transform definitions used in the widely used FFTW library in which the forward transforms are not normalized.[41] These relationships are practically useful when implementing DTT-based collocation methods. Note, however, it is the properties of the inverse transform (and not its related forward transform) that must be used when considering the range of wavenumbers and symmetry for which the transform is defined.

3.3. *Spectral gradient calculations*

Given a discrete input sequence (for example, the acoustic pressure or particle velocity defined on a regularly sampled Cartesian grid), spectral gradient calculation using a discrete trigonometric basis proceeds according to Eqs. (11) and (12), where the continuous cosine and sine transforms are replaced by their discrete counterparts from Table 1. The forward transform is chosen based on the symmetry of the input sequence (which is linked to the desired boundary conditions as discussed in Secs. 1 and 4.1). This then defines the appropriate inverse transform to use, as the symmetry of the function changes from symmetric to antisymmetric and vice versa after differentiation.[42]

Consider an input sequence f which has N representative samples and WSWA symmetry. From Table 1, the appropriate forward transform to use is $\mathcal{C}_3$. After differentiation, the symmetry of the output sequence will be WAWS, thus the corresponding inverse transform must be $\mathcal{S}_3^{-1}$, which from Eq. (13) can be computed using $\frac{1}{M}\mathcal{S}_2$. Using Eq. (12), the derivative can thus be computed via

$$\frac{d}{dx}f = \mathcal{S}_3^{-1}\{-k\mathcal{C}_3\{f\}\} = \frac{1}{M}\mathcal{S}_2\{-k\mathcal{C}_3\{f\}\}, \tag{14}$$

where $M = 2N$.

Despite the apparent simplicity of DTT-based gradient calculation, there is one complicating factor related to its numerical implementation. As mentioned in Sec. 3.2, within each of the transform groups, the DTTs may be defined for different ranges of the same set of wavenumbers (see Table 1). For instance, the $\mathcal{C}_1$ transform is defined for $n = 0, 1, \ldots, \frac{M}{2}$ where $k_n = \frac{2\pi}{M} n$, while the $\mathcal{S}_1$ transform is defined for $n = 1, \ldots, \frac{M}{2} - 1$. Thus, to differentiate a sequence of WSWS-type in an analogous manner to Eq. (14), the first and last frequency components (corresponding to $n = 0, \frac{M}{2}$) must be dropped from the discrete array before taking the inverse transform to obtain the differentiated result. For some combinations of DTTs, the reverse is true, and zeros must be appended and prepended to the discrete array before taking the inverse. The required sequence of operations for calculating the gradient of a function using DTTs is summarized in Table A.1 for periodic transforms (groups I and III) and in Table A.4 for antiperiodic transforms (groups II and IV).

3.4. *Interpolation on staggered grids*

DTTs can also be used for interpolation analogous to the shift property of the Fourier transform given in Eq. (5). The general relations are discussed by Britanak,[40] although in some circumstances the expressions are rather complex. The simplest case is resampling a function from whole-sample symmetry to half-sample symmetry and vice versa (this corresponds to shifting by half a grid point as required for staggered grid calculations). This can be achieved by converting to and from k-space using the appropriate DTTs. For example, to interpolate a sequence of WSWA-type to a sequence of HSHA-type, the function is transformed into k-space using $\mathcal{C}_3$, and back using $\mathcal{C}_4^{-1}$, which from Eq. (13) can be computed using $\frac{1}{M}\mathcal{C}_4$.

$$g = \mathcal{C}_4^{-1}\{\mathcal{C}_3\{f\}\} = \frac{1}{M}\mathcal{C}_4\{\mathcal{C}_3\{f\}\}. \tag{15}$$

Again, attention must be paid to the range of wavenumbers for which the forward and inverse transforms are defined. The required sequence of operations for interpolating a function using DTTs is summarized in Table A.2 for periodic transforms (groups I and III) and in Table A.5 for antiperiodic transforms (groups II and IV).

Of particular interest in the context of PSTD methods is the combination of differentiation and interpolation, where the gradient values are resampled at points shifted forwards or backwards by half the grid point spacing.[32] The combination of these operations follows analogously to the discussion of the individual differentiation and interpolation operations. The required sequence of operations for both differentiating and interpolating a function using DTTs is summarized in Table A.3 for periodic transforms (groups I and III) and in Table A.6 for antiperiodic transforms (groups II and IV). An illustration of a symmetric periodic sequence of WSWS-type after differentiation, interpolation, and differentiation and interpolation is shown in Fig. 4.

Note, each of the four transform groups in Table 1 is closed under differentiation and staggered-grid interpolation. Because of the direct relationship between DTTs and SPSs, this is equivalent to saying that differentiating or interpolating a SPS will always produce

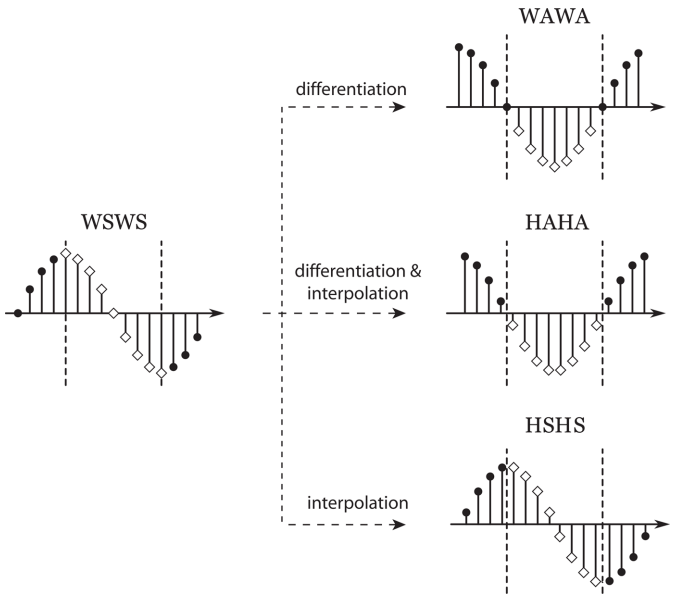

Fig. 4. Illustrative example showing the transformation of a symmetric periodic sequence by differentiation, differentiation + interpolation onto a staggered grid, and interpolation onto a staggered grid. In each case, a different set of group I transformations is used which yields a different symmetry and representative sample size (illustrated with the white diamonds).

another SPS in the same group. The closed nature of these groups is clear when considering that under differentiation the symmetry of a function changes (symmetric ↔ antisymmetric) and that under interpolation the point of symmetry changes (whole-sample ↔ half-sample).

4. The PSTD Method with Discrete Trigonometric Transforms

4.1. *Formulation*

Now that the required ingredients to form a PSTD method using DTTs have been introduced, we can return to the solution of the wave equation given in Eq. (1) subject to different boundary conditions at each end of the domain. In 1D, there are four possible combinations of the left-right boundary condition: Neumann–Neumann, Neumann–Dirichlet, Dirichlet–Neumann, and Dirichlet–Dirichlet. Assuming the acoustic pressure p is calculated on the regular grid and the particle velocity u on the staggered grid (analogous to the Fourier PSTD solution given in Eq. (7)), these correspond to symmetries for the acoustic pressure field of WSWS, WSWA, WAWS, and WAWA, respectively. The particle velocity will instead have half-sample symmetries (i.e. W→H) with the symmetries reversed (i.e. S→A and A→S). These symmetries are summarized in Table 2.

Given the chosen boundary conditions at each end of the domain (and thus the appropriate symmetries and DTTs to use), the spatial gradients shifted to and from the staggered grid can be calculated as discussed in Sec. 3.4. Using the first-order accurate forward difference from Eq. (6) to integrate in time, a DTT-based PSTD solution for Eq. (1) can be

Table 2. Transforms and k-space coefficients used for the DTT-based PSTD solution of the wave equation given in Eq. (16). The k-space coefficients in the periodic case also include shift terms to move to and from the staggered grid points. These shifts are implicit in the DTT pairs specified.

Boundary condition		Sequence type		Transforms		k-space coefficient	
left	right	p	u	$\mathcal{T}_1$	$\mathcal{T}_2$	a_1	a_2
Neumann	Neumann	WSWS	HAHA	$\mathcal{C}_1$	$\mathcal{S}_2$	$-k$	k
Neumann	Dirichlet	WSWA	HAHS	$\mathcal{C}_3$	$\mathcal{S}_4$	$-k$	k
Dirichlet	Neumann	WAWS	HSHA	$\mathcal{S}_3$	$\mathcal{C}_4$	k	$-k$
Dirichlet	Dirichlet	WAWA	HSHS	$\mathcal{S}_1$	$\mathcal{C}_2$	k	$-k$
Periodic				$\mathcal{F}$	$\mathcal{F}$	$ike^{ik\frac{\Delta x}{2}}$	$ike^{-ik\frac{\Delta x}{2}}$

written as

$$u^{n+\frac{1}{2}} = u^{n-\frac{1}{2}} - \frac{\Delta t}{\rho_0}\mathcal{T}_2^{-1}\{a_1\mathcal{T}_1\{p^n\}\}, \tag{16a}$$

$$p^{n+1} = p^n - \Delta t\rho_0 c_0^2\mathcal{T}_1^{-1}\{a_2\mathcal{T}_2\{u^{n+\frac{1}{2}}\}\}. \tag{16b}$$

The transforms $\mathcal{T}_1$ and $\mathcal{T}_2$ and k-space coefficients a_1 and a_2 for the different boundary conditions are given in Table 2. As u has the same symmetry as $\partial p/\partial x$, the forward transform in Eq. (16b) matches the inverse transform used in Eq. (16a) and vice versa.

4.2. *Numerical results*

The discrete equations given in Eq. (16) were implemented in MATLAB (R2019b), with the DTTs computed using a MEX interface to the C++ FFTW library.[43] FFTW provides an efficient implementation of both the FFT and the group I and II DTTs.[41] MATLAB uses FFTW to compute FFTs as part of its inbuilt functions, however, it does not currently provide functions that link to the DTT implementations in FFTW. The MEX interface and example scripts used in this paper are available open-source from GitHub.[44]

To demonstrate the effect of the different boundary conditions on the propagation of an acoustic wave, five simulations were performed using each of the transform combinations outlined in Table 2. The grid size was set to $N = 256$, with a grid spacing of $\Delta x = 1/N$. The sound speed and density were set to the properties of water, with $c_0 = 1500$ m.s^{-1} and $\rho_0 = 1000$ kg.m^{-3}, and the time step was given by a Courant–Friedrichs–Lewy (CFL) number of 0.2, where $\Delta t = \text{CFL}\Delta x/c_0$. The initial pressure distribution p_0 was defined as a Gaussian, and the initial particle velocity u_0 was defined as $u_0 = -p_0/(\rho_0 c_0)$. This gives a traveling wave propagating in the negative x-direction.

The simulation results for each of the boundary conditions given in Table 2 are shown in Fig. 5. When a Fourier basis is used, the wave wraps from one side of the domain to the other when it reaches the boundary, a behavior that is normally counteracted through

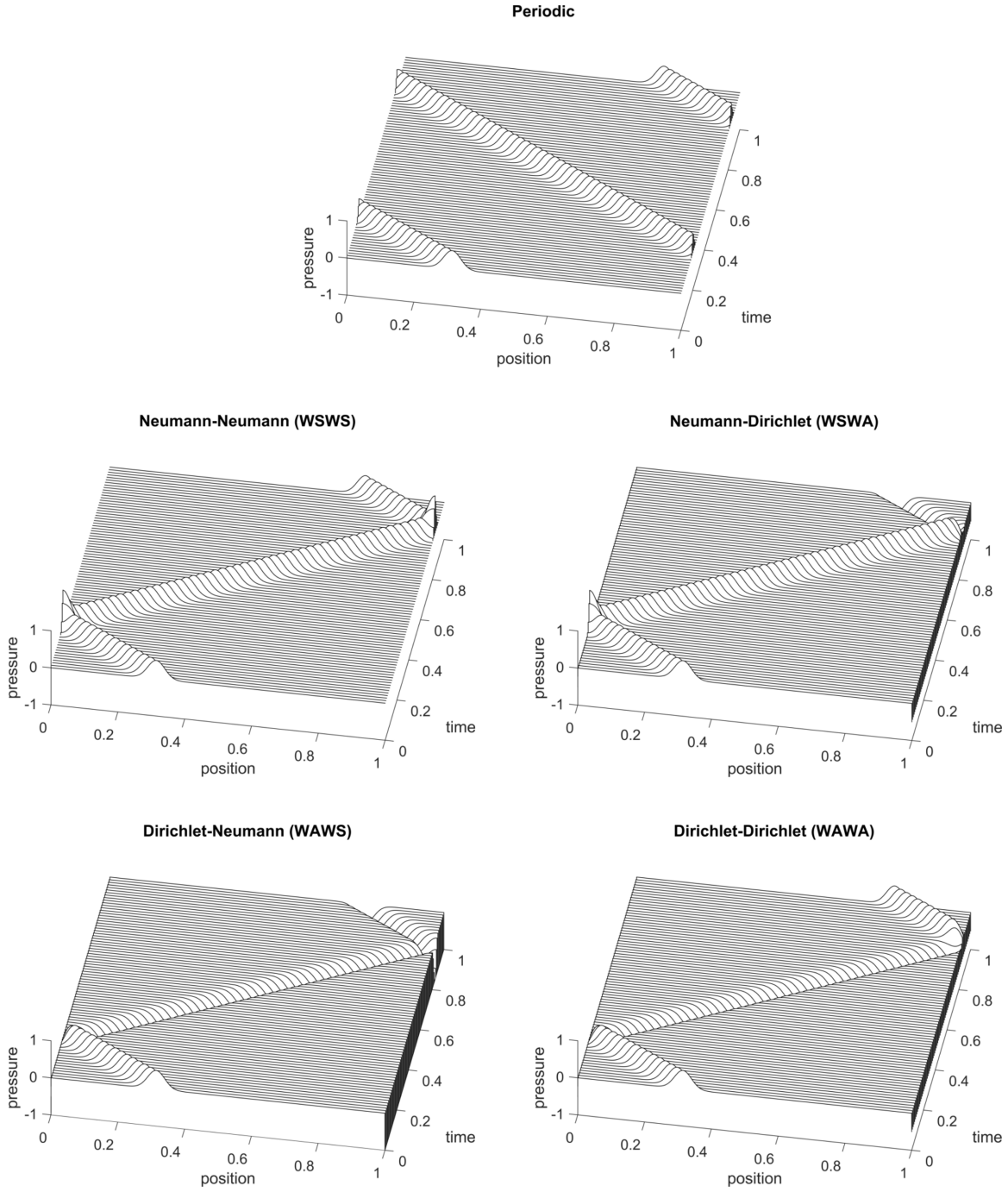

Fig. 5. Propagation of a 1D acoustic wave subject to different boundary conditions. The solution is calculated using a pseudospectral time domain method using either a Fourier basis or a sine or cosine basis for the spatial gradient calculation (see Table 2).

the application of a perfectly matched layer. If a cosine or sine basis is used, the waves are instead reflected from the boundaries. When a symmetric or Neumann boundary condition is used, the reflected wave has the same phase as the incident wave. When an antisymmetric or Dirichlet boundary condition is used, the reflected wave is 180 degrees out-of-phase.

To demonstrate the accuracy of the implemented boundary conditions, additional simulations were performed using a Neumman boundary condition at each end of the domain. In this case, the initial pressure distribution was set to a Gaussian in the center of the grid and the initial particle velocity was set to zero (accounting for the staggered grid). The simulations were repeated for CFL numbers decreasing from 5×10^{-1} to 5×10^{-4} (the stability limit is $2/\pi$). For each simulation, the number of time steps was adjusted such that the outgoing waves propagated exactly to the boundaries and back to the center of the grid where the reflected waves recombine to form the initial condition. The L_∞ error was then calculated by comparing the calculated pressure field at the last time step with the initial condition.

The error convergence against CFL number is shown in Fig. 6. In this case, the errors arise due to numerical dispersion introduced by the time stepping scheme, not due to the accuracy of the boundary condition. As expected, the error reduces algebraically with the CFL number until machine precision is reached. The rate of convergence is governed by the order of the time stepping scheme used, as with conventional Fourier-based PSTD schemes.[4] For the chosen numerical scheme, engineering levels of accuracy, e.g. 0.1%, can be reached with relatively modest CFL numbers.

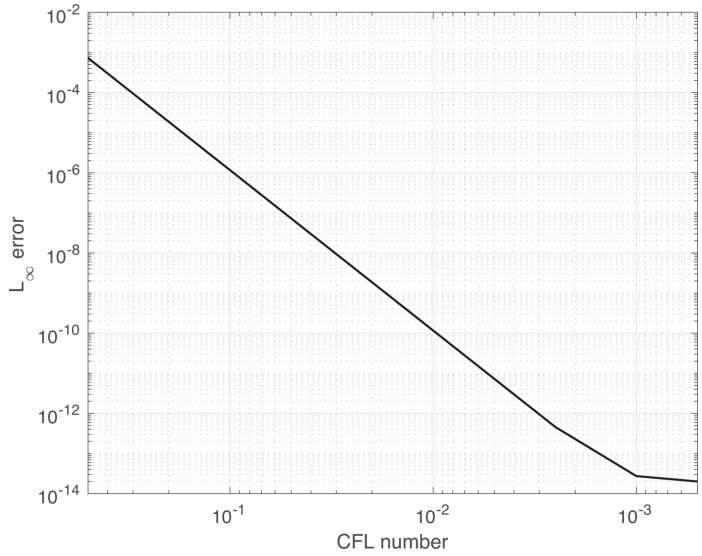

Fig. 6. L_∞ error in the pressure field after propagating a Gaussian initial pressure distribution from the center of the computational domain to the boundaries and back using a DTT-based PSTD method with a Neumann boundary condition at each end of the domain.

4.3. *Higher dimensions*

Much of the previous discussion focuses on the solution of the 1D wave equation using a DTT-based PSTD method. However, it is straightforward to extend this to higher dimensions. For example, in 2D, Eq. (16) becomes

$$\begin{aligned} u_x^{n+\frac{1}{2}} &= u_x^{n-\frac{1}{2}} - \frac{\Delta t}{\rho_0}\mathcal{T}_{x,2}^{-1}\{a_{x,1}\mathcal{T}_{x,1}\{p^n\}\}, \\ u_y^{n+\frac{1}{2}} &= u_y^{n-\frac{1}{2}} - \frac{\Delta t}{\rho_0}\mathcal{T}_{y,2}^{-1}\{a_{y,1}\mathcal{T}_{y,1}\{p^n\}\}, \\ p^{n+1} &= p^n - \Delta t \rho_0 c_0^2 (\mathcal{T}_{x,1}^{-1}\{a_{x,2}\mathcal{T}_{x,2}\{u_x^{n+\frac{1}{2}}\}\} + \mathcal{T}_{y,1}^{-1}\{a_{y,2}\mathcal{T}_{y,2}\{u_y^{n+\frac{1}{2}}\}\}), \end{aligned} \tag{17}$$

where u_x and u_y are the components of the particle velocity in the x and y directions. Again, the transforms $\mathcal{T}_1$ and $\mathcal{T}_2$ and k-space coefficients a_1 and a_2 for the different boundary conditions are given in Table 2. However, in this case, different boundary conditions can be chosen for the x and y directions (denoted by the subscript x and y). This gives rise to 16 possible combinations of Neumann or Dirichlet boundary conditions on each of the four edges of the domain. Two examples are given in Fig. 7 which shows the propagation of a Gaussian initial pressure distribution in a rectangular cavity using either Neumann or Dirichlet boundary conditions on all four boundaries. The application of DTT-based PSTD methods to the solution of other partial differential equations (for example, the diffusion equation) is also straightforward.

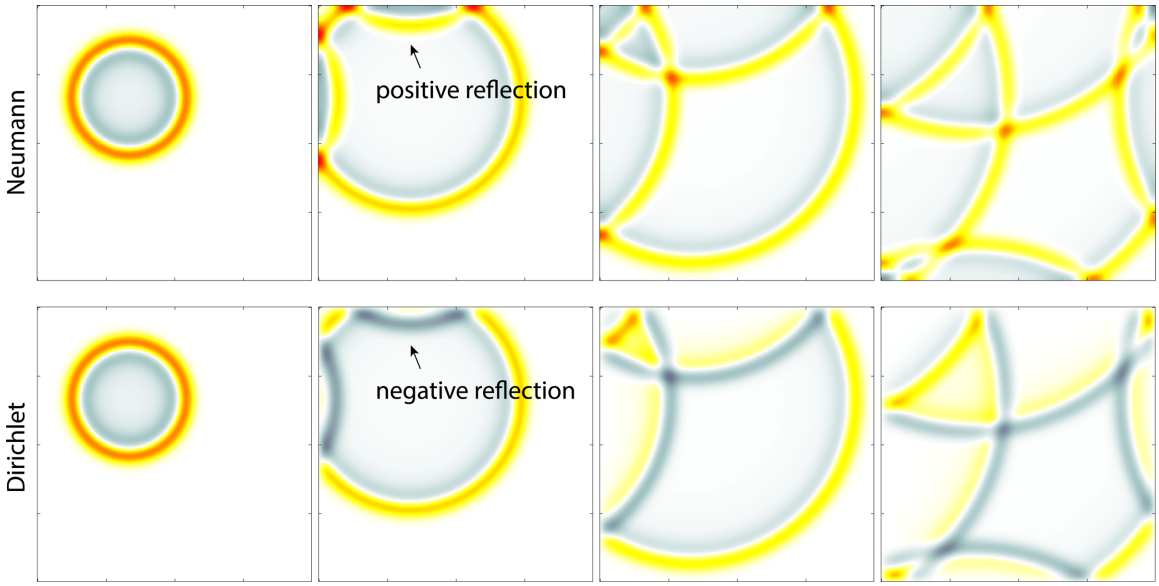

Fig. 7. Four snapshots of the propagation of a 2D acoustic wave in a rectangular cavity subject to Neumann and Dirichlet boundary conditions. The solution is calculated using a pseudospectral time domain method with a sine or cosine basis for the spatial gradient calculation (see Table 2).

4.4. *Computational considerations*

In addition to the ability to implicitly enforce Dirichlet and Neumann boundary conditions, the use of a sine or cosine basis compared to a Fourier basis has several computational advantages. First, the DTTs are all real-to-real transforms (rather than real-to-complex or complex-to-complex). For 2D and 3D transforms, this means the output of the DTTs will consume half the memory required by the corresponding FFT (in 1D, if the input is real, the output of the FFT will be symmetric so only the unique half needs to be stored). Multiplications in k-space are also cheaper to compute, as they operate on real data rather than complex. Finally, interpolation to and from staggered grids is done automatically by selecting the appropriate forward and inverse transforms, rather than by multiplying by a shift-factor in k-space.

In general, the computational complexity of computing the DCT and DST is $O(N \log N)$. However, according to Ref. 41, due to implementation details, the transforms $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{S}_2$, $\mathcal{S}_3$ are the fastest, while the transforms $\mathcal{C}_1$, $\mathcal{S}_1$ can sometimes be significantly slower. The computation time is also significantly less when the implied period M (sometimes called the logical DTT size) has small prime factors. This differs from the FFT, where the best performance is obtained when the sequence length N has small prime factors. Table 1 outlines how to compute M from N for the different transform types.

Other properties of the DTT-based PSTD method are equivalent to conventional Fourier-based schemes. For example, in both cases, the wave field is decomposed into a finite sum of trigonometric basis functions, and thus the same constraints on smoothness and convergence apply (see e.g. Ref. 30). Similarly, the stability limits for the DTT-based scheme are identical to the Fourier case, and higher-order time stepping schemes can equally be used.

5. Nonreflecting Boundary Conditions

As described above and shown in Fig. 5, DTTs can be used to model four basic pairs of boundary conditions in each dimension: Neumann–Neumann, Neumann–Dirichlet, Dirichlet–Neumann, and Dirichlet–Dirichlet. For convenience, these cases will be denoted in terms of the reflection coefficients of the left and right interfaces (R_L, R_R): $(1, 1)$, $(1, -1)$, $(-1, 1)$, and $(-1, -1)$. Interestingly, the solution for more general boundary conditions can be constructed as a linear combination of these four cases (up to a time limit). In other words, the field that would result from any arbitrary real reflection coefficients on the left and right boundaries can be found by summing the fields from these four cases with suitably chosen weights.

To see how to choose the weights, first consider the implied periodicity in the four basic cases, as shown in Fig. 8. In particular, note the polarity and magnitude of the image sources. These can be written into a matrix, where each column corresponds to the polarity of the image sources for a particular pair of boundary conditions, i.e. each column in M

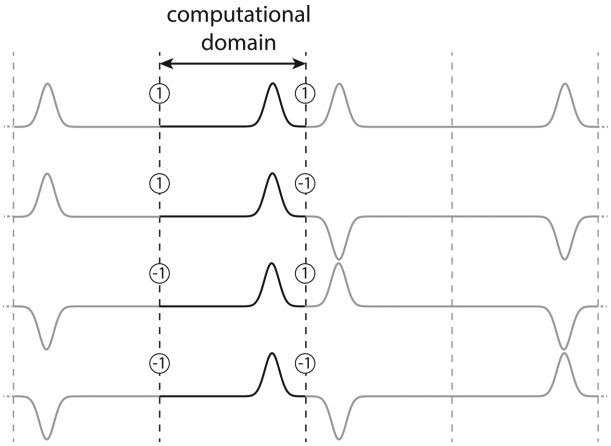

Fig. 8. The implied image sources for the four basic cases of Neumann–Neumann, Neumann–Dirichlet, Dirichlet–Neumann, and Dirichlet–Dirichlet boundary conditions. The corresponding reflection coefficients are shown in circles. One periodic unit is shown. (Fig. 10 shows an extended case.)

corresponds to a row in Fig. 8:

$$\mathrm{M} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{18}$$

Note that M/2 is unitary. Now, if the four fields calculated using the different sets of boundary conditions are written as $\mathbf{p} = [p_{(1,1)}, p_{(1,-1)}, p_{(-1,1)}, p_{(-1,-1)}]^{\mathrm{T}}$, and the corresponding weights for these fields as $\mathbf{w} = [w_{(1,1)}, w_{(1,-1)}, w_{(-1,1)}, w_{(-1,-1)}]^{\mathrm{T}}$, then the field for an arbitrary reflection coefficient at each end of the domain can be written as

$$p_{(R_L,R_R)}(x,t) = \mathbf{w}^{\mathrm{T}}\mathbf{p}(x,t). \tag{19}$$

The weights are calculated from

$$\mathbf{w} = \frac{1}{4}\mathrm{M}^{\mathrm{T}}\mathbf{r}, \tag{20}$$

where $\mathbf{r}$ corresponds to the amplitude of the implied image sources and is given by $\mathbf{r} = [R_L, 1, R_R, R_L R_R]^{\mathrm{T}}$.

One example of interest is the case where both boundaries are nonreflecting, i.e. $R_L = R_R = 0$. It can immediately be seen that the weights in this case are

$$\begin{bmatrix} w_{(1,1)} \\ w_{(1,-1)} \\ w_{(-1,1)} \\ w_{(-1,-1)} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{21}$$

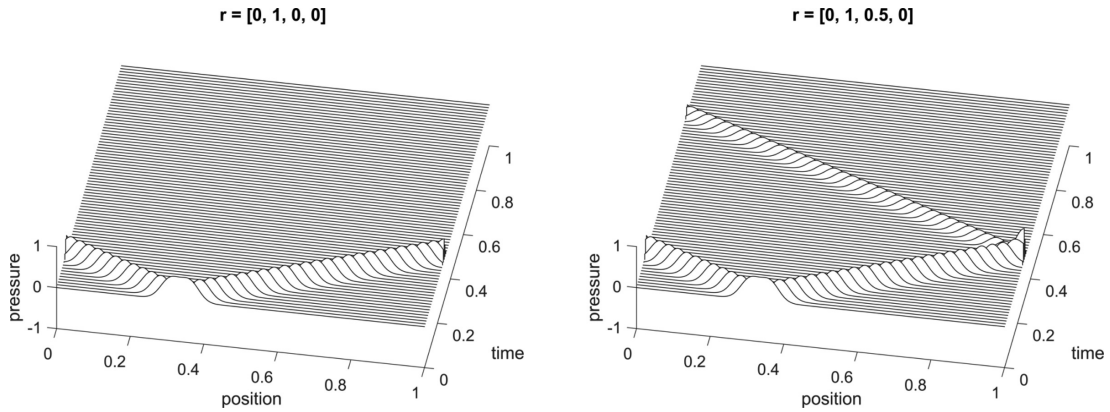

Fig. 9. Propagation of a 1D acoustic wave subject to different reflection coefficients at each end of the domain. The chosen reflection coefficients define the amplitudes of the implied image sources $\mathbf{r}$ and the weightings for the four basic fields $\mathbf{w}$. (left panel) $R_L = 0$ and $R_R = 0$. (right panel) $R_L = 0$ and $R_R = 0.5$.

In other words, the field in the case of nonreflecting boundaries can be constructed by adding together the four basic fields and dividing by four (see Fig. 10). This case is discussed in detail in Ref. 45. Other cases can be calculated similarly. For example, the case for $R_L = 0$ and $R_R = 0.5$ requires weights $\mathbf{w} = [3, 1, 3, 1]^{\mathrm{T}}/8$. Simulation results for these two cases are given in Fig. 9. In this example, the initial pressure is defined as a Gaussian and the initial particle velocity is set to zero, and thus the wave propagates in both directions. After

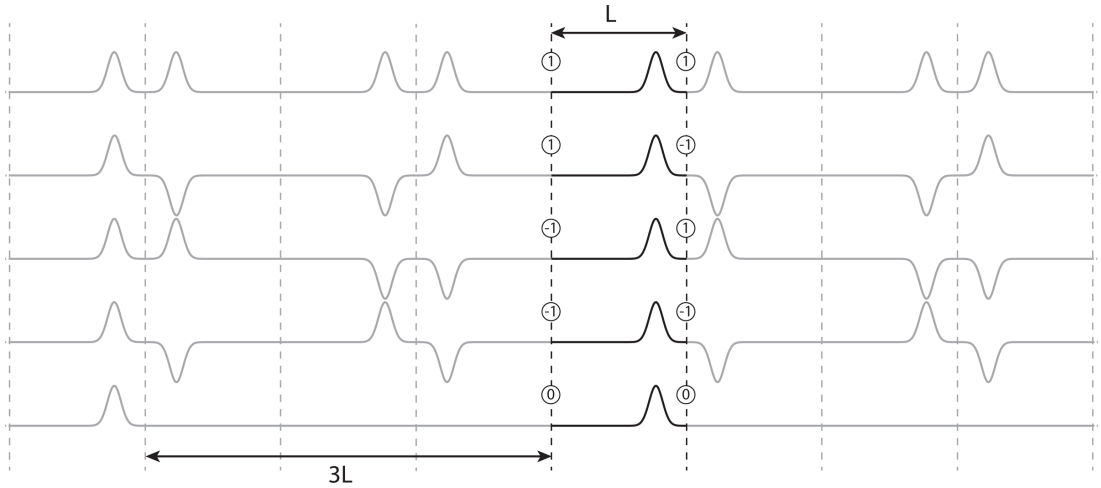

Fig. 10. An extension of Fig. 8, with the bottom line showing the weighted summation of the four basic fields above. This represents the case where the reflection coefficients are both zero. The reflection coefficients are shown in circles. The reason for the time limit of $T = 3L/c_0$ in this case is that the inherent periodicity in the transforms means that after this time there will be an arrival in the computational domain from the image source on the left.

the waves have left the domain, the wave cancellation is theoretically exact, and for the examples give in Fig. 9 the pressure is zero to machine precision.

While this approach allows boundaries with arbitrary real reflection coefficients to be defined (including nonreflecting boundaries), there are two important limitations. First, the linear combination approach only works for times $t = [0, T)$. At time T, additional waves enter the field. The reason for this limit is the inherent periodicity, which is visualized for the nonreflecting case in Fig. 10. In the case when both boundaries are nonreflecting, the limit is $T = 3L/c_0$, where L is the size of the computational domain, although in the general case the limit is stricter, $T = 2L/c_0$. Second, the calculation requires four models to be run in the 1D case, although the models are independent and so can be computed in parallel. The same idea (with the same time limits) can be extended to higher dimensions, however, the number of basic models required grows as 2^{2d}, where d is the number of spatial dimensions. Thus to form a nonreflecting boundary requires 16 simulations in 2D, and 64 in 3D. On the other hand, within the time limits mentioned above, the approach is theoretically exact, which may be an advantage in some situations over conventional absorbing boundary conditions or perfectly matched layers which typically only give a few decimal points of accuracy.

6. Summary

A DTT-based PSTD method for the wave equation is described that allows Neumann (sound-hard) and Dirichlet (sound-soft) boundary conditions to be imposed. The spatial gradients are calculated using a spectral collocation method with a discrete cosine or sine basis, and time integration is performed using a conventional finite difference scheme. The different combinations of staggered grids and boundary conditions lead to the use of 16 different discrete trigonometric transforms (8 DCTs and 8 DSTs). These can be grouped into four distinct classes that are closed under differentiation and staggered-grid interpolation. Practical details of how to use these transforms to compute spatial gradients are provided. Numerical examples of the solution to the wave equation in 1D and 2D in a rectangular domain subject to different boundary conditions are also given. Finally, the extension to boundaries with arbitrary real reflection coefficients is discussed. The DTT library and example scripts used in this paper are available open-source from GitHub.[44]

Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), UK, under grant numbers EP/M011119/1, EP/L020262/1, EP/P008860/1, and EP/S026371/1. This work was also supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602. This work was also supported in part by the Australian Research Council/Microsoft Linkage Project LP100100588.

References

1. B. Fornberg and D. M. Sloan, A review of pseudospectral methods for solving partial differential equations, *Acta Numer.* **3** (1994) 203–267.
2. L. N. Trefethen, *Spectral Methods in MATLAB* (Siam, Philadelphia, 2000).
3. Q. H. Liu, Large-scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (PSTD) algorithm, *IEEE Trans. Geosci. Remote Sens.* **37** (1999) 917–926.
4. J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Dover Publications, Mineola, New York, 2001).
5. J. S. Hesthaven, S. Gottlieb and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Vol. 21 (Cambridge University Press, London, 2007).
6. T. D. Mast, L. P. Souriau, D.-L. Liu, M. Tabei, A. I. Nachman and R. C. Waag, A k-space method for large-scale models of wave propagation in tissue, *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **48** (2001) 341–354.
7. J. L. Robertson, B. T. Cox, J. Jaros and B. E. Treeby, Accurate simulation of transcranial ultrasound propagation for ultrasonic neuromodulation and stimulation, *J. Acoust. Soc. Am.* **141** (2017) 1726–1738.
8. B. Fornberg, The pseudospectral method: Comparisons with finite differences for the elastic wave equation, *Geophysics* **52** (1987) 483–501.
9. N. Albin and O. P. Bruno, A spectral FC solver for the compressible Navier–Stokes equations in general domains I: Explicit time-stepping, *J. Comput. Phys.* **230** (2011) 6248–6270.
10. B. E. Treeby, J. Jaros, A. P. Rendell and B. Cox, Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method, *J. Acoust. Soc. Am.* **131** (2012) 4324–4336.
11. J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, *J. Comput. Phys.* **114** (1994) 185–200.
12. J.-P. Bérenger, Perfectly matched layer (PML) for computational electromagnetics, *Synth. Lect. Comput. Electromagn.* **2** (2007) 1–117.
13. C. Spa, A. Garriga and J. Escolano, Impedance boundary conditions for pseudo-spectral time-domain methods in room acoustics, *Appl. Acoust.* **71** (2010) 402–410.
14. C. Spa, J. Escolano and A. Garriga, Semi-empirical boundary conditions for the linearized acoustic euler equations using pseudo-spectral time-domain methods, *Appl. Acoust.* **72** (2011) 226–230.
15. J. D. Jackson, *Classical Electrodynamics* (Wiley, New York, 1999).
16. L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders, *Fundamentals of Acoustics* (John Wiley and Sons, New York, 1999).
17. B. Cox and P. Beard, Photoacoustic tomography with a single detector in a reverberant cavity, *J. Acoust. Soc. Am.* **125** (2009) 1426–1436.
18. G. Strang, The discrete cosine transform, *SIAM Rev.* **41** (1999) 135–147.
19. D. Kosloff and R. Kosloff, A non-periodic Fourier method for solution of the classical wave equation, *Comput. Phys. Commun.* **30** (1983) 333–336.
20. T. Furumura and H. Takenaka, 2.5D modelling of elastic waves using the pseudospectral method, *Geophys. J. Int.* **124** (1996) 820–832.
21. W.-K. Leung and Y. Chen, Transformed-space nonuniform pseudospectral time-domain algorithm, *Microw. Opt. Technol. Lett.* **28** (2001) 391–396.

22. M. W. Feise, J. B. Schneider and P. J. Bevelacqua, Finite-difference and pseudospectral time-domain methods applied to backward-wave metamaterials, *IEEE Trans. Antennas Propag.* **52** (2004) 2955–2962.
23. A. Bueno-Orovio and V. M. Pérez-García, Spectral smoothed boundary methods: the role of external boundary conditions, *Numer. Methods Partial Differ. Equ. Int. J.* **22** (2006) 435–448.
24. B. Long and E. Reinhard, Real-time fluid simulation using discrete sine/cosine transforms, in *Proc. 2009 Symp. Interactive 3D graphics and games* (2009), pp. 99–106.
25. M. Van Reeuwijk, S. A. Mathias, C. T. Simmons and J. D. Ward, Insights from a pseudospectral approach to the Elder problem, *Water Resour. Res.* **45**.
26. W. Bao, X. Dong and X. Zhao, An exponential wave integrator sine pseudospectral method for the Klein–Gordon–Zakharov system, *SIAM J. Sci. Comput.* **35** (2013) A2903–A2927.
27. I. Ito, Pseudo spectral method based on symmetric extension, in *2018 7th European Workshop Visual Information Processing (EUVIP)* (IEEE, 2018), pp. 1–6.
28. Z. Wang and B. Hunt, The discreteW transform, *Appl. Math. Comput.* **16** (1985) 19–48.
29. S. A. Martucci, Symmetric convolution and the discrete sine and cosine transforms, *IEEE Trans. Signal Process.* **42** (1994) 1038–1051.
30. B. Fornberg, *A Practical Guide to Pseudospectral Methods* (Cambridge University Press, London, 1998).
31. M. Tabei, T. D. Mast and R. C. Waag, A k-space method for coupled first-order acoustic propagation equations, *J. Acoust. Soc. Amer.* **111** (2002) 53–63.
32. B. Fornberg, High-order finite differences and the pseudospectral method on staggered grids, *SIAM J. Numer. Anal.* **27** (1990) 904–918.
33. H.-W. Chen, Staggered-grid pseudospectral viscoacoustic wave field simulation in two-dimensional media, *J. Acoust. Soc. Am.* **100** (1996) 120–131.
34. G. Strang, *Computational Science and Engineering* (Wellesley-Cambridge Press, 2007).
35. G. Wojcik, B. Fomberg, R. Waag, L. Carcione, J. Mould, L. Nikodym and T. Driscoll, Pseudospectral methods for large-scale bioacoustic models, in *1997 IEEE Ultrasonics Symp. Proc.* Vol. 2 (IEEE, 1997), pp. 1501–1506.
36. J. Lu, J. Pan and B. Xu, Time-domain calculation of acoustical wave propagation in discontinuous media using acoustical wave propagator with mapped pseudospectral method, *J. Acoust. Soc. Am.* **118** (2005) 3408–3419.
37. M. Hornikx, R. Waxler and J. Forssén, The extended Fourier pseudospectral time-domain method for atmospheric sound propagation, *J. Acoust. Soc. Am.* **128** (2010) 1632–1646.
38. B. Treeby, E. Wise and B. Cox, Nonstandard Fourier pseudospectral time domain (PSTD) schemes for partial differential equations, *Commun. Comput. Phys.* **24** (2018) 623–634.
39. J. Jaros, A. P. Rendell and B. E. Treeby, Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound, *Int. J. High Perform. Comput. Appl.* **30** (2016) 137–155.
40. V. Britanak, P. C. Yip and K. R. Rao, *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations* (Elsevier, Oxford, 2007).
41. M. Frigo and S. G. Johnson, FFTW (User Manual), Massachusetts Institute of Technology.
42. R. Reeves and K. Kubik, Shift, scaling and derivative properties for the discrete cosine transform, *Signal Process.* **86** (2006) 1597–1603.
43. M. Frigo and S. G. Johnson, The design and implementation of FFTW3, *Proc. IEEE* **93** (2005) 216–231.

44. B. E. Treeby, Matlab discrete trigonometric transform library (2020) https://github.com/ucl-bug/matlab-dtts, Last accessed 25 April 2020.
45. W. D. Smith, A nonreflecting plane boundary for wave propagation problems, *J. Comput. Phys.* **15** (1974) 492–503.

Appendix A. Tables describing DTT methods

Table A.1. Differentiation for periodic SPSs.

Original sequence $\mathbf{u}$			Final sequence $\mathbf{v}$		
Symmetry	Representative sample	Forward	k-space operation	Inverse	Symmetry
WSWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_1\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M}{2}-1}\hat{u}_{\frac{M}{2}-1}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_1\hat{\mathbf{v}}$	WAWA
HSHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_2\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M}{2}-1}\hat{u}_{\frac{M}{2}-1}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_3\hat{\mathbf{v}}$	HAHA
WAWA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_1\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M}{2}-1}\hat{u}_{\frac{M}{2}-1}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_1\hat{\mathbf{v}}$	WSWS
HAHA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_2\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M}{2}-1}\hat{u}_{\frac{M}{2}-1}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_3\hat{\mathbf{v}}$	HSHS
WSHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_5\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_5\hat{\mathbf{v}}$	WAHA
HSWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_6\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_7\hat{\mathbf{v}}$	HAWA
WAHA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M+1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_5\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_5\hat{\mathbf{v}}$	WSHS
HAWA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M+1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_6\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_7\hat{\mathbf{v}}$	HSWS

Table A.2. Interpolation between staggered grids for periodic SPSs.

Original sequence $\mathbf{u}$			Final sequence $\mathbf{v}$		
Symmetry	Representative sample	Forward	k-space operation	Inverse	Symmetry
WSWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_1\mathbf{u}$	$\hat{\mathbf{v}} = \{\hat{u}_0, \ldots, \hat{u}_{\frac{M}{2}-1}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_3\hat{\mathbf{v}}$	HSHS
HSHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_2\mathbf{u}$	$\hat{\mathbf{v}} = \{\hat{u}_1, \ldots, \hat{u}_{\frac{M}{2}-1}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_1\hat{\mathbf{v}}$	WSWS
WAWA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_1\mathbf{u}$	$\hat{\mathbf{v}} = \{\hat{u}_1, \ldots, \hat{u}_{\frac{M}{2}-1}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_3\hat{\mathbf{v}}$	HAHA
HAHA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_2\mathbf{u}$	$\hat{\mathbf{v}} = \{\hat{u}_1, \ldots, \hat{u}_{\frac{M}{2}-1}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_1\hat{\mathbf{v}}$	WAWA
WSHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_5\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_7\hat{\mathbf{v}}$	HSWS
HSWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_6\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_5\hat{\mathbf{v}}$	WSHS
WAHA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M+1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_5\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_7\hat{\mathbf{v}}$	HAWA
HAWA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M+1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_6\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_5\hat{\mathbf{v}}$	WAHA

Table A.3. Differentiation with staggered grids for periodic SPSs.

Original sequence $\mathbf{u}$			Final sequence $\mathbf{v}$		
Symmetry	Representative sample	Forward	k-space operation	Inverse	Symmetry
WSWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_1\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M}{2}}\hat{u}_{\frac{M}{2}}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_3\hat{\mathbf{v}}$	HAHA
HSHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_2\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M}{2}-1}\hat{u}_{\frac{M}{2}-1}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_1\hat{\mathbf{v}}$	WAWA
WAWA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_1\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M}{2}-1}\hat{u}_{\frac{M}{2}-1}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_3\hat{\mathbf{v}}$	HSHS
HAHA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_2\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M}{2}}\hat{u}_{\frac{M}{2}}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_1\hat{\mathbf{v}}$	WSWS
WSHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_5\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_7\hat{\mathbf{v}}$	HAWA
HSWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_6\mathbf{u}$	$\hat{\mathbf{v}} = \{-k_1\hat{u}_1, \ldots, -k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}, 0\}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_5\hat{\mathbf{v}}$	WAHA
WAHA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M+1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_5\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_7\hat{\mathbf{v}}$	HSWS
HAWA	$\mathbf{u} = \{u_1, \ldots, u_{\frac{M+1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_6\mathbf{u}$	$\hat{\mathbf{v}} = \{0, k_1\hat{u}_1, \ldots, k_{\frac{M-1}{2}}\hat{u}_{\frac{M-1}{2}}\}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_5\hat{\mathbf{v}}$	WSHS

Table A.4. Differentiation for antiperiodic SPSs.

Original sequence $\mathbf{u}$			Final sequence $\mathbf{v}$		
Symmetry	Representative sample	Forward	k-space operation	Inverse	Symmetry
WSWA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_3\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_2\hat{\mathbf{v}}$	WAWS
HSHA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_4\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_4\hat{\mathbf{v}}$	HAHS
WAWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_3\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_2\hat{\mathbf{v}}$	WSWA
HAHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_4\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_4\hat{\mathbf{v}}$	HSHA
WSHA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_7\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_6\hat{\mathbf{v}}$	WAHS
HSWA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_8\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_8\hat{\mathbf{v}}$	HAWS
WAHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_7\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_6\hat{\mathbf{v}}$	WSHA
HAWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_8\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k}\cdot\hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_8\hat{\mathbf{v}}$	HSWA

Table A.5. Interpolation between staggered grids for antiperiodic SPSs.

Original sequence $\mathbf{u}$			Final sequence $\mathbf{v}$		
Symmetry	Representative sample	Forward	k-space operation	Inverse	Symmetry
WSWA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_3\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_4\hat{\mathbf{v}}$	HSHA
HSHA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_4\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_2\hat{\mathbf{v}}$	WSWA
WAWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_4\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_4\hat{\mathbf{v}}$	HAHS
HAHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_4\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_2\hat{\mathbf{v}}$	WAWS
WSHA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_7\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_8\hat{\mathbf{v}}$	HSWA
HSWA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_8\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_6\hat{\mathbf{v}}$	WSHA
WAHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_7\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_8\hat{\mathbf{v}}$	HAWS
HAWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_8\mathbf{u}$	$\hat{\mathbf{v}} = \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_6\hat{\mathbf{v}}$	WAHS

Table A.6. Differentiation with staggered grids for antiperiodic SPSs.

Original sequence $\mathbf{u}$			Final sequence $\mathbf{v}$		
Symmetry	Representative sample	Forward	k-space operation	Inverse	Symmetry
WSWA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_3\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_4\hat{\mathbf{v}}$	HAHS
HSHA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{C}_4\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_2\hat{\mathbf{v}}$	WAWS
WAWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_4\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_4\hat{\mathbf{v}}$	HSHA
HAHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M}{2}-1}\}$	$\hat{\mathbf{u}} = \mathcal{S}_4\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_2\hat{\mathbf{v}}$	WSWA
WSHA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_7\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_8\hat{\mathbf{v}}$	HAWS
HSWA	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{C}_8\mathbf{u}$	$\hat{\mathbf{v}} = -\mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{S}_6\hat{\mathbf{v}}$	WAHS
WAHS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_7\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_8\hat{\mathbf{v}}$	HSWA
HAWS	$\mathbf{u} = \{u_0, \ldots, u_{\frac{M-1}{2}}\}$	$\hat{\mathbf{u}} = \mathcal{S}_8\mathbf{u}$	$\hat{\mathbf{v}} = \mathbf{k} \cdot \hat{\mathbf{u}}$	$\mathbf{v} = \frac{1}{M}\mathcal{C}_6\hat{\mathbf{v}}$	WSHA