P-SV wave propagation in heterogeneous media: Velocity-stress distributional finite-difference method

Yder J. Masson, Jean Virieux

P-SV wave propagation in heterogeneous media: Velocity-stress distributional finite-difference method

Yder Masson[*†] *and Jean Virieux*[‡]

[*] *Université de Pau et des Pays de l'Adour, E2S UPPA, CNRS, LFCR, UMR5150, 64000 Pau, France. Email: yder_masson@berkeley.edu*

[†] *Inria, Centre de recherche Bordeaux Sud-Ouest, Magique 3D, 33405 Talence, Aquitaine, France.*

[‡] *Université Grenoble Alpes, ISTerre. CS 40700 38058 GRENOBLE Cedex 9, France. Email: Jean.Virieux@univ–grenoble–alpes.fr*

ABSTRACT

We present a two-dimensional distributional-finite-difference algorithm for modeling the propagation of seismic waves in heterogeneous media in the time domain. We revisit the classic staggered finite-difference algorithm by substituting the standard finite-difference operators with the recently introduced distributional-finite-difference operators with methodological differences we underline. We show that the distributional-finite-difference operators improve simulation accuracy while maintaining the simple structure of the finite-difference algorithm. Thanks to its weak formalism, the newly proposed algorithm accurately and naturally accounts for the free surface, which is a substantial improvement for finite-difference approaches. Moreover, we propose an efficient factorization for the distributional-finite-difference operators. It limits the computational cost of the proposed algorithm to twice that of the finite-difference algorithm. Numerical examples demonstrate that the extra computational burden is well compensated by the superior accuracy of the distributional-finite-difference operators.

INTRODUCTION

Numerical simulations of seismic wave propagation are quite challenging when considering the two speeds of propagation associated with compressive and shear particle motions. In heterogeneous media, the conversions between these two types of waves make it difficult to model the seismic data using asymptotic methods, such as ray theory, because the number of ray paths rapidly becomes considerable in heterogeneous media. To overcome this difficulty, different volumetric approaches, such as the spectral, pseudo-spectral, finite-difference, and finite-element modeling techniques, have

been proposed. A review of these popular modeling methods is carried out in Virieux et al. (2011). Recently, Masson (2022) proposed a distributional finite-difference method, denoted by the DFD keyword thereafter. The DFD approach lies somewhere between the pseudo-spectral/finite-difference methods and the finite/spectral-element methods and combines some desirable features of the former methods in a relatively simple manner. Similarly to the finite-difference approach, DFD is efficient because it relies on compactly supported operations. The CPU time scales linearly with the problem dimension, and heterogeneity can be incorporated by weighting the finite-difference coefficients. Similarly to the finite/spectral-element methods, DFD decomposes the space into multiple elements (with arbitrary sizes), and a volumetric mesh may be employed to better describe the model whenever needed. In DFD, however, the numerical wavefield is discontinuous between neighboring elements; Both the displacement and the stress can be explicitly defined at the surface of the elements. It is not necessary to invert a global mass matrix to proceed with time integration. Boundary conditions are accurately verified with an integration-by-parts procedure. The algorithmic structure of DFD is similar to that of the finite-difference method, and the different field variables may be represented in different spaces or grids. Thus, special care must be taken to eliminate spurious(non-physical) parasitic modes (see, e.g., New et al., 1998), for example, the odd-even decoupling observed when using centered finite-difference operators. In Masson (2022) a MacCormack type of scheme is proposed to address this issue (see e.g., Zhang and Chen, 2006). In this study, we consider an alternative approach where the bases representing the wavefield are analogous to the staggered grid in which the variables are not defined at the same position (as opposed to the collocated grid). Such staggered discretization has found success in electromagnetism (Yee, 1966), in seismic source modeling (Madariaga, 1976) and in seismic wave propagation (Virieux, 1986) following finite-difference techniques. Our objective in this work is an illustration of the 2D P-SV wave propagation, which has been the well-established framework for highlighting the different characteristics of various proposed numerical schemes. The extension to 3D geometry follows the same strategy, which is based on the staggered distribution of unknowns to be computed for honoring the partial differential equations as done with the DFD approach.

The first-order elastodynamic system of equations has retained less attention for continuous finite-element methods. The second-order wave equation has been considered more attractive because of the reduction of unknowns by removing stress components, making the two approaches the often-used method in seismology (Komatitsch et al., 2000a,b; Tromp et al., 2008). A comparison of such approaches is given in Moczo et al. (2001). As far as we know, the staggered discretization using continuous finite-element formalism has been investigated solely by Bécache et al. (2001) with the so-called mixed-finite-element method which discretizes velocity and stress with specific basis functions for stress.

The new method we propose preserves the intrinsic staggered property of the block-anti-diagonal first-order system using staggered basis functions for discretization. The specific choice of the shape functions makes the DFD approach very similar to finite-difference techniques: we shall use B-splines in this work and refer the reader

to the presentation of Masson (2022) for other alternative functions. The boundary conditions are treated with the usual integration-by-parts procedure. Finally, let us underline that the strict staggered discrete design allows employing a parsimonious formulation by removing the stress unknowns at the discrete level as done for finite-difference techniques (Luo and Schuster, 1990). This will not be considered in this work devoted to a practical illustration of the method and not to its high-performance features.

Before detailing the content of our manuscript, let us underline the principal aspects in which our approach differs from the general method introduced in Masson (2022). The first aspect is the choice of the basis functions employed for representing the field variables. Masson (2022) represents the displacements in a first basis (with the basis function skewed to the left) and the stresses in a second basis (with basis functions skewed to the right). In this study, we represent the field variables in different bases and organize the basis functions according to the staggered grid. The staggered approach permits the computation of the partial derivatives by applying 1D operators in a single spatial direction. The method in Masson (2022) requires applying the 1D operators in the two spatial directions. The second aspect in which the two approaches differ is the time scheme. Masson (2022) employs a centered second-order time integration scheme with a displacement formulation. In this study, we employ a first-order leap-frog scheme. Finally, the two studies account for material heterogeneity in different manners. Masson (2022), includes material heterogeneity using a mass lumping technique, where a diagonal mass matrix approximates the exact mass matrix. In the present study, we construct a virtual grid from the basis functions, and we employ standard techniques developed in FDM to discretize material heterogeneity at the grid points.

In the first section, we describe the similarity between previous finite-difference approaches by Virieux (1986) and Levander (1988) and the new distributional-finite-difference formulation of Masson (2022). This will highlight the very specific design of the shape functions for describing each component of the elastodynamic equations in the new formulation. In the second section, we review the numerical structure of the so-called mass and stiffness matrices occurring in weak formulations, and we point out the related finite-difference numerical similarity. In the third section, we introduce the source injection as it is done nowadays in finite-difference approaches. We then describe the averaging procedure for representing the variations of the material properties in the fourth section. The fifth section is dedicated to simple examples with a quantitative comparison of this new DFD approach to standard FD approaches. We start by analyzing the numerical error and computational efficiency and then proceed with more illustrative examples before the conclusion of this novel systematic application of the staggered property at the discretization level.

ELASTODYNAMIC EQUATIONS

In this section, we revisit the classic finite-difference (FD) algorithm for solving the 2D elastodynamic set of equations in a Cartesian coordinate system, i.e., as introduced in Virieux (1986) and later extended in Levander (1988). We propose substituting the standard finite-difference operators with the distributional-finite-difference (DFD) operators introduced in Masson (2022). This substitution preserves the global structure of the former FD algorithm while taking advantage of the superior accuracy of the DFD operators, in particular when accounting for free-surface boundary conditions.

Algorithmic overview

The general algorithm discussed in this section relies on the first-order elastodynamic system of equations

$$\rho\partial_t v_x = \partial_x\sigma_{xx} + \partial_y\sigma_{xy} + f_x \tag{1a}$$
$$\rho\partial_t v_y = \partial_x\sigma_{xy} + \partial_y\sigma_{yy} + f_y \tag{1b}$$

$$\partial_t\sigma_{xx} = (\lambda+2\mu)\partial_x v_x + \lambda\partial_y v_y \tag{1c}$$
$$\partial_t\sigma_{yy} = \lambda\partial_x v_x + (\lambda+2\mu)\partial_y v_y \tag{1d}$$
$$\partial_t\sigma_{xy} = \mu(\partial_x v_y + \partial_y v_x) \tag{1e}$$

where

- v_i is the velocity component in direction i,
- σ_{ij} are the stresses,
- ρ is the density,
- λ, μ are the Lamé parameters,
- ∂_t is the partial derivative with respect to time,
- ∂_i is the partial derivative with respect to spatial direction i.

Let us underline the anti-symmetry of such a differential system which is the backbone of the staggered strategy. In the FD method, the equation 1 is transformed into a discrete linear system as follows. First, by discretizing the velocity, the stresses, and the material properties on staggered grids, as illustrated in Figure 1a. Second, by replacing the partial derivatives with finite-difference operators. For time discretization, a leapfrog scheme is employed. In the proposed algorithm, we replace the spatial partial derivatives with DFD operators, and we represent the field variables using specific orthonormal basis functions. To outline the similarities between the

FD and the DFD algorithms, we write the discrete version of the equation 1 using the general form:

$$\frac{1}{\Delta t}\Big[\mathbf{V}_x(t+\Delta t)-\mathbf{V}_x(t)\Big]= \boldsymbol{\rho}_{21}^{-1}\circ\Big[\mathbf{D}_x^2\cdot\mathbf{S}_{xx}+\mathbf{S}_{xy}\cdot(\mathbf{D}_y^1)^T+\mathbf{f}_x\Big](t+\frac{\Delta t}{2}) \quad (2a)$$

$$\frac{1}{\Delta t}\Big[\mathbf{V}_y(t+\Delta t)-\mathbf{V}_y(t)\Big]= \boldsymbol{\rho}_{12}^{-1}\circ\Big[\mathbf{D}_x^1\cdot\mathbf{S}_{xy}+\mathbf{S}_{yy}\cdot(\mathbf{D}_y^2)^T+\mathbf{f}_y\Big](t+\frac{\Delta t}{2}) \quad (2b)$$

$$\frac{1}{\Delta t}\left[\boldsymbol{S}_{xx}(t+\frac{\Delta t}{2})-\boldsymbol{S}_{xx}(t-\frac{\Delta t}{2})\right]= \Big[\mathbf{M}_{11}\circ(\mathbf{D}_x^1\cdot\mathbf{V}_x)+\boldsymbol{\lambda}_{11}\circ\big(\mathbf{V}_y\cdot(\mathbf{D}_y^1)^T\big)\Big](t) \quad (2c)$$

$$\frac{1}{\Delta t}\left[\boldsymbol{S}_{yy}(t+\frac{\Delta t}{2})-\boldsymbol{S}_{yy}(t-\frac{\Delta t}{2})\right]= \Big[\boldsymbol{\lambda}_{11}\circ(\mathbf{D}_x^1\cdot\mathbf{V}_x)+\mathbf{M}_{11}\circ\big(\mathbf{V}_y\cdot(\mathbf{D}_y^1)^T\big)\Big](t) \quad (2d)$$

$$\frac{1}{\Delta t}\left[\boldsymbol{S}_{xy}(t+\frac{\Delta t}{2})-\boldsymbol{S}_{xy}(t-\frac{\Delta t}{2})\right]= \boldsymbol{\mu}_{22}\circ\Big[\mathbf{D}_x^2\cdot\mathbf{V}_y+\mathbf{V}_x\cdot(\mathbf{D}_y^2)^T\Big](t) \quad (2e)$$

where

$\mathbf{V}_i(t)$ are 2D arrays containing the values or expansion coefficients representing the velocity $v_i(t)$, $\mathbf{V}_x(t)$ has dimensions $(N_x-1)\times N_y$ and $\mathbf{V}_y(t)$ has dimensions $N_x\times(N_y-1)$,

$\boldsymbol{S}_{ij}(t)$ are 2D arrays containing the values or expansion coefficients representing the stresses $\sigma_{ij}(t)$, $\mathbf{S}_{xx}$ and $\mathbf{S}_{yy}$ have dimensions $N_x\times N_y$, and $\mathbf{S}_{xy}$ has dimensions $(N_x-1)\times(N_y-1)$,

$\mathbf{f}_i(t)$ are 2D arrays containing the values or expansion coefficients representing the source or external force field, $\mathbf{f}_x$ has dimensions $(N_x-1)\times N_y$ and $\mathbf{f}_y$ has dimensions $N_x\times(N_y-1)$,

Δt is the time step,

$\mathbf{D}_\theta^k$ are differential matrices approximating the first-order derivative, the superscript k denotes the basis or grid in which the computed derivative is represented and the subscript θ denotes the spatial direction in which the derivative is taken, $\mathbf{D}_x^1$ has dimensions $N_x \times (N_x - 1)$, $\mathbf{D}_x^2$ has dimensions $(N_x - 1) \times N_x$, $\mathbf{D}_y^1$ has dimensions $N_y \times (N_y - 1)$ and $\mathbf{D}_y^2$ has dimensions $(N_y - 1) \times N_y$,

$\boldsymbol{\rho}_{kl}^{-1}$ contain coefficients associated with the inverse of the density, the subscript kl denotes the basis or grid to which the property is associated, $\boldsymbol{\rho}_{21}^{-1}$ has dimensions $(N_x - 1) \times N_y$ and $\boldsymbol{\rho}_{12}^{-1}$ has dimensions $N_x \times (N_y - 1)$,

$\boldsymbol{\lambda}_{kl}$ contain coefficients associated with the Lamé parameter λ, the subscript kl denotes the basis or grid to which the property is associated, $\boldsymbol{\lambda}_{11}$ has dimension $N_x \times N_y$,

$\boldsymbol{\mu}_{kl}$ contain coefficients associated with the shear modulus, the subscript kl denotes the basis or grid to which the property is associated, μ, $\boldsymbol{\mu}_{22}$ has dimension $(N_x - 1) \times (N_y - 1)$,

$\mathbf{M}_{kl}$ contain coefficients associated with the elastic modulus $M = \lambda + 2\mu$, the subscript kl denotes the basis or grid to which the property is associated, $\mathbf{M}_{11}$ has dimension $N_x \times N_y$,

$\cdot$ denotes a standard matrix product,

$\circ$ denotes the element-wise or the Hadamard product.

In the expression above, the indices (i,j) of the arrays correspond to (row, column) and are associated with directions (x,y).

Let us now detail the structure of the operators $\mathbf{D}_\theta^k$ ($k = 1, 2$; $\theta = x, y$) employed in the Virieux and Levander FD algorithms and, for the newly proposed DFD algorithm. In the different numerical illustrations for the three approaches, the same standard leapfrog time integration of the equation 2 is used.

The Virieux algorithm (2^{nd}-order accuracy in space)

In the Virieux algorithm, the elastic field variables and the material properties are discretized on staggered cubic lattices, as illustrated in Figure 1a. More precisely, in equation 2, the 2D arrays

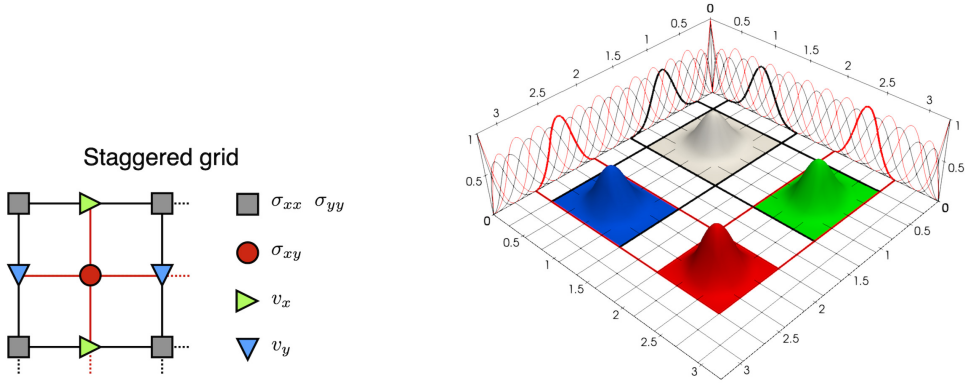
$$\mathbf{V}_x(t) \stackrel{\text{def}}{=} \big[\mathbf{V}_x(t)\big]_{ij} = v_x(x_i^2, y_j^1, t) \tag{3a}$$

$$\mathbf{V}_y(t) \stackrel{\text{def}}{=} \big[\mathbf{V}_y(t)\big]_{ij} = v_y(x_i^1, y_j^2, t) \tag{3b}$$

$$\boldsymbol{S}_{xx}(t) \stackrel{\text{def}}{=} \big[\boldsymbol{S}_{xx}(t)\big]_{ij} = \sigma_{xx}(x_i^1, y_j^1, t) \tag{3c}$$

$$\boldsymbol{S}_{yy}(t) \stackrel{\text{def}}{=} \big[\boldsymbol{S}_{yy}(t)\big]_{ij} = \sigma_{yy}(x_i^1, y_j^1, t) \tag{3d}$$

$$\boldsymbol{S}_{xy}(t) \stackrel{\text{def}}{=} \big[\boldsymbol{S}_{xy}(t)\big]_{ij} = \sigma_{xy}(x_i^2, y_j^2, t) \tag{3e}$$



(a) (b)

Figure 1: (a) shows the position of the wavefield components on the finite-difference grid as introduced in Virieux (1986). (b) shows the 2D staggered sets of B-spline basis functions employed for constructing the orthonormal bases representing the wavefield variables. The $B_{ij}^{11}(x,y)$ in black are used to construct $\hat{B}_{ij}^{11}(x,y)$ representing the normal stresses. The $B_{ij}^{22}(x,y)$ in red are used to construct $\hat{B}_{ij}^{22}(x,y)$ representing the shear stress. $B_{ij}^{21}(x,y)$ in green are used to construct $\hat{B}_{ij}^{21}(x,y)$ representing the horizontal velocity. $B_{ij}^{12}(x,y)$ in blue are used to construct $\hat{B}_{ij}^{12}(x,y)$ representing the vertical velocity. The red and black curves on the sides of the domain represent the 1D sets of staggered B-spline basis functions $B_{x_i}^1$, $B_{x_i}^2$, $B_{y_j}^1$, $B_{y_j}^2$ employed for constructing the 2D basis functions $B_{ij}^{kl}(x,y)$.

contain the values of field variables evaluated at specific grid points

$$x_i^1 = i\Delta x \quad \text{for} \quad i = 1, N_x \tag{4a}$$
$$x_i^2 = (i + 1/2)\Delta x \quad \text{for} \quad i = 1, N_x - 1 \tag{4b}$$
$$y_j^1 = j\Delta y \quad \text{for} \quad j = 1, N_y \tag{4c}$$
$$y_j^2 = (j + 1/2)\Delta y \quad \text{for} \quad j = 1, N_y - 1. \tag{4d}$$

Leaving aside the proper treatment of the boundary conditions, the finite-difference operators $\mathbf{D}_\theta^1$ and $\mathbf{D}_\theta^2$ ($\theta = x, y$) with dimensions $(N_\theta, N_\theta - 1)$ and $(N_\theta - 1, N_\theta)$, respectively, are second-order accurate and have the following band-diagonal structure.

$$\mathbf{D}_\theta^2 = \left| \mathbf{D}_\theta^2 \right| = \begin{bmatrix} -1 & +1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & +1 \end{bmatrix} \tag{5a}$$

$$\mathbf{D}_\theta^1 = -(\mathbf{D}_\theta^2)^T \tag{5b}$$

The boundary conditions must be implemented independently, i.e., without modifying equation 2. For example, by defining artificial boundaries within the computational domain and assigning specific values enforcing the boundary conditions to the field variables outside these boundaries. In this work, we implement the free surface using the image method for Virieux and Levander algorithms (Levander, 1988; Graves, 1996). We refer the reader to Kristek et al. (2002) for alternative approaches. For the DFD algorithm, the boundary conditions are naturally inserted into the discrete operators.

The Levander algorithm (4^{th}-order accuracy in space)

The Levander algorithm has the same geometrical structure as the Virieux algorithm, except that the finite-difference operators $\mathbf{D}_\theta^k$ ($k = 1, 2$; $\theta = x, y$) are now fourth-order accurate leading to following expressions

$$\mathbf{D}_\theta^2 = \left| \mathbf{D}_\theta^2 \right| =$$

$$\begin{bmatrix} -c_1 & +c_1 & -c_2 & 0 & \cdots & 0 \\ +c_2 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & -c_2 \\ 0 & \cdots & 0 & +c_2 & -c_1 & +c_1 \end{bmatrix} \tag{6a}$$

$$\mathbf{D}_\theta^1 = (\mathbf{D}_\theta^2)^T \tag{6b}$$

where $c_1 = 9/8$ and $c_2 = 1/24$ (notice that the coefficients are truncated to the sides).

The distributional-finite-difference algorithm

In the distributional-finite-difference algorithm, a similar strategy is taken, and field variables at a given time are expanded in specific orthonormal bases as the following linear combinations

$$\begin{aligned}
v_x(x,y,t) &= \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_y} \mathrm{v}_{x_{ij}}(t)\hat{B}^{21}_{ij}(x,y) \\
&= \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_y} \mathrm{v}_{x_{ij}}(t)\hat{B}^{2}_{x_i}(x)\hat{B}^{1}_{y_j}(y) \qquad (7a)\\
v_y(x,y,t) &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y-1} \mathrm{v}_{y_{ij}}(t)\hat{B}^{12}_{ij}(x,y) \\
&= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y-1} \mathrm{v}_{y_{ij}}(t)\hat{B}^{1}_{x_i}(x)\hat{B}^{2}_{y_j}(y) \qquad (7b)\\
\sigma_{xx}(x,y,t) &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \mathrm{s}_{xx_{ij}}(t)\hat{B}^{11}_{ij}(x,y) \\
&= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \mathrm{s}_{xx_{ij}}(t)\hat{B}^{1}_{x_i}(x)\hat{B}^{1}_{y_j}(y) \qquad (7c)\\
\sigma_{yy}(x,y,t) &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \mathrm{s}_{yy_{ij}}(t)\hat{B}^{11}_{ij}(x,y) \\
&= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \mathrm{s}_{yy_{ij}}(t)\hat{B}^{1}_{x_i}(x)\hat{B}^{1}_{y_j}(y) \qquad (7d)\\
\sigma_{xy}(x,y,t) &= \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_y-1} \mathrm{s}_{xy_{ij}}(t)\hat{B}^{22}_{ij}(x,y) \\
&= \sum_{i=1}^{N_x-1}\sum_{j=1}^{N_y-1} \mathrm{s}_{xy_{ij}}(t)\hat{B}^{2}_{x_i}(x)\hat{B}^{2}_{y_j}(y) \qquad (7e)
\end{aligned}$$

where the two-dimensional basis functions $\hat{B}^{kl}_{ij}(x,y) = \hat{B}^{k}_{x_i}(x)\hat{B}^{l}_{y_j}(y)$ are composed of a tensorial product of one-dimensional functions $\hat{B}^{k}_{x_i}(x)$ and $\hat{B}^{l}_{y_j}(y)$ representing the field variables in directions x and y, respectively. The 2D orthonormal basis functions $\hat{B}^{kl}_{ij}(x,y)$ are linear combinations of the 2D B-spline basis functions $B^{kl}_{ij}(x,y)$ pictured

in Figure 1b that are shifted similarly to the finite-difference grids. In this study, we construct the 1D orthonormal basis functions $\hat{B}^k_{x_i}(x)$ and $\hat{B}^l_{y_j}(y)$ from the staggered sets of B-spline basis functions

$$\begin{array}{ll} B^1_{x_i}(x) = \beta^p_i(x) & (i = 1, N_x) \\ B^2_{x_i}(x) = \beta^{p-1}_i(x) & (i = 1, N_x - 1) \\ B^1_{y_j}(x) = \beta^p_j(y) & (j = 1, N_y) \\ B^2_{y_j}(y) = \beta^{p-1}_j(y) & (j = 1, N_y - 1) \end{array} \tag{8}$$

where the B-spline basis functions $\beta^p_i(x)$ with polynomial order p and evenly spaced knot vector are described in appendix A and pictured in Figure A-1.

Consequently, in equation 2, the 2D arrays contain the expansion coefficients in the expressions above at the discrete level

$$\mathbf{V}_x(t) \overset{\text{def}}{=} \big[\mathbf{V}_x(t)\big]_{ij} = \mathrm{v}_{x_{ij}}(t) \tag{9a}$$

$$\mathbf{V}_y(t) \overset{\text{def}}{=} \big[\mathbf{V}_y(t)\big]_{ij} = \mathrm{v}_{y_{ij}}(t) \tag{9b}$$

$$\boldsymbol{S}_{xx}(t) \overset{\text{def}}{=} \big[\boldsymbol{S}_{xx}(t)\big]_{ij} = \mathrm{s}_{xx_{ij}}(t) \tag{9c}$$

$$\boldsymbol{S}_{yy}(t) \overset{\text{def}}{=} \big[\boldsymbol{S}_{yy}(t)\big]_{ij} = \mathrm{s}_{yy_{ij}}(t) \tag{9d}$$

$$\boldsymbol{S}_{xy}(t) \overset{\text{def}}{=} \big[\boldsymbol{S}_{xy}(t)\big]_{ij} = \mathrm{s}_{xy_{ij}}(t). \tag{9e}$$

In the next section, the DFD operators $\mathbf{D}^1_\theta$ and $\mathbf{D}^2_\theta$ ($\theta = x, y$) are constructed using an approach analogous to taking the weak form of the wave equation.

NUMERICAL MASS AND STIFFNESS MATRICES, AND DISCRETE DFD OPERATORS

We proceed with a 1D derivation because the 2D DFD operators can be constructed from the 1D DFD operators using a tensor product (Masson, 2022). The DFD operators shall approximate in a shifted basis the derivative of a function represented in a given basis.

Because the velocities and the stresses in Equation 7 are represented in different bases, the objective is to obtain discrete operators that act on a function represented in a first basis B^1 and return its derivative in a second basis B^2, and vice-versa.
Let us underline that such alternate strategy is implicitly attached to the staggered-grid structure of centered-finite difference operators (based on Lagrange polynomials): in a discrete world, a function cannot be evaluated at the same place as its derivative. This feature should be preserved when considering other bases. This is why the four bases we consider in a 2D geometry are spatially shifted similarly to the finite-difference grids. Let us see now how we approximate in a shifted basis the derivative of a function expanded in another basis. The way we proceed differs from the so-called

strong-symmetry strategy of Bécache et al. (2001) limited to meshes with squares whose edges are parallel to the coordinate axis.

We start with two functions $f(\theta)$ and $g(\theta)$ that can be identified as the velocity and the stress wavefields, for example. Both $f(\theta)$ and $g(\theta)$ are defined in a given interval $[\theta_1, \theta_2]$ which corresponds to the computational domain. In what follows, the parameter θ denotes the direction in which the derivative is taken, we have $\theta = x$ or $\theta = y$. Similarly, $[\theta_1, \theta_2] = [x_1, x_2]$ and $[\theta_1, \theta_2] = [y_1, y_2]$ are the bounds of the computational domain in directions x and y, respectively. The functions $f(\theta)$ and $g(\theta)$ are known. The functions $f(\theta)$ are represented in the basis B^1_θ and the functions $g(\theta)$ are represented in the basis B^2_θ, we have:

$$f(\theta) = \sum_{j=1}^{N_\theta} \mathbf{f}_{1_j} B^1_{\theta_j}(\theta) \tag{10a}$$

$$g(\theta) = \sum_{j=1}^{N_\theta - 1} \mathbf{g}_{2_j} B^2_{\theta_j}(\theta) \tag{10b}$$

where $B^1_{\theta_j}(\theta)$ and $B^2_{\theta_j}(\theta)$ are the basis functions. The expansions coefficients $\mathbf{f}_{1_j}$ and $\mathbf{g}_{2_j}$ are known and stored in vectors $\mathbf{f}_1$ and $\mathbf{g}_2$, respectively.

To obtain the derivatives of the functions $f(\theta)$ and $g(\theta)$, one could proceed as usual and compute their exact derivatives $f'(\theta)$ and $g'(\theta)$ using expressions

$$f'(\theta) = \sum_{j=1}^{N_\theta} \mathbf{f}_{1_j} (B^1_{\theta_j})'(\theta) \tag{11a}$$

$$g'(\theta) = \sum_{j=1}^{N_\theta - 1} \mathbf{g}_{2_j} (B^2_{\theta_j})'(\theta) \tag{11b}$$

where $(B^1_{\theta_j})'$ and $(B^2_{\theta_j})'$ are the derivatives of the basis functions $B^1_{\theta_j}$ and $B^2_{\theta_j}$. The problem here is that the derivatives $f'(\theta)$ and $g'(\theta)$ are not represented in the desired bases but as combinations the basis functions' derivatives $(B^1_{\theta_j})'$ and $(B^2_{\theta_j})'$. What we want is a representation of $f'(\theta)$ in the other basis B^2 and similarly a representation of $g'(\theta)$ in the basis B^1. In general, such representations are approximations because the bases B^1 and B^2 define different spaces. Therefore, we need to find two optimal approximations $\tilde{f}'(\theta)$ and $\tilde{g}'(\theta)$ of $f'(\theta)$ and $g'(\theta)$ represented in the bases B^2 and B^1 i.e., we want to find the approximations

$$\tilde{f}'(\theta) = \sum_{j=1}^{N_\theta - 1} \mathbf{f}'_{2_j} B^2_{\theta_j}(\theta) \tag{12a}$$

$$\tilde{g}'(\theta) = \sum_{j=1}^{N_\theta} \mathbf{g}'_{1_j} B^1_{\theta_j}(\theta) \tag{12b}$$

where the expansions coefficients $\mathbf{f}'_{2_j}$ and $\mathbf{g}'_{1_j}$ are unknown and stored in vector $\mathbf{f}'_2$ and $\mathbf{g}'_1$, respectively. We adopt a dual approach to obtain the expansion coefficients $\mathbf{f}'_{2_j}$ and $\mathbf{g}'_{1_j}$ that shall give the best approximations of $f'(\theta)$ and $g'(\theta)$. The representations $\mathbf{f}'_2$ and $\mathbf{g}'_1$ are derived jointly the equalities (a) and (b) in equations 10-15 and 18-20. It shows the important symmetry between the operator $\mathbf{D}^1$ going from basis B^2 to basis B^1 and its twin operator $\mathbf{D}^2$ going from basis B^1 to basis B^2.

To determine the representations $\mathbf{f}'_2$ and $\mathbf{g}'_1$, we adopt a variational approach called the least-squares-residuals method. We start with the desired result or target equalities

$$\tilde{f}'(\theta) = f'(\theta) \tag{13a}$$
$$\tilde{g}'(\theta) = g'(\theta). \tag{13b}$$

By multiplying the first equality with a test function ϕ_i, the second equality with a test function ψ_i, by integrating over the 1D domain $[\theta_1, \theta_2]$, and using integration by parts, we obtain

$$\int_{\theta_1}^{\theta_2} \phi_i \tilde{f}' d\theta = \int_{\theta_1}^{\theta_2} \phi_i f' d\theta = -\int_{\theta_1}^{\theta_2} \phi'_i f d\theta + [\phi_i f]_{\theta_1}^{\theta_2} \tag{14a}$$
$$\int_{\theta_1}^{\theta_2} \psi_i \tilde{g}' d\theta = \int_{\theta_1}^{\theta_2} \psi_i g' d\theta = -\int_{\theta_1}^{\theta_2} \psi'_i g d\theta + [\psi_i g]_{\theta_1}^{\theta_2} \tag{14b}$$

By substituting $f', g', \tilde{f}'$ and $\tilde{g}'$ with their expressions in equations 11 and 12 and, by choosing test functions as $\phi_i = B^2_{\theta_i}$ and $\psi_i = B^1_{\theta_i}$, we obtain the linear systems

$$\mathbf{M}_{\boldsymbol{\theta_2}} \mathbf{f}'_2 = \mathbf{K}_{\boldsymbol{\theta_{21}}}\ \mathbf{f}_1 = -\left[(\mathbf{K}_{\boldsymbol{\theta_{12}}})^T - \mathbf{B}_{\boldsymbol{\theta_{21}}}\right] \mathbf{f}_1 \tag{15a}$$
$$\mathbf{M}_{\boldsymbol{\theta_1}} \mathbf{g}'_1 = \mathbf{K}_{\boldsymbol{\theta_{12}}}\ \mathbf{g}_2 = -\left[(\mathbf{K}_{\boldsymbol{\theta_{21}}})^T - \mathbf{B}_{\boldsymbol{\theta_{12}}}\right] \mathbf{g}_2. \tag{15b}$$

In the expressions above, the matrix

$$\mathbf{M}_{\boldsymbol{\theta_k}} \stackrel{\text{def}}{=} (\mathbf{M}_{\boldsymbol{\theta_k}})_{ij} = \int_{\theta_1}^{\theta_2} B^k_{\theta_i}(\theta) B^k_{\theta_j}(\theta) d\theta \qquad \left\{ \begin{array}{l} \theta = x, y \\ k = 1, 2 \end{array} \right. \tag{16}$$

is usually called the mass matrix, and the matrix

$$\begin{aligned} \mathbf{K}_{\boldsymbol{\theta_{kl}}} &\stackrel{\text{def}}{=} (\mathbf{K}_{\boldsymbol{\theta_{kl}}})_{ij} \\ &= \int_{\theta_1}^{\theta_2} B^k_{\theta_i}(\theta) (B^l_{\theta_j})'(\theta) d\theta \\ &= -\int_{\theta_1}^{\theta_2} (B^k_{\theta_i}(\theta))' B^l_{\theta_j}(\theta) d\theta + \left[B^k_{\theta_i}(\theta) B^l_{\theta_j}(\theta)\right]_{\theta_1}^{\theta_2} \\ &\stackrel{\text{def}}{=} -(\mathbf{K}_{\boldsymbol{\theta_{lk}}})^T + \mathbf{B}_{\boldsymbol{\theta_{kl}}} \end{aligned} \tag{17}$$

for $\theta = x, y$ and $(k, l) = (1, 2), (2, 1)$ is usually called the stiffness matrix. The matrix $\mathbf{B}_{\boldsymbol{\theta_{kl}}}$ accounts for the boundary conditions. With the bases considered in this study,

the elements of $\mathbf{B}_{\theta_{\mathbf{kl}}}$ are all zero except the first one $(\mathbf{B}_{\theta_{\mathbf{kl}}})_{11} = -1$ and the last one $(\mathbf{B}_{\theta_{\mathbf{kl}}})_{mn} = 1$. The relations in equations 15 routinely appear in finite/spectral element analysis. However, in the present case, let us notice that the stiffness matrices are constructed using two different sets of basis functions (and, consequently, two different sets of test functions). In other words, such a weak formulation alternates minimization between two least-squares-residuals expressions.

In principle, one could stop here and solve the linear system in Equations 15 to obtain the representations $\mathbf{f}_2'$ and $\mathbf{g}_1'$. However, to ensure the numerical reciprocity and the stability of the numerical scheme, we want to obtain adjoint pairs of operators. This means that the operator $\mathbf{D}^1 = \mathbf{D}$ going from basis B^2 to basis B^1 is the transpose of its twin operator $\mathbf{D}^2 = \mathbf{D}^T$ going from basis B^1 to basis B^2 (once the boundary conditions have been applied). Such operators are obtained using adequate basis transformations. We follow the approach introduced in Masson (2022). We take the Cholesky factorizations of the mass matrices

$$\mathbf{M}_{\theta_{\mathbf{2}}} = \mathbf{L}_{\theta_{\mathbf{2}}} \mathbf{L}_{\theta_{\mathbf{2}}}^T \tag{18a}$$

$$\mathbf{M}_{\theta_{\mathbf{1}}} = \mathbf{L}_{\theta_{\mathbf{1}}} \mathbf{L}_{\theta_{\mathbf{1}}}^T \tag{18b}$$

and using the changes of variable

$$\hat{\mathbf{f}}_2' = \mathbf{L}_{\theta_{\mathbf{2}}}^T \mathbf{f}_2' \qquad \hat{\mathbf{f}}_1 = \mathbf{L}_{\theta_{\mathbf{1}}}^T \mathbf{f}_1 \tag{19a}$$

$$\hat{\mathbf{g}}_1' = \mathbf{L}_{\theta_{\mathbf{1}}}^T \mathbf{g}_1' \qquad \hat{\mathbf{g}}_2 = \mathbf{L}_{\theta_{\mathbf{2}}}^T \mathbf{g}_2 \tag{19b}$$

the two linear systems displayed in equations 15 become

$$\hat{\mathbf{f}}_2' = \mathbf{D}_\theta^2 \, \hat{\mathbf{f}}_1 \tag{20a}$$

$$\hat{\mathbf{g}}_1' = \mathbf{D}_\theta^1 \, \hat{\mathbf{g}}_2 \tag{20b}$$

where the vectors $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{g}}_2$ now contain the expansion coefficients representing $f(\theta)$ and $g(\theta)$ in some new orthonormal bases $\hat{B}_\theta^1$ and $\hat{B}_\theta^2$, respectively. Similarly, the vectors $\hat{\mathbf{f}}_2'$ and $\hat{\mathbf{g}}_1'$ contain the expansion coefficients representing the approximate derivatives $\tilde{f}'(\theta)$ and $\tilde{g}'(\theta)$ in the orthonormal bases $\hat{B}_\theta^2$ and $\hat{B}_\theta^1$, respectively. The new basis functions are linear combinations of the original ones. We have

$$\hat{B}_{\theta_i}^k(\theta) = \sum_{j=1}^{N_\theta^k} \left(\mathbf{L}_{\theta_{\mathbf{k}}}^{-1}\right)_{ij} B_{\theta_j}^k(\theta) \qquad \left\{ \begin{array}{l} \theta = x, y \\ k = 1, 2 \end{array} \right. \tag{21}$$

and they are orthonormal in the sense that

$$\hat{\mathbf{M}}_{\theta_{\mathbf{k}}} \overset{\text{def}}{=} (\hat{\mathbf{M}}_{\theta_{\mathbf{k}}})_{ij} = \int_{\theta_1}^{\theta_2} \hat{B}_{\theta_i}^k(\theta) \hat{B}_{\theta_j}^k(\theta) d\theta = \delta_{ij} \overset{\text{def}}{=} \mathbf{I} \tag{22}$$

where δ_{ij} is the Kronecker Delta. Notice that the orthogonalization of the basis functions naturally occurs during the derivation of the adjoint operators, and the bases we start with do not need to be orthogonal. Notice that, in general, one could

use any factorization of the form $\mathbf{M} = \mathbf{A}\mathbf{A}^T$ for the mass matrices; the Cholesky factorization has been selected for its simplicity and efficiency. Finally, the adjoint DFD operators in Equation 20 are

$$\mathbf{D}^1_\theta = \mathbf{L}^{-1}_{\theta_1}\mathbf{K}_{\theta_{12}}(\mathbf{L}^{-1}_{\theta_2})^T = -(\mathbf{D}^2_\theta)^T + \mathbf{L}^{-1}_{\theta_1}\mathbf{B}_{\theta_{12}}(\mathbf{L}^{-1}_{\theta_2})^T \tag{23a}$$

$$\mathbf{D}^2_\theta = \mathbf{L}^{-1}_{\theta_2}\mathbf{K}_{\theta_{21}}(\mathbf{L}^{-1}_{\theta_1})^T = -(\mathbf{D}^1_\theta)^T + \mathbf{L}^{-1}_{\theta_2}\mathbf{B}_{\theta_{21}}(\mathbf{L}^{-1}_{\theta_1})^T, \tag{23b}$$

where one can see the important symmetry between directions 1 and 2. When setting the last right-hand-side terms of the equation 23 equal to zero, we observe that the inverse of the transpose of the DFD operators also give the expansion coefficients of the approximate derivative $\tilde{f}'(\theta)$ and $\tilde{g}'(\theta)$ but subject to the boundary conditions $f(\theta_1) = 0$ and $f(\theta_2) = 0$ and $g(\theta_1) = 0$ and $g(\theta_2) = 0$, respectively. Therefore, the boundary condition at the surface computational domain Equation 2 can be implemented as follow:

- To obtain a free surface, the last right-hand-side terms of the equation 23 must be taken equal to zero in the operators in Equations 2a, 2b and 35, *e.g.* as done in the finite/spectral element method
- To obtain a fixed surface, the last right-hand-side terms of the equation 23 must be taken equal to zero in the operators in Equations 2c, 2d and 2e.

In these cases, because the DFD operators are adjoint, the wave equation remains self-adjoint when substituting the partial derivative with the discrete operators. Such a very important feature at the discretization level ensures the numerical reciprocity and stability of the proposed algorithm.

Practical implementation of the discrete operators

As opposed to the FD operators in equations 5 and 6, the DFD operators are full matrices. However, they depend on band diagonal matrices only, and their products with a vector can be computed efficiently with the following matrix structures:

$$\left|\boxed{\mathbf{D}^1_\theta}\right| = \left|\mathbf{L}_{\theta_1}\right|^{-1}\left|\mathbf{K}_{\theta_{12}}\right|\left|\mathbf{L}^T_{\theta_2}\right|^{-1} \tag{24a}$$

$$\left|\boxed{\mathbf{D}^2_\theta}\right| = \left|\mathbf{L}_{\theta_2}\right|^{-1}\left|\mathbf{K}_{\theta_{21}}\right|\left|\mathbf{L}^T_{\theta_1}\right|^{-1} \tag{24b}$$

where

- $\mathbf{L}_{\theta_1}$ is an $N_\theta \times N_\theta$ lower triangular matrix with bandwidth $p+1$ computed from equations 16 and 18.

- $\mathbf{L}_{\theta_2}$ is an $(N_\theta - 1) \times (N_\theta - 1)$ lower triangular matrix with bandwidth p computed from equations 16 and 18.
- $\mathbf{K}_{\theta_{12}}$ is computed from the equation , it is an $N_\theta \times (N_\theta - 1)$ band diagonal matrix with $p-1$ and p non zero diagonals above and below its upper left corner, respectively. Its total bandwidth is $2p-1$.
- $\mathbf{K}_{\theta_{21}}$ is computed from the equation , it is an $(N_\theta - 1) \times N_\theta$ band diagonal matrix with p and $p-1$ non zero diagonals above and below its upper left corner, respectively. Its total bandwidth is $2p-1$.

As an example, the evaluation of the product $\mathbf{y} = \mathbf{D}_\theta^1 \, \mathbf{x}$ takes the following steps

- Solve $\mathbf{L}_{\theta_2}^T \mathbf{a} = \mathbf{x}$ using a backward substitution algorithm
- Compute $\mathbf{b} = \mathbf{K}_{\theta_{12}} \mathbf{a}$
- Solve $\mathbf{L}_{\theta_1} \mathbf{y} = \mathbf{b}$ using a forward substitution algorithm.

The total computational cost of these operations is roughly equivalent to that of applying a finite-difference matrix with bandwidth $4p+1$. As a consequence, for a given approximation/polynomial order, the computational cost of the algorithm in the equation 2 is about four times larger with the DFD operators than with the FD operators. In the next section, we propose a more efficient factorization that further reduces the computational cost of the DFD operators to twice that of the FD operators. We shall see that the superior accuracy of the DFD operators compensates for the additional computational burden. For a given problem dimension, to achieve a given accuracy, DFD is faster than FD.

A more efficient factorization of the DFD operators

We now show that the DFD operators can be factorized more efficiently, thanks to the particular choice of basis functions in equations 8. We take advantage of the fact that the derivative of a function represented in the B-spline basis $B^1 = \beta^p$ with polynomial order p, $N_{\theta_1} = N_\theta$ basis functions and knot vector $\mathbf{k_1} = \mathbf{k} = [k_1, \ldots, k_{N+p+1}]$ can be straightforwardly computed and represented in the B-spline basis $B^2 = \beta^{p-1}$ with polynomial order $p-1$, $N_{\theta_2} = N_\theta - 1$ basis functions and knot vector $\mathbf{k_2} = [k_2, \ldots, k_{N+p}]$. By applying the variable changes $\mathbf{g}' = (\mathbf{L}_{\theta_2}^T)^{-1} \hat{\mathbf{f}}_2'$ and $\mathbf{g} = (\mathbf{L}_{\theta_1}^T)^{-1} \hat{\mathbf{f}}_1$ in the equation A-8, from Equation 20 we identify an alternative expression for $\mathbf{D}_\theta^2$, we have

$$\hat{\mathbf{f}}_2' = \mathbf{L}_{\theta_2}^T \mathbf{Q} (\mathbf{L}_{\theta_1}^T)^{-1} \, \hat{\mathbf{f}}_1 = \mathbf{D}_\theta^2 \, \hat{\mathbf{f}}_1, \tag{25}$$

and an expression for the operator $\mathbf{D}_\theta^1$ is obtained using the equation 23a. The DFD operators now take the form

$$\left|\mathbf{D}_\theta^1\right| = \left|\mathbf{L}_{\theta_1}\right|^{-1} \left[\left|\mathbf{Q}_\theta^T\right| \left|\mathbf{L}_{\theta_2}\right| + \left| \begin{matrix} \mathbf{q}_1 \\ \mathbf{A} \\ \mathbf{q}_2 \end{matrix} \right| \right] \tag{26a}$$

$$\left|\mathbf{D}_\theta^2\right| = \left|\mathbf{L}_{\theta_2}^T\right| \left|\mathbf{Q}_\theta\right| \left|\mathbf{L}_{\theta_1}^T\right|^{-1} \tag{26b}$$

where

- $\mathbf{Q}_\theta$ is the $(N_\theta - 1) \times N_\theta$ banded matrix with bandwidth 2 in the equation A-9.
- The vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ are the first and last rows of the matrix $\mathbf{A} = \mathbf{B}_{\theta_{12}}(\mathbf{L}_{\theta_2}^{-1})^T$ or the first and last columns of the matrix $\mathbf{L}_{\theta_2}^{-1}$.

The bandwidth of the matrix $\mathbf{Q}_\theta$ is constant and independent of the polynomial order p of the basis functions. Thus, the factorization above is more efficient than that in the equation 24. To apply the operators $\mathbf{D}_\theta^1$ and $\mathbf{D}_\theta^2$, one proceeds as follows:

- The product $\mathbf{y} = \mathbf{D}_\theta^1\, \mathbf{x}$ is computed with the following steps:

- Compute $\mathbf{a} = \mathbf{L}_{\theta_2}\mathbf{x}$
- Compute $\mathbf{b} = \mathbf{Q}_\theta^T\mathbf{a}$
- Update the first and last element of $\mathbf{b}$ using

$$\begin{array}{lcl} b_1 & \Leftarrow b_1 + & \mathbf{q}_1^T \cdot \mathbf{x} \\ b_N & \Leftarrow b_N + & \mathbf{q}_2^T \cdot \mathbf{x} \end{array}$$

- Solve $\mathbf{L}_{\theta_1}\mathbf{y} = \mathbf{b}$ using a forward substitution algorithm.

- The product $\mathbf{y} = \mathbf{D}_\theta^2\, \mathbf{x}$ is computed with the following steps:

- Solve $\mathbf{L}_{\theta_1}^T\mathbf{a} = \mathbf{x}$ using a backward substitution algorithm
- Compute $\mathbf{b} = \mathbf{Q}_\theta\, \mathbf{a}$
- Compute $\mathbf{y} = \mathbf{L}_{\theta_2}^T\mathbf{b}$.

- The product $\mathbf{y} = (\mathbf{D}_\theta^1)^T\, \mathbf{x}$ is computed with the following steps:

- Solve $\mathbf{L}_{\theta_1}^T\mathbf{a} = \mathbf{x}$ using a backward substitution algorithm
- Compute $\mathbf{b} = \mathbf{Q}_\theta\mathbf{a}$

- Compute $\mathbf{y} = \mathbf{L}_{\theta_2}^T \mathbf{b}$
- Update the result vector $\mathbf{y}$ using:

$$\mathbf{y} \Leftarrow \mathbf{y} + a_1 \mathbf{q}_1 + a_N \mathbf{q}_2$$

.

- The product $\mathbf{y} = (\mathbf{D}_\theta^2)^T \mathbf{x}$ is computed with the following steps:

- Compute $\mathbf{a} = \mathbf{L}_{\theta_2} \mathbf{x}$
- Compute $\mathbf{b} = \mathbf{Q}_\theta^T \mathbf{a}$
- Solve $\mathbf{L}_{\theta_1} \mathbf{y} = \mathbf{b}$ using a forward substitution algorithm.

The total computational cost for applying $\mathbf{D}_\theta^1$, $(\mathbf{D}_\theta^1)^T$ and $\mathbf{D}_\theta^2$, $(\mathbf{D}_\theta^2)^T$ is about equivalent to that of applying finite-difference matrices with bandwidth $2p+5$ and $2p+3$, respectively. Thus, when p is large, the algorithm in equations 2 is only twice as expensive with the DFD operators than with the FD operators. The expressions in Equations 24 and 26 are equivalent. The form in Equation 26 that is faster and specific to the B-spline basses is employed in all the numerical examples. However, the proposed algorithm could be implemented using other basis functions. In this case, Equation 24 should be used.

AVERAGING OF THE MATERIAL PARAMETERS

For handling heterogeneous models, Masson (2022) has proposed the inclusion of effective properties in Equation 2 by lumping a weighted version of the identity mass matrix in Equation 22. In this study, we adopt a different approach so that the same procedure can be used to represent heterogeneity in the DFD and the FD algorithms. The idea is to define a virtual staggered grid for DFD where each basis function is associated with a unique grid point. This permits to construct the arrays $\boldsymbol{\rho}^{-1}$, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and $\mathbf{M}$ in Equation 2 by evaluating the material properties at the grid points.

We define the virtual DFD grid coordinates from the 1D basis in the equation 8 through the following expressions:

$$x_i^k = \frac{\int_{L_x} \hat{B}_{x_i}^k \; x \; dx}{\int_{L_x} \hat{B}_{x_i}^k \; dx} \qquad\qquad y_j^l = \frac{\int_{L_y} \hat{B}_{y_j}^l \; y \; dy}{\int_{L_y} \hat{B}_{y_j}^l \; dy}. \tag{27}$$

The coordinates (x_i^k, y_j^l) define the virtual DFD grids similarly to the FD grid coordinate in the equation 4. They can be employed to assign the values to the material

parameters arrays, for example, using:

$$\boldsymbol{\rho}_{kl}^{-1} \stackrel{\text{def}}{=} \left[\boldsymbol{\rho}_{kl}^{-1}\right]_{i,j} = \rho(x_i^k, y_j^l) \tag{28a}$$

$$\boldsymbol{\lambda}_{kl} \stackrel{\text{def}}{=} \left[\boldsymbol{\lambda}_{kl}\right]_{i,j} = \lambda(x_i^k, y_j^l) \tag{28b}$$

$$\boldsymbol{\mu}_{kl} \stackrel{\text{def}}{=} \left[\boldsymbol{\mu}_{kl}\right]_{i,j} = \mu(x_i^k, y_j^l) \tag{28c}$$

$$\mathbf{M}_{kl} \stackrel{\text{def}}{=} \left[\mathbf{M}_{kl}\right]_{i,j} = \lambda(x_i^k, y_j^l) + 2\mu(x_i^k, y_j^l). \tag{28d}$$

In the situation where the medium has heterogeneity at scales smaller than the grid spacing, one can take one further step and average the material properties within the grid cells to obtain effective material parameters. Here, we follow the simple approach by Moczo et al. (2014). The average material properties are computed using the volumetric averages

$$\boldsymbol{\rho}_{kl}^{-1} \stackrel{\text{def}}{=} \left[\boldsymbol{\rho}_{kl}^{-1}\right]_{i,j} = \frac{1}{S_{i,j}} \int_{a_i}^{b_i} \int_{a_j}^{b_j} \rho \, dx \, dy \tag{29a}$$

$$\boldsymbol{K}_{kl} \stackrel{\text{def}}{=} \left[\boldsymbol{K}_{kl}\right]_{i,j} = \frac{1}{S_{i,j}} \left[\int_{a_i}^{b_i} \int_{a_j}^{b_j} \frac{1}{K} \, dx \, dy \right]^{-1} \tag{29b}$$

where $(\boldsymbol{K}, K) = (\boldsymbol{\lambda}, \lambda), (\mathbf{M}, \lambda + 2\mu), (\boldsymbol{\mu}, \mu)$, the grid cells boundaries are

$$a_i = \frac{x_i^k + x_{i-1}^k}{2} \qquad b_i = \frac{x_i^k + x_{i+1}^k}{2} \tag{30a}$$

$$a_j = \frac{y_j^l + y_{j-1}^l}{2} \qquad b_j = \frac{y_j^l + y_{j+1}^l}{2} \tag{30b}$$

and the cell surfaces and

$$S_{i,j} = (b_i - a_i)(b_j - a_j). \tag{31}$$

In the numerical examples, Equation 29 is used to compute the material parameters in both the FD and the DFD algorithms. We shall show that this simple strategy leads to quite accurate simulations.

POINT SOURCE REPRESENTATION

In the numerical examples, we represent the seismic source using the moment tensor point source

$$\mathbf{f}(\mathbf{x}, t) = -\nabla \cdot [\, \boldsymbol{M} \, \delta(\mathbf{x} - \mathbf{x}_s)] \, s(t) \tag{32}$$

where $s(t)$ is the source time function, $\boldsymbol{M}(t)$ is the moment tensor, δ is the delta function and $\mathbf{x}$ is the source position vector. A point force vector source $\boldsymbol{f}$ may also be accounted for using

$$\mathbf{f}(\mathbf{x}, t) = \boldsymbol{f} \, \delta(\mathbf{x} - \mathbf{x}_s) \, s(t) \tag{33}$$

Practically, we account for the delta function using the 2D array

$$\boldsymbol{\delta}_s^{kl} \stackrel{\text{def}}{=} \left(\boldsymbol{\delta}_s^{kl}\right)_{ij} = \iint \phi_{ij}^{kl}(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}_s)\,dx\,dy = \phi_{ij}^{kl}(\mathbf{x}_s) \tag{34}$$

where $\phi_{ij}^{kl}(\mathbf{x}_s)$ are the values of the basis function representing the wavefield at the source position. In the DFD algorithm we use $\phi_{ij}^{kl}(\mathbf{x}_s) = \hat{B}_{ij}^{kl}(\mathbf{x}_s)$ and in the FD algorithm we construct $\phi_{ij}^{kl}(\mathbf{x}_s)$ following the approach from Hicks (2002). In the numerical examples, the 2D source arrays in the equation 2 are computed by applying the differential operators. We have:

$$\mathbf{f}_x^t = \left[M_{xx}\left(\mathbf{D}_x^2 \cdot \boldsymbol{\delta}_s^{11}\right) + M_{xy}\left(\boldsymbol{\delta}_s^{22} \cdot (\mathbf{D}_y^1)^T\right)\right] \cdot s(t) \tag{35a}$$

$$\mathbf{f}_y^t = \left[M_{yx}\left(\mathbf{D}_x^1 \cdot \boldsymbol{\delta}_s^{22}\right) + M_{yy}\left(\boldsymbol{\delta}_s^{11} \cdot (\mathbf{D}_y^2)^T\right)\right] \cdot s(t). \tag{35b}$$

In the case of a point force vector source, one should use

$$\mathbf{f}_x^t = f_x \cdot \boldsymbol{\delta}_s^{21} \cdot s(t) \tag{36a}$$

$$\mathbf{f}_y^t = f_y \cdot \boldsymbol{\delta}_s^{12} \cdot s(t). \tag{36b}$$

In the examples, the source time function employed is the Ricker wavelet

$$s(t) = \left(1 - 2\pi^2 f_0^2 t^2\right) e^{-\pi^2 f_0^2 t^2} \tag{37}$$

where f_0 is the peak frequency.

EXAMPLES

Numerical accuracy and computational efficiency

In this section, we assess the accuracy of the DFD operators in Equation 23 by modeling the natural modes of a string with fixed ends. Such a setup allows for simple analytical removal of the time dispersion error. In general, the time dispersion error can be removed using pre- and post-processing of the numerical solution (see, e.g., Wang and Xu, 2015; Koene et al., 2018). This is of interest in DFD modeling because the error associated with spatial discretization is much smaller than that associated with time discretization. In the future, it would be interesting to develop an optimal time integration scheme where the spatial and temporal errors cancel out (see, e.g., Geller and Takeuchi, 1998). We present the estimated numerical error and the CPU time needed to complete modes simulations, and we discuss the spatial operators' convergence rate for FD and DFD.

The theoretical displacement corresponding to the nth vibrational mode of a string with fixed ends has the form

$$u_n(x,t) = \sin(\frac{n\pi}{L}x)\cos(\frac{n\pi c}{L}t) \tag{38}$$

where L is the length of the string, c is the wave speed, and $u_n(x,t)$ is the solution of the 1D wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{39}$$

subject to boundary conditions

$$u(x=0,t) = 0 \tag{40a}$$
$$u(x=L,t) = 0 \tag{40b}$$

and initial values

$$u(x,t=0) = u_n(x,t) \tag{41a}$$
$$\dot{u}(x,t=0) = \dot{u}_n(x,t) = 0. \tag{41b}$$

Equation 38 represents a standing wave with frequency $f_n = \frac{nc}{2L}$ and wavelength $\lambda = \frac{2L}{n}$. The displacement corresponding to the first 3 normal modes of vibrations ($n = 1, 2, 3.$) at time $t = 0$ is pictured in Figure 2c.

To model the vibrational modes numerically, we employ the one-dimensional version of the algorithm in Equation 2. We have:

$$\frac{1}{\Delta t}\Big[\mathbf{V}(t+\Delta t) - \mathbf{V}(t)\Big] = (c_{\text{eff}})^2\Big[\mathbf{D}^2 \cdot \boldsymbol{S}(t+\frac{\Delta t}{2})\Big] \tag{42a}$$

$$\frac{1}{\Delta t}\left[\boldsymbol{S}(t+\frac{\Delta t}{2}) - \boldsymbol{S}(t-\frac{\Delta t}{2})\right] = \mathbf{D}^1 \cdot \mathbf{V}(t) \tag{42b}$$

where the vectors $\mathbf{V}$ and $\mathbf{S}$ represent the velocity and the stress along the string and c_{eff} is an effective wave speed employed to remove the time dispersion (see e.g., Wang and Xu, 2015; Koene et al., 2018). We have

$$c_{\text{eff}} = \frac{c}{\theta} \qquad \theta = \frac{\sin\left(\frac{\omega\Delta t}{2}\right)}{\frac{1}{2}\omega\Delta t} \qquad \omega = 2\pi f_n = \frac{\pi n c}{L}. \tag{43}$$

For a given mode of vibration n, the effective velocity c_{eff} ensures that the frequency of the mode modeled numerically is the same as that of the theoretical solution in Equation 38.

For the FD simulations, the operators $\mathbf{D}^1$ and $\mathbf{D}^2$ are constructed from the finite difference coefficients. The cases p =2 and 4 correspond to the matrices in Equations 5 and 6. For DFD, the operators are computed from Equation 23.

To impose the initial values according to Equation 41, we initialize the values of the vectors $\mathbf{V}$ and $\mathbf{S}$ at times $t = 0$ and $t = -\frac{\Delta t}{2}$, respectively. For FD, we set $\mathbf{V}(t=0) = 0$ and the values of the stress in vector $\mathbf{S}(t = -\frac{\Delta t}{2})$ are obtained by evaluating the spatial derivative of theoretical solution in Equation 38 at the grid

points. For DFD, we set $\mathbf{V}(t=0)=0$ and the coefficients in the vector $\mathbf{S}(t=-\frac{\Delta t}{2})$ representing the stress in basis $\hat{B}^1$ are estimated using a least square fit of the spatial derivative of the theoretical solution in Equation 38.

To enforce the boundary conditions according to Equation 40, we proceed as follows. For FD, the values in vector $\mathbf{V}(t)$ associated with grid points x_i outside the domain $\Omega=[0,L]$ are assigned by taking the first-order finite-difference derivative of the analytical solution, we have $\dot{u}_n(x_i,t_j)\approx\dot{u}_n^{\mathrm{FD}}(x_i,t_j)=[u_n(x_i,t_j+\frac{\Delta_1}{2})-u_n(x_i,t_j-\frac{\Delta_1}{2})]/\Delta t$ (for consistency with the numerical time scheme). The values in vector $\mathbf{S}(t+\frac{\Delta t}{2})$ associated with grid points outside the domain $\Omega=[0,L]$ are taken as the spatial derivative of the analytical solution in Equation 38. For DFD, the boundary conditions are imposed by evaluating the boundary term on the right-hand-side of Equation 23b from the analytical solution in Equation 38, i.e. using $\left[\mathbf{L}_{\theta_2}^{-1}\mathbf{B}_{\theta_{21}}(\mathbf{L}_{\theta_1}^{-1})^T\right]\boldsymbol{S}(t+\frac{\Delta t}{2})=\mathbf{L}_{\theta_2}^{-1}\mathbf{s}(t+\frac{\Delta t}{2})$ where $\mathbf{s}(t)=[-\frac{\partial}{\partial x}u_n(0,t),0,\ldots,0,\frac{\partial}{\partial x}u_n(L,t)]$.

Imposing exact initial value and boundary conditions together with the time-dispersion removal ensures that the numerical error present in the simulations is only related to the intrinsic accuracy of the operators $\mathbf{D}^1$ and $\mathbf{D}^2$.

To compare the numerical solution to the analytical solution in Equation 38, we integrate the velocity field to obtain the displacement. In agreement with the discrete-time integration, we employ the first-order Euler scheme

$$\mathbf{u}[i,1]=u_n(x_i,t_1=-\frac{\Delta t}{2}) \tag{44a}$$

$$\mathbf{u}[i,j]=\mathbf{u}[i,j-1]+\Delta t\mathbf{v}[i,j] \tag{44b}$$

where $\mathbf{u}[i,j]$ contains the displacement evaluated at the grid points (x_i,t_j) where $t_i=(j-\frac{3}{2})\Delta t\quad(j=1,\ldots,N_t)$ and $x_i=(i-1)\Delta x\quad(i=1,\ldots,N_x)$. The initial displacement $\mathbf{u}[i,1]$ is computed from the analytical solution in Equation 38. The vector $\mathbf{v}$ is obtained by evaluating the velocity $\mathbf{V}(t)$ at the grid points (x_i,t_j). We estimate the total numerical error in each simulation using the norm

$$\epsilon=\frac{||\mathbf{u}-\mathbf{u}_{\mathrm{analytical}}||}{N_x\times N_t} \tag{45}$$

where the solution $\mathbf{u}_{\mathrm{analytical}}$ contain the values of the theoretical displacement in Equations 38 evaluated at the grid points (x_i,t_j).

In Figure 2, we present results obtained when modeling the first 50 modes of vibrations using the FD and DFD operators with the order of accuracy p =2, 4, 6, and 8. We ran 400 simulations, one simulation per method (FD/DFD), order of accuracy ($p=2,4,6,8$) and mode ($n=1,\ldots,50$). In all simulations, the length of the string is $L=1$ m and the wave speed c_{eff} is derived from the theoretical value $c=1$ m/s. The duration of the simulations is 1 s (i.e. half the period of the fundamental mode $n=1$), the number of time steps is $N_t=500$ and the time step is $\Delta t=2\times10^{-3}$ s. For the FD simulations we used $N_x=50$ grid points inside the domain $\Omega=[0,L]$. For the DFD simulations we used $N_x=50$ basis functions to represent the wavefield in the domain $\Omega=[0,L]$.

In Figure 2a, we represent the error ϵ in Equation 45 as a function of the number of points per wavelength ($N_p = N_x/\lambda$). For the FD simulations (in red), when a sufficient number of points per wavelength is employed ($N_p \geq 5$), the numerical error decreases accordingly to the theoretical convergence rate of the finite-difference approximation (dashed lines). For the DFD simulations (in blue), the numerical error decreases much faster than for FD when a few points per wavelength are employed (say $N_p \leq 10$). When the number of points per wavelength is large ($N_p \geq 10$), the convergence rate of DFD is similar to that of FD. However, the error measured for DFD is several orders of magnitude smaller than that measured for FD. This highlights the superior accuracy of DFD, which requires fewer points per wavelength to reach the same accuracy as FD.

Because the computational cost for applying the operators $\mathbf{D}^1$ and $\mathbf{D}^2$ in DFD is about twice that of FD, we verify that this extra computational cost does not annihilate the gain in accuracy observed for DFD. For a problem with a given physical size, the computational cost of a simulation is proportional to the number of points per wavelength for both FD and DFD. Therefore, we define the effective CPU time by multiplying the number of points per wavelength by the CPU time measured for the simulation (i.e. measured using Fortran's intrinsic function **cpu_time**). In Figure 2b we represent the same data as in Figure 2a, but the numerical error is plotted as a function of the effective CPU time. We observe that the convergence rates as a function of the CPU time are similar for FD and DFD (when $N_p \geq 10$). Because the CPU cost associated with DFD is larger than that of FD, the superior accuracy of DFD is less prominent. However, the numerical error associated with DFD is still about one order of magnitude smaller than that of FD. Therefore, we emphasize that a significant gain in efficiency is expected with DFD, particularly for large simulations in 3D.

Azimuthal dispersion in a homogeneous medium

To investigate the relative accuracy of the numerical algorithms considered, we ran two sets of four simulations where a P-wave and S-wave propagate in a homogeneous medium, as illustrated in Figure 3. Both sets of simulations consist of one simulation using the Virieux algorithm, one simulation using the Levander algorithm, and two simulations using the DFD algorithm with polynomial orders $p = 2$ and $p = 4$ in the equation 8. In all simulations, the computational domain has dimension 6920 m $\times$ 6920 m. The material properties are homogeneous, the P-wave velocity is $v_p = 3000$ m/s, the S-wave velocity is $v_s = 1730$ m/s and the density is$\rho = 2500$ kg/m^3. The seismic source is placed at the center of the computational domain, the source time function is the Ricker wavelet in the equation 37 with peak frequency $f_0 = 10$ Hz and an estimated maximum frequency $f_{max} \approx 2.5 f_0 = 25$ Hz. The moment tensor is taken as

$$\begin{vmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{vmatrix} = \begin{vmatrix} 4 & 1 \\ -1 & 4 \end{vmatrix}, \tag{46}$$

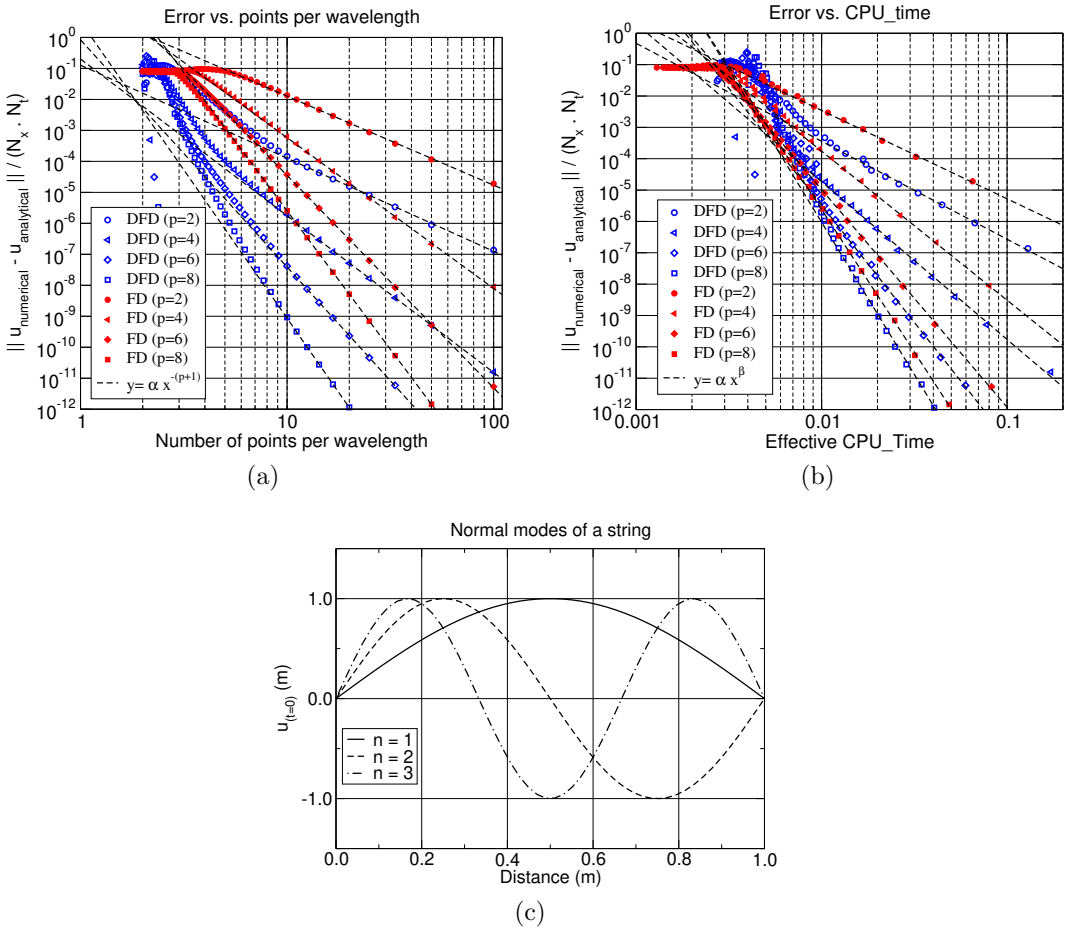


Figure 2: (a) The numerical error ϵ in Equation 45 is plotted as a function of the number of points per wavelength for the first 50 modes. The dashed lines represent the theoretical convergence rate. (b) shows the same data set as (a), but the numerical error is plotted as a function of the effective CPU time (i.e. the product between the number of points per wavelength and the CPU time measured for the simulation). The dashed lines show the least-square fit of the data for $N_p \geq 10$. (c) shows the initial displacement value in the simulations for the first three modes.

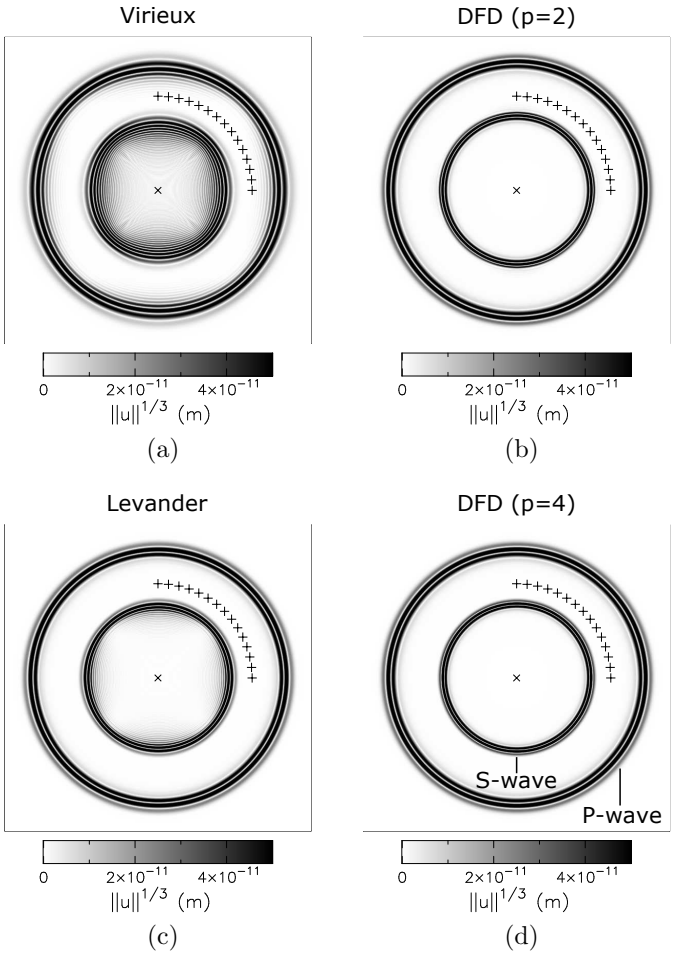


Figure 3: Comparison of the displacement wavefields obtained when modeling P and S wave propagation in a homogeneous medium using different algorithms. The magnitude of the displacement vector is raised to power 1/3 to enhance small amplitudes. A red cross marks the source location, and the receivers are represented using blue pluses. (a) shows the wavefield obtained using the Virieux algorithm, (b) shows the wavefield obtained using the DFD algorithm with $p = 2$, (c) shows the wavefield obtained using the Levander algorithm, and (d) shows the wavefield obtained using the DFD algorithm with $p = 4$.

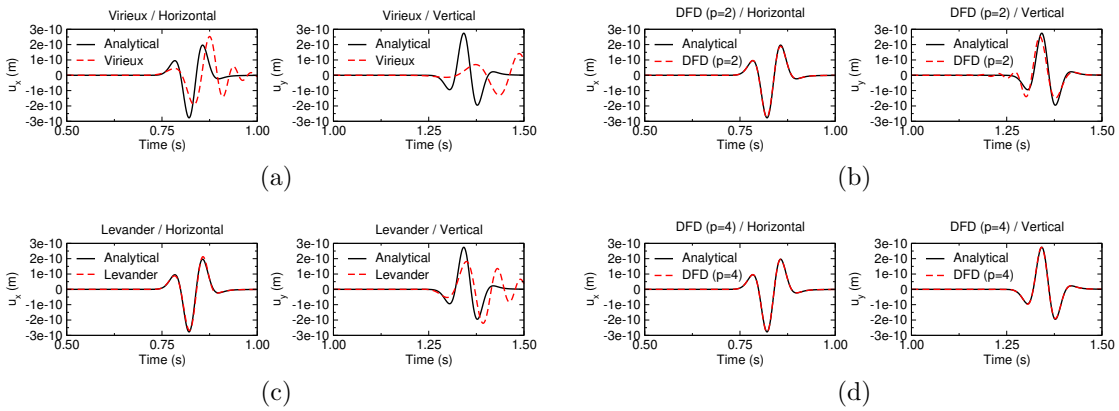


Figure 4: Comparison of the seismograms obtained when modeling wave propagation in a homogeneous medium using different algorithms. The recordings correspond to the lowermost receiver at the same height as the source in Figure 3. The left panels show the horizontal displacement, and the right panels show the vertical displacement. The first wave-packet is associated with the P-wave, and the second wave-packet is associated with the S-wave. The dashed red lines correspond to the computed seismograms, and the solid black lines correspond to the analytical solution. (a) shows the seismograms computed using the Virieux algorithm, (b) shows the seismograms computed using the DFD algorithm with $p = 2$, (c) shows the seismograms computed using the Levander algorithm, and (d) shows the seismograms computed using the DFD algorithm with $p = 4$.

so that the P-wave and the S-wave have about the same amplitude at the receivers. To quantify the numerical dispersion as a function of the propagating direction, the displacement wavefield is recorded at 15 receivers placed at the same distance $d \approx 2129$ m from the source. The angle between the source-receiver direction and the horizontal direction ranges from 0 to $\pi/2$.

In the first set of simulations, we employ a small number of points per wavelength to enhance the numerical dispersion of the different numerical schemes. At the maximum frequency $f_{max} \approx 2.5 f_0$, there are 5.2 and 3.0 grid points per wavelength for the P-wave and the S-wave, respectively. Notice that this is well below the recommended values, which should be around 10 and 5 points per wavelength for the Virieux and the Levander algorithms with the setup considered. It makes the numerical error larger for an easier visual comparison, but the FD algorithms should not be used this way in practical applications. For the FD simulations the grid spacings are $\Delta x = \Delta y \approx 23$ m. In the DFD simulations, the number of basis functions representing the wavefield is the same as the number of grid points in the FD simulations to achieve an equivalent spatial sampling. We choose the time step $\Delta t = 7.7 \times 10^{-4}$ s, small enough to minimize the error due to time discretization.

The snapshots obtained for the first set of simulations are pictured in Figure 3. When comparing the two snapshots obtained for the FD simulations in Figure 3ac, we

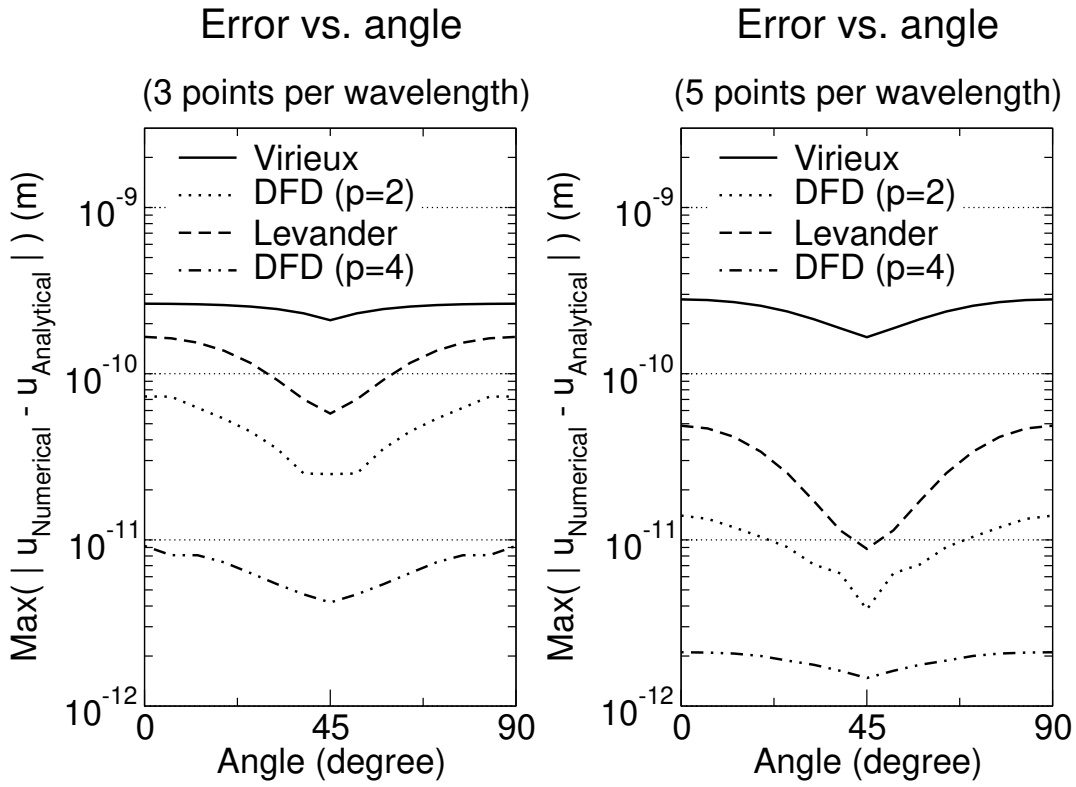


Figure 5: Comparison of the error measured when modeling P and S wave propagation in a homogeneous medium using different algorithms. The error is measured with respect to the angle between the line passing by the source and the receivers and the horizontal direction. It is taken as the maximum of the absolute value of the difference between the computed seismograms $\mathbf{u}_{\text{Numerical}}(t)$ and the analytical solution $\mathbf{u}_{\text{Analytical}}(t)$. These results correspond to the simulations illustrated in Figure 3. The left panel shows the results obtained for the simulations employing 3 points per S-wavelength, and the right panel shows the results obtained for the simulations using 5 points per S-wavelength.

see that the Levander algorithm produces less numerical dispersion than the Virieux algorithm. Thus, the Levander algorithm benefits from the higher order of approximation employed for spatial discretization. For the DFD simulation in Figure 3bd, numerical dispersion decreases drastically. In particular, the DFD simulation with $p = 2$ in Figure 3b shows almost no dispersion compared to results obtained for Virieux and Levander schemes.

In Figure 4, we present the seismograms recorded at the first receiver, which is located at the height of the source and has the most prominent numerical dispersion (together with that exactly above the source). We compare numerical seismograms to analytical solutions obtained using the software gar6more (Diaz and Ezziani, 2010) based on the Cagniard-De Hoop method (De Hoop, 1970). We observe that all algorithms well model the P-wave except the Virieux algorithm, which shows significant numerical dispersion. Only the DFD algorithm with $p = 4$ accurately matches the analytical solution for the S-wave. The DFD algorithm with $p = 2$ matches the analytical solution reasonably well but produces a bit of numerical dispersion. The seismograms computed with the Virieux and Levander algorithms are highly affected by numerical dispersion.

As a more systematic error with respect to azimuthal variations, we measure the maximum difference between computed seismograms and analytical solutions for each receiver represented by crosses in the upper quarter of the Figure 3 at all times. We summarize the results in Figure 5. For all algorithms, we observe that the error is minimum when the waves propagate in the directions 45 degrees away from the horizontal and the vertical directions (where the error is the highest). The Virieux algorithm produces the most significant error regardless of the propagation angle, followed by the Levander, the DFD with $p = 2$, and the DFD with $p = 4$. Interestingly the error decreases significantly faster with the DFD algorithm (i.e., when going from $p = 2$ to $p = 4$) than with the FD algorithm (i.e., when going from the Virieux to the Levander algorithm). We attribute this observation to the higher convergence rate of B-spline bases.

In the second set of simulations, we increase the number of points per wavelength to reduce the numerical dispersion. At the maximum frequency $f_{max} \approx 2.5 f_0$, there are 8.7 and 5.0 grid points per wavelength for the P-wave and the S-wave, respectively. For the FD simulations the grid spacings are $\Delta x = \Delta y \approx 13.8$ m. As for the first set of simulations, the DFD simulations have the same dimension as the FD simulation, and the time step $\Delta t = 4.6 \times 10^{-4}$ s is small enough to limit the dispersion associated with time discretization. The error measured for the second set of simulations is plotted in the right panel in Figure 5. We notice a significant reduction in the numerical error when using five or more points per wavelength. The error reduction is small for the Virieux algorithm because of the low correlation between the computed seismograms and analytical solutions. The DFD algorithm with $p = 4$ has the highest accuracy, followed by the DFD with $p = 2$, the Levander, and the Virieux algorithms. It is interesting to notice that, when using 3 points per minimum wavelength for the DFD algorithm with $p = 2$ (i.e., in the left panel), we obtain an error that is about the

same as that obtained when using 5 points per minimum wavelength for the Levander algorithm (i.e., in the right panel). Thus, the DFD operators significantly reduce the number of points per wavelength needed to limit the numerical dispersion. In 3D, this represents an important gain in computational efficiency and an interesting reduction in memory usage.

Surface wave propagation in a homogeneous medium

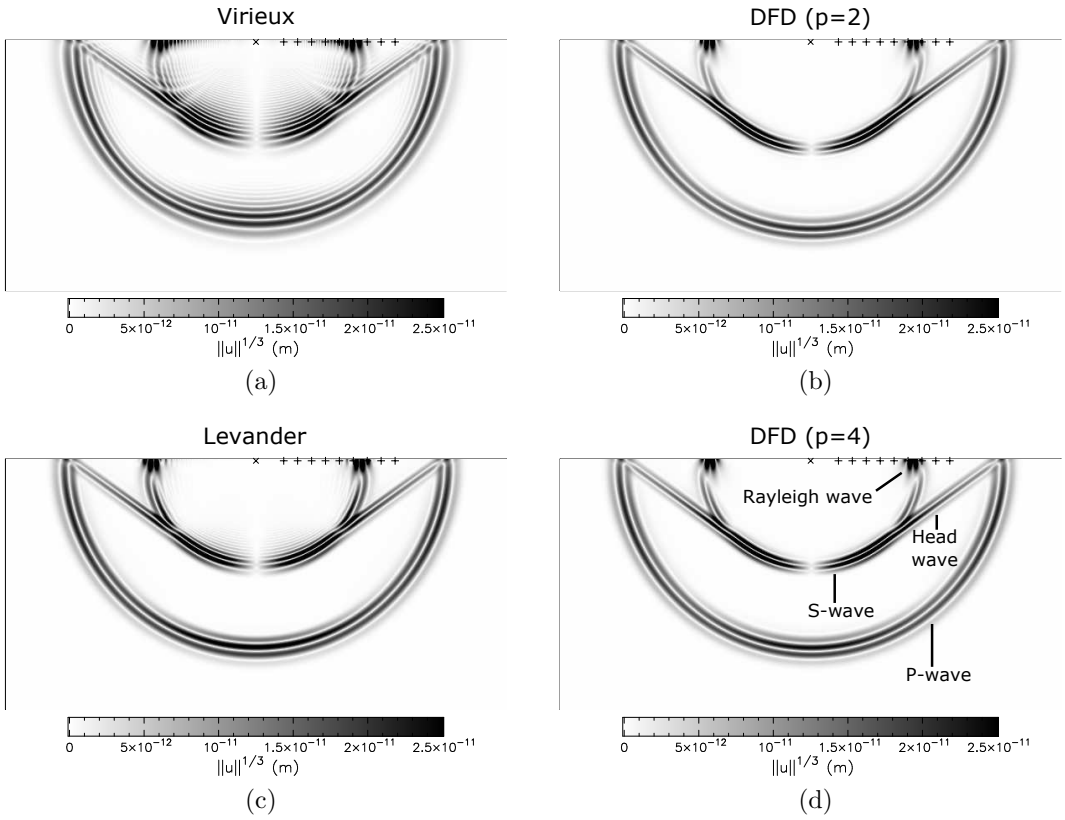


Figure 6: Comparison of the displacement wavefields obtained when modeling surface-wave propagation in a homogeneous medium using different algorithms. The visible seismic phases are labeled in (d). The magnitude of the displacement vector is raised to power 1/3 to enhance small amplitudes. A red cross marks the source location, and the receivers are represented using blue pluses. (a) shows the wavefield obtained using the Virieux algorithm, (b) shows the wavefield obtained using the DFD algorithm with $p = 2$, (c) shows the wavefield obtained using the Levander algorithm, and (d) shows the wavefield obtained using the DFD algorithm with $p = 4$.

In FD modeling, the proper treatment of surface wave propagation is notoriously challenging and has been studied extensively, and the following citations (Robertsson, 1996; Mittet, 2002; Kristek et al., 2002; Cao et al., 2018) could be used for collecting the high number of papers on this topic. In this respect, the DFD algorithm is

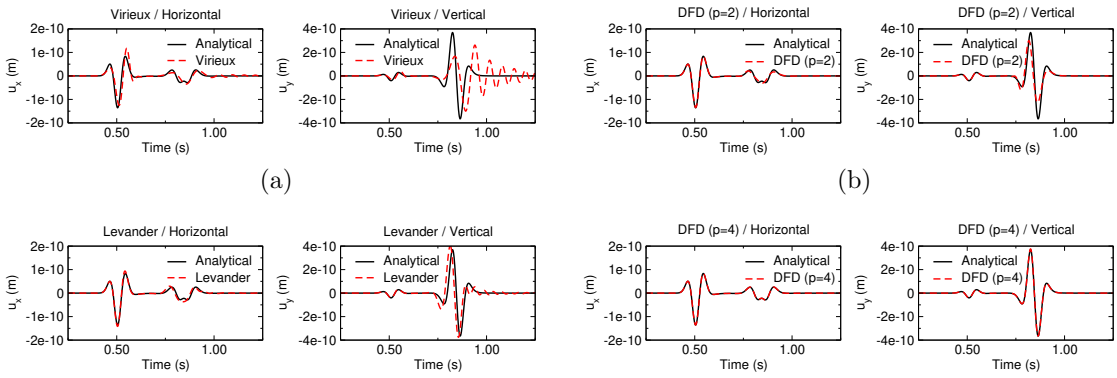


Figure 7: Comparison of the seismograms obtained when modeling surface-wave propagation in a homogeneous medium using different algorithms. The recordings correspond to the mid-point receiver in the receiver line in Figure 6. The left panels show the horizontal displacement, and the right panels show the vertical displacement. The first wave-packet is associated with P-wave, and the second wave-packet is associated with the Rayleigh wave. The dashed red lines correspond to the computed seismograms, and the solid black lines correspond to the analytical solution. (a) shows the seismograms computed using the Virieux algorithm, (b) shows the seismograms computed using the DFD algorithm with $p = 2$, (c) shows the seismograms computed using the Levander algorithm, and (d) shows the seismograms computed using the DFD algorithm with $p = 4$.

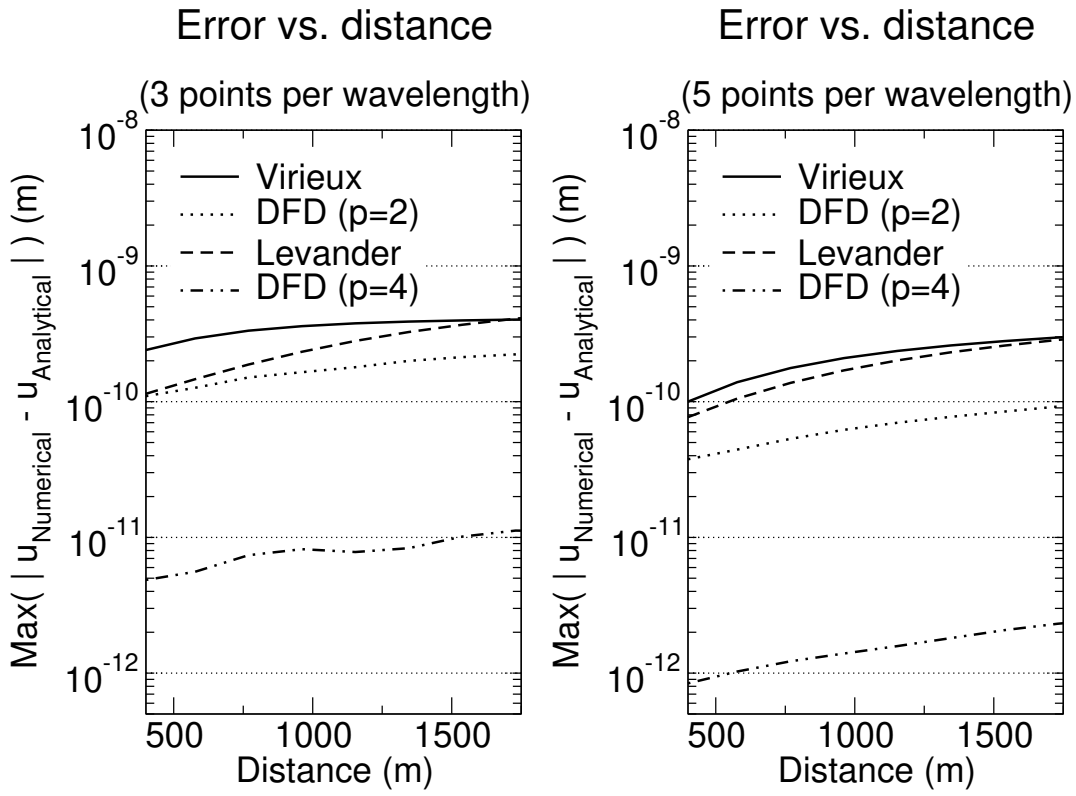


Figure 8: Comparison of the error measured when modeling surface-wave propagation in a homogeneous medium using different algorithms. The error is measured as a function of the distance to the source. It is taken as the maximum of the absolute value of the difference between the computed seismograms $\mathbf{u}_{\text{Numerical}}(t)$ and the analytical solution $\mathbf{u}_{\text{Analytical}}(t)$. These results correspond to the simulations illustrated in Figure 6. The left panel shows the results obtained for the simulations employing 3 points per S-wavelength, and the right panel shows the results obtained for the simulations using 5 points per S-wavelength.

desirable because the free surface is accounted for naturally and accurately by the DFD operators. To illustrate this advantage, following the approach in the previous section, we ran two series of simulations using the four algorithms considered, as illustrated in Figure 6.

In all simulations, the computational domain has dimensions 6920 m by 1460 m. The material properties are homogeneous, the P-wave velocity is $v_p = 3000$ m/s, the S-wave velocity is $v_s = 1730$ m/s and the density is $\rho = 2500$ kg/m^3. The seismic source is placed at the center of the free surface at a depth of 25 m. We simulate an explosive source using the moment tensor

$$\begin{vmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}. \tag{47}$$

The source time function is the Ricker wavelet in the equation 37 with peak frequency $f_0 = 10$ Hz and an estimated maximum frequency $f_{max} \approx 2.5 f_0 = 25$ Hz. The principal waves propagating through the medium are annotated in Figure 6d. Our analysis focuses on the Rayleigh wave propagating at the surface of the velocity model. The displacement wavefield is recorded at nine receivers placed at a depth of 25 m and distances to the source varying from 384 m to 1922 m. We compare the numerical seismograms to analytical solutions obtained using the software *gar6more* (Diaz and Ezziani, 2010).

In the first set of simulations, we employ 5.2 and 3.0 grid points per wavelength for the P-wave and the S-wave, respectively. Remember that this is much smaller than the recommended values, which are 10 and 5 points per wavelength for the Virieux and the Levander algorithms. For the FD simulations the grid spacings are $\Delta x = \Delta y \approx 23$ m. The grid dimensions in the DFD simulations are the same as in the FD simulations. The propagation distance between the source and the furthest receiver is about 25 times the S-wave characteristic wavelength for which numerical dispersion should be noticeable. The time step is $\Delta t = 7.7 \times 10^{-4}$ s, which is sufficiently small to neglect the effects of numerical dispersion associated with the time discretization in our comparisons.

As in the previous examples, the two simulations based on the FD algorithms exhibit significantly more numerical dispersion than those based on the DFD algorithms, and the Levander scheme appears more accurate than the Virieux scheme. The Rayleigh wave spreads significantly as it propagates with the FD algorithms but not with the DFD algorithms.

In Figure 7, we present the seismograms recorded by the receiver located at the center of the receiver line. All algorithms match the analytical solution for the first wave packet associated with the P-wave. Only the DFD algorithm with $p = 4$ accurately matches the analytical solution for the second wave packet associated with the Rayleigh wave. For the DFD algorithm with $p = 2$, the fit is still relatively good, but the surface-wave amplitude is slightly damped. For the FD algorithms, the seismograms are significantly affected by the numerical dispersion, which appears as a

ringing starting with and extending long after the arrival of the surface wave. This ringing is especially pronounced for the Virieux algorithm.

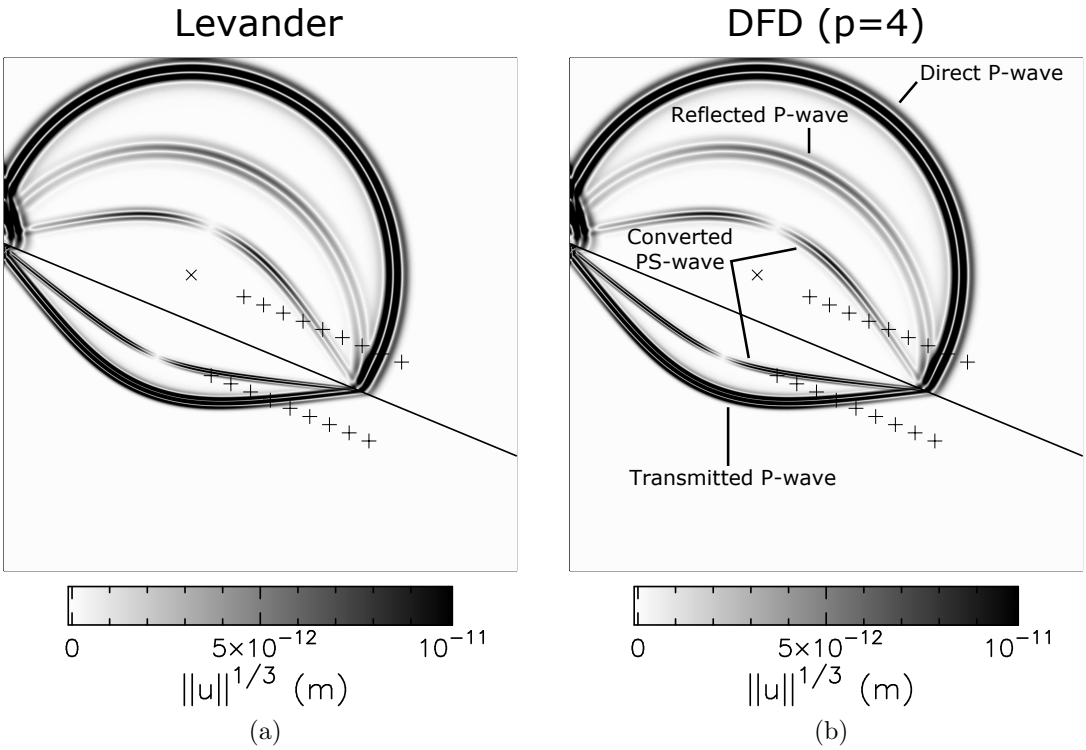
In Figure 8, we present variations in the error as a function of the distance to the source. As expected, the error increases with distance regardless of the algorithm employed. The growth of the error with distance appears less pronounced with the DFD algorithms than with the FD algorithm. At long distances, the error measured for the DFD algorithm with $p = 2$ is significantly smaller than the error observed for both FD schemes. The error measured for the DFD algorithm with $p = 4$ is more than one order of magnitude smaller than for the other three algorithms. Notice however that the accuracy of the FD simulation could be improved using a better algorithm for representing the free surface (see, e.g., Kristek et al., 2002).

In the right panel of Figure 8, we present the error measured in the second set of simulations, where we increase the number of grid points per minimum wavelength to minimize the numerical dispersion. At the maximum frequency $f_{max} \approx 2.5 f_0$, there are 8.7 and 5.0 grid points per wavelength for the P-wave and the S-wave, respectively. For the FD simulations the grid spacings are $\Delta x = \Delta y \approx 13.8$ m. As for the first set of simulations, the DFD simulations have the same dimension as the FD simulation. The time step $\Delta t = 4.6 \times 10^{-4}$ s is small enough to ignore the dispersion associated with time discretization. When increasing the minimum of points per wavelength, we observe that the error reduction is relatively small in the FD case and a lot more important in the DFD case. This example shows that the higher convergence rate of the DFD operators based on B-spline bases is preserved at the boundaries of the computational domain.

Wave propagation near a discontinuity

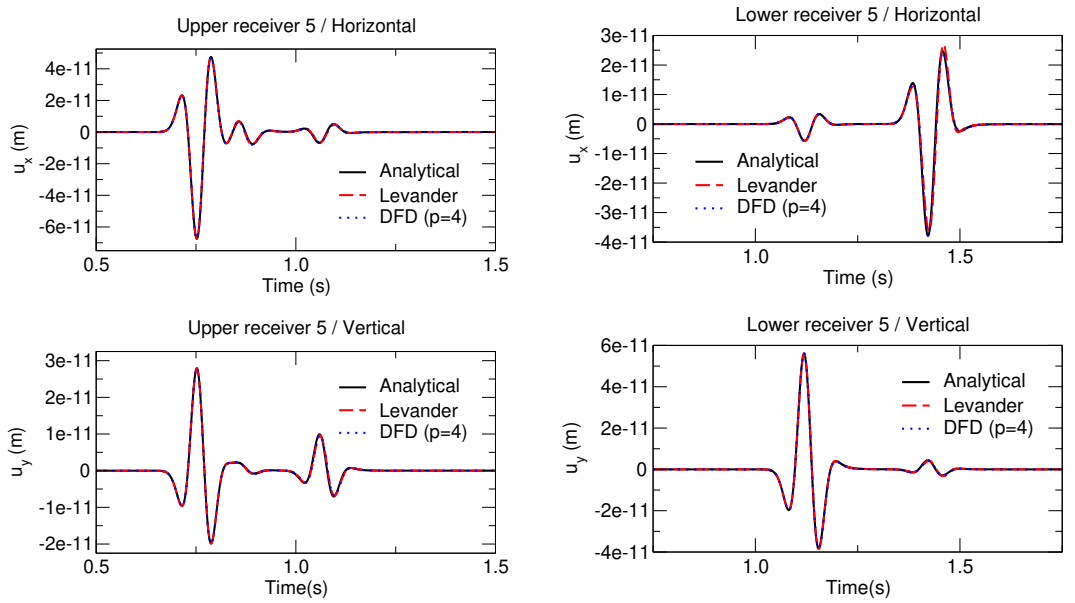
One attractive feature of FD modeling is that heterogeneity in the material properties can be simply and efficiently accounted for. This is not necessarily straightforward in other numerical methods, such as the finite element method or the spectral element method. In this section, we investigate the accuracy of the averaging procedure adapted from Moczo et al. (2014) which permits us to account for heterogeneity in the DFD modeling in the same way as in the FD modelling. For a quantitative analysis with analytical solutions, we consider wave propagation in a two-layered medium with an oblique interface, as illustrated in Figure 9.

For our comparison, we ran two simulations, one using the Levander algorithm and one using the DFD algorithm with $p = 4$ to minimize the numerical dispersion. In both simulations, the computational domain has dimension 6920 m $\times$ 6920 m. The material properties are homogeneous in the two layers. In the upper layer, the P-wave velocity is $v_p = 3000$ m/s, the S-wave velocity is $v_s = 1730$ m/s and the density is $\rho = 2500$ kg/m^3. In the lower layer, the P-wave velocity is $v_p = 1500$ m/s, the S-wave velocity is $v_s = 865$ m/s and the density is $\rho = 2500$ kg/m^3. The seismic source is placed at the center of the computational domain. The source time function is the



(a) (b)

Figure 9: Comparison of the displacement wavefields obtained when modeling wave propagation near an oblique discontinuity. The magnitude of the displacement vector is raised to power 1/3 to enhance small amplitudes. The principal waves are labeled in (b). A red cross marks the source location, and the receivers are represented using blue pluses. (a) shows the wavefield obtained using the Levander algorithm, (b) shows the wavefield obtained using the DFD algorithm with $p = 4$.



(a) (b)

Figure 10: Comparison of the seismograms obtained when modeling surface-wave propagation near a discontinuity. In (a), the recordings correspond to the mid-point receiver in the upper receiver line in Figure 9). In (b), the recordings correspond to the mid-point receiver in the lower receiver line in Figure 9).

Ricker wavelet in the equation 37 with peak frequency $f_0 = 10$ Hz and an estimated maximum frequency $f_{max} \approx 2.5 f_0 = 25$ Hz. The moment tensor is taken as

$$\begin{vmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}, \tag{48}$$

to mimic an explosive source. In the upper layer, there are 17.2 and 10 grid points per wavelength for the P-wave and the S-wave, respectively. In the lower layer, there are 8.7 and 5 grid points per wavelength for the P-wave and the S-wave, respectively. For the FD simulation the grid spacings are $\Delta x = \Delta y \approx 6.9$ m. In the DFD simulation, the grid dimensions are the same as in the FD simulations. The time step is $\Delta t = 2.3 \times 10^{-4}$ s. The displacement wavefield is recorded along two lines of receivers apart from the interface. The angle between the interface and the horizontal direction is $\pi/8$. The principal waves generated at the interface are annotated in Figure 9b.

When looking at the snapshots in Figure 9, we observe that the two wavefields match very well, and there is no visible numerical dispersion.

In Figure 10, we present the seismograms recorded at the two receivers located at the center of the receiver lines. We compare the numerical seismograms to the analytical solutions obtained using the software *gar6more* (Diaz and Ezziani, 2010). The upper receiver records the direct P-wave, the reflected P-wave, and the reflected PS-wave. The bottom receiver records the transmitted P-wave and the transmitted

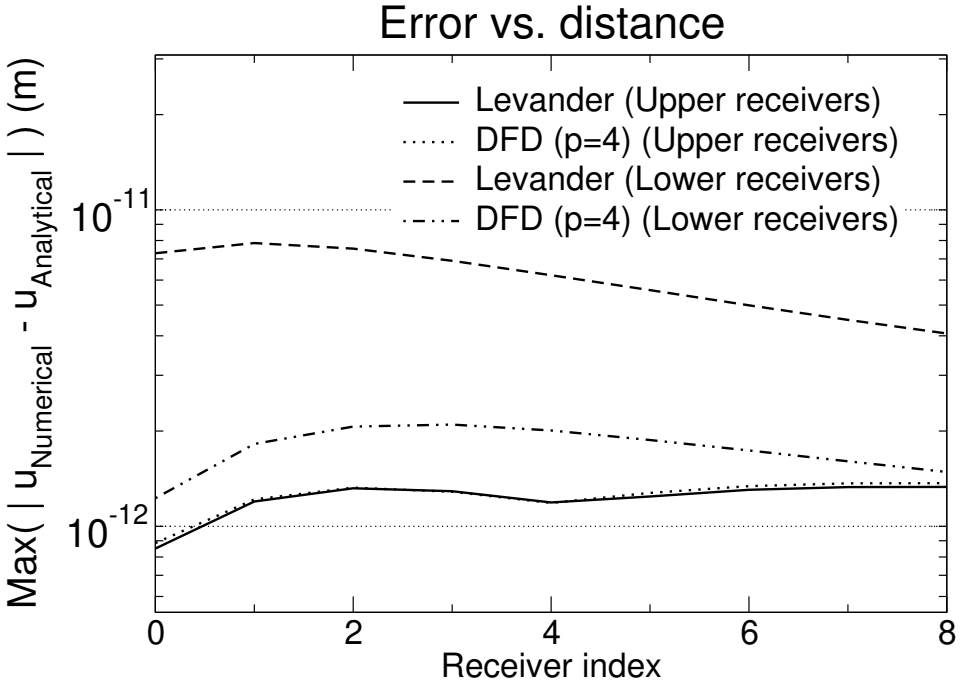


Figure 11: Comparison of the error measured when modeling wave propagation near a discontinuity. The error is taken as the maximum of the absolute value of the difference between the computed seismograms $\mathbf{u}_{\text{Numerical}}(t)$ and the analytical solutions $\mathbf{u}_{\text{Analytical}}(t)$. These results correspond to the simulations illustrated in Figure 9.

PS-wave. All seismograms show an excellent match with the analytical solution. The only noticeable difference is in the amplitude of the reflected PS wave, which is slightly higher for the Levander algorithm.

In Figure 11, we quantify the numerical error observed in the two simulations along the two receiver lines. Most of the error is attributed to a small-time shift between the computed seismograms and the analytical solution. Because a sufficient number of points per minimum wavelength is employed, this time shift is mostly due to the imperfect representation of the interface by the averaged properties in Equation 29 (see, e.g., Symes et al., 2009; Symes and Vdovina, 2009). Notice that the virtual grid points used for the DFD simulation in Equation 27 are different from those in Equation 4. Thus the smoothed interface is represented differently in the DFD and the FD simulations. Despite these slightly different representations, the observed error is almost the same in both simulations for the upper receivers. For the lower receivers, the error in the DFD simulation is much smaller than that of the FD simulation. Thus, the proposed approach for computing the effective properties in the DFD algorithm appears to be as accurate as what is done in FD modeling while considering significant jumps in material properties along the interface.

Let us consider another example of wave propagation in a model that is heterogeneous almost everywhere.

Wave propagation in a smooth random medium

In our last example, we consider wave propagation in a smoothly varying heterogeneous medium pictured in Figure 12c. We create the velocity distribution from a random map obtained by filtering a realization of the white noise with dimension 1000 points by 1000 points. A 2-point running average is applied in both directions (with periodic boundary conditions). It is passed 5000 times in the x direction and 1000 times in the y direction to obtain the random map. Thus, the correlation length is five times larger in the x direction than in the y direction for such a model.

We compare two simulations because no analytical solution is available. The first simulation is our reference simulation. It is an FD simulation using the Levander algorithm and a very fine grid to ensure accuracy. The second simulation is a DFD simulation with (p=4) and employs a coarse grid to verify the accuracy of the DFD algorithm.

In the two simulations, the computational domain has dimensions 6920 m by 6920 m. The P-wave velocity is varying between $v_p = 1500$ m/s and $v_p = 3000$ m/s. The S-wave velocity is varying between $v_s = 865$ m/s and $v_s = 1730$ m/s. The density $\rho = 2500$ kg/m^3 is constant as well as the ratio v_p/v_s. The seismic source is placed at the center of the domain. The moment tensor is the same as in the equation 46 and generates a P-wave and an S-wave having about the same amplitude at the receivers. The source time function is the Ricker wavelet in the equation 37 with peak frequency $f_0 = 10$ Hz and an estimated maximum frequency $f_{max} \approx 2.5 f_0 = 25$

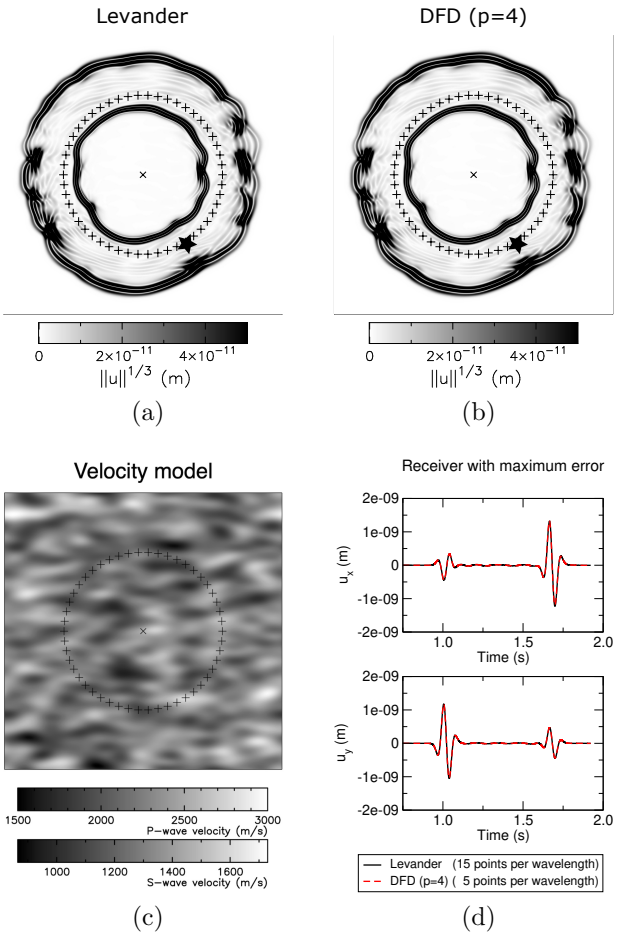


Figure 12: Comparison of the displacement wavefields obtained when modeling wave propagation in a smooth random medium. The velocity distribution is shown in (c). (a) shows the wavefield obtained using the Levander algorithm. (b) shows the wavefield obtained using the DFD algorithm with p= 4. A red cross marks the source location, and the 50 receivers are represented using blue pluses. (d) shows the seismograms recorded at the receiver marked with a blue star. It is the receiver where the measured difference (L^2-Norm) between the seismograms in simulations (a) and (b) is maximum.

Hz. The wavefield is recorded at 50 receivers placed at the same distance $d = 1977$ m to the source.

In the FD simulation, the grid spacings are $\Delta x = \Delta y \approx 4.6$ m. The time step is $\Delta t = 1.54 \times 10^{-4}$ s. At the maximum frequency $f_{max} = 2.5 f_0$, there is at least 15 grid points per S-wavelength (where $v_s = 865$ m/s). In the DFD simulation the grid spacings are $\Delta x = \Delta y \approx 13.8$ m. The time step is $\Delta t = 4.6 \times 10^{-4}$ s. At the maximum frequency $f_{max} = 2.5 f_0$, 5 grid points per S-wavelength.

When comparing the wavefields in Figure 12a and 12b, we observe a perfect match. Due to multipathing and focusing effects, the waves propagating in the x direction are very distorted. The waves propagating in the y direction are less affected and more coherent. This is a consequence of the anisotropic distribution of velocities.

In Figure 12d, we present seismograms recorded at the receiver for which the difference between the FD and the DFD simulations is the largest. We observe a remarkable agreement between the two sets of seismograms despite the smaller number of points per minimum wavelength employed in the DFD simulation. This confirms that our averaging strategy to assign the weights associated with the material property is accurate and adequate. Practically, with the introduction of the virtual DFD grid points, one can seamlessly substitute the FD operators with the DFD operators in the equation 2 and proceed as in FD modeling when accounting for the material properties.

With these two illustrations of numerical simulations in different heterogeneous models; we show that such a simple DFD approach with the averaging procedure may be considered for practical applications where material heterogeneity must be accounted for. Notice that our method makes it also possible to easily implement more accurate approaches (see, e.g., Capdeville et al., 2010; Kristek et al., 2017; Fichtner and Hanasoge, 2017).

CONCLUSION

We propose an essential update of the classical staggered-grid finite-difference algorithms, the backbone of many geophysical applications. The principal improvement is the substitution of the classic FD operators with the recently introduced DFD operators. This substitution preserves the geometrical structure of the algorithm at the discretization level. We show that the DFD operators reduce numerical dispersion and improve accuracy through numerical examples with quantitative comparisons when analytical solutions are available. Such an approach with a rather simple numerical algorithm will permit the development of wave-propagation-modeling software with enhanced efficiency and reduced memory usage.

The modified algorithm naturally accounts for the free surface boundary condition because the DFD operators are obtained through variational calculus. Therefore, we achieve an accuracy similar to that of the finite/spectral element method, for example,

when modeling surface wave propagation.

We propose a simple approach to account for heterogeneity in the velocity model, where the physical properties are directly assigned to virtual grid points associated with the basis functions. Such averaging approach at the grid level makes possible the use of standard FD procedures to discretize the material parameters. We verified the accuracy of the proposed approach.

On their own, the improvements above are important achievements. However, the structure of an FD algorithm combined with a variational treatment of the boundary conditions opens other perspectives. It should make it possible to take advantage of and recycle many algorithms developed in FD modeling, for example, the PML conditions that are difficult to adapt to variational methods.

In this work, we considered the DFD algorithm in its simplest form. Nonetheless, it is possible to account for anisotropic media and curvilinear coordinates and perform multi-domain modeling. Further, a unique FD scheme can be used to simulate wave propagation in solid and fluid domains and at solid/fluid interfaces where the numerical grid is allowed to be non-conformal. This is the object of a forthcoming publication

ACKNOWLEDGEMENT

The authors would like to thank Peter Moczo and two anonymous reviewers for their insightful and constructive comments that helped me improve the quality of this paper. We also thank Julien Diaz for providing his software. Yder Masson would like to thank the grant I-SITE E2S/UPPA/THERMapp and the France-Berkeley Fund (Project #10-2019) for partial support.

APPENDIX A

STAGGERED B-SPLINE BASES

The B-spline basis β^p with basis functions $\beta_i^p(x)$ $(i = 1, N)$ and polynomial order p is constructed from the knot vector $\mathbf{k}$ with $N + p + 1$ elements

$$k_i = \begin{cases} x_1 & \text{for } i = 1, p+1 \\ x_1 + \frac{i-p-1}{N-p}(x_2 - x_1) & \text{for } i = p+2, N \\ x_2 & \text{for } i = N+1, N+p+1 \end{cases} \tag{A-1}$$

The knot vector $\mathbf{k}$ starts and ends with $p + 1$ repeated knots and has equally spaced internal knots. From the knot vector, we can define the zero-order B-spline basis β^0 with basis functions

$$\beta_i^0(x) = \begin{cases} 1 & \text{if } k_i < x < k_{i+1} \\ 0 & \text{otherwise} \end{cases} . \tag{A-2}$$

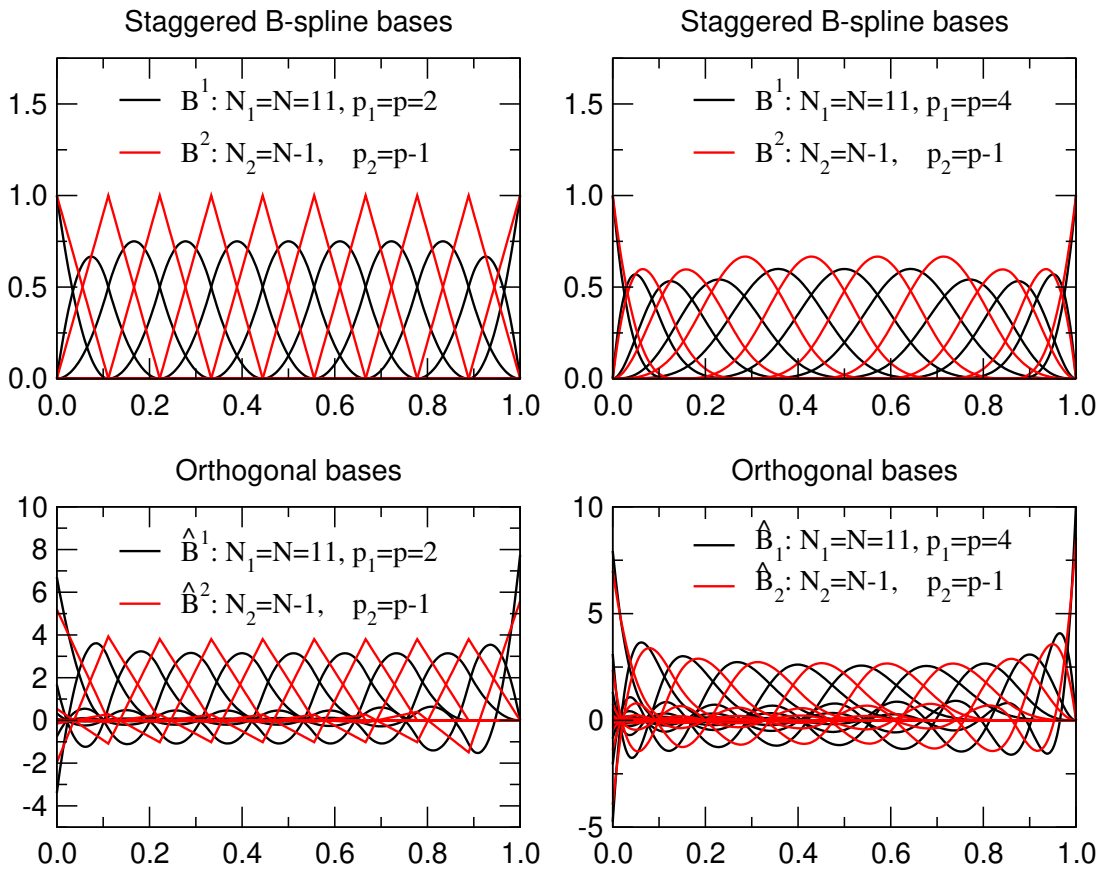


Figure A-1: Illustration of some staggered pairs of bases $B^1 = \beta^p / B^2 = \beta^{p-1}$(top panels) and their associated orthonormal bases $\hat{B}_1/\hat{B}_2$ (bottom panels), for $p = 1, 2$ (Left panels) and $p = 3, 4$ (right panels).

The basis functions of the B-spline basis β^p with polynomial order p can be constructed recursively using the Cox–de Boor formula

$$\beta_i^p(x) = \frac{x - k_i}{k_{i+p} - k_i}\beta_i^{p-1}(x) + \frac{k_{i+p+1} - x}{k_{i+p+1} - k_{i+1}}\beta_{i+1}^{p-1}(x). \tag{A-3}$$

In our application, we employ pairs B^1/B^2 of B-spline bases with staggered basis functions. The basis $B^1 = \beta^p$ has $N_1 = N$ basis function with polynomial order $p_1 = p$ and the basis $B^2 = \beta^{p-1}$ has $N_2 = N - 1$ basis functions with polynomial order $p_2 = p - 1$. From the equation A-1, we see that the knot vector $\mathbf{k}_2$ associated with B^2 is the same as the knot vector $\mathbf{k}_1$ associated with B^1 without its first and last elements. The pairs of staggered bases B^1/B^2 obtained for $p = 2, N = 11$ and $p = 4, N = 11$ are illustrated in the top panels of Figure A-1.

The derivatives of the basis functions β_i^p can be expressed as a function of the basis functions with reduced polynomial order β_i^{p-1} , we have

$$\frac{d\beta_i^p(x)}{dx} = \frac{p}{k_{i+p} - k_i}\beta_i^{p-1}(x) - \frac{p}{k_{i+p+1} - k_{i+1}}\beta_{i+1}^{p-1}(x). \tag{A-4}$$

It follows that the derivative $g'(x)$ of a function $g(x)$ represented in basis $B^1 = \beta^p$ with knot vector $\mathbf{k_1} = \mathbf{k} = [k_1, \ldots, k_{N+p+1}]$ as the linear combination

$$g(x) = \sum_{i=1}^{N} \mathrm{g}_i \beta_i^p(x) \tag{A-5}$$

with expansion coefficients g_i is given by

$$g'(x) = \sum_{i=1}^{N-1} \mathrm{g}'_i \beta_i^{p-1}(x) \tag{A-6}$$

where

$$\mathrm{g}'_i = \frac{p}{k_{i+p+1} - k_{i+1}}(\mathrm{g}_{i+1} - \mathrm{g}_i). \tag{A-7}$$

are the expansion coefficients of $g'(x)$ in basis $B^2 = \beta^{p-1}$ with knot vector $\mathbf{k_2} = [k_2, \ldots, k_{N+p}]$. In matrix form, we have

$$\mathbf{g}' = \mathbf{Q}\mathbf{g} \tag{A-8}$$

where the vectors $\mathbf{g}$ and $\mathbf{g}'$ contain the expansion coefficients g_i $(i = 1, N)$ and g'_i $(i = 1, N - 1)$, respectively. The matrix $\mathbf{Q}$ is a finite-difference like matrix with bandwidth 2 and dimension $(N - 1, N)$, we have

$$\mathbf{Q} = \begin{bmatrix} -\alpha_1 & +\alpha_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\alpha_{N-1} & +\alpha_{N-1} \end{bmatrix} \tag{A-9}$$

where

$$\alpha_i = \frac{p}{k_{i+p+1} - k_{i+1}}. \tag{A-10}$$

REFERENCES

Bécache, E., P. Joly, and C. Tsogka, 2001, A new family of mixed finite elements for the linear elastodynamic problem: SIAM Journal on Numerical Analysis, **39**, 2109–2132.

Cao, J., J.-B. Chen, and M.-X. Dai, 2018, An adaptive free-surface expression for three-dimensional finite-difference frequency-domain modelling of elastic wave: Geophysical Prospecting, **66**, 707–725.

Capdeville, Y., L. Guillot, and J.-J. Marigo, 2010, 2-d non-periodic homogenization to upscale elastic media for p-sv waves: Geophysical Journal International, **182**, 903–922.

De Hoop, A., 1970, The surface line source problem in elastodynamics: De Ingenieur, **82**.

Diaz, J., and A. Ezziani, 2010, Analytical solution for waves propagation in heterogeneous acoustic/porous media. part i: the 2d case: Communications in Computational Physics, **7**, 171–194.

Fichtner, A., and S. M. Hanasoge, 2017, Discrete wave equation upscaling: Geophysical Journal International, **209**, 353–357.

Geller, R. J., and N. Takeuchi, 1998, Optimally accurate second-order time-domain finite difference scheme for the elastic equation of motion: one-dimensional cases: Geophysical Journal International, **135**, 48–62.

Graves, R. W., 1996, Simulating seismic wave propagation in 3d elastic media using staggered-grid finite differences: Bulletin of the seismological society of America, **86**, 1091–1106.

Hicks, G. J., 2002, Arbitrary source and receiver positioning in finite-difference schemes using kaiser windowed sinc functions: Geophysics, **67**, 156–165.

Koene, E. F., J. O. Robertsson, F. Broggini, and F. Andersson, 2018, Eliminating time dispersion from seismic wave modeling: Geophysical Journal International, **213**, 169–180.

Komatitsch, D., C. Barnes, and J. Tromp, 2000a, Simulation of anisotropic wave propagation based upon a spectral element method: Geophysics, **65**, 1251–1260.

———, 2000b, Wave propagation near a fluid-solid interface: a spectral element approach: Geophysics, **65**, 623–631.

Kristek, J., P. Moczo, and R. J. Archuleta, 2002, Efficient methods to simulate planar free surface in the 3d 4th-order staggered-grid finite-difference schemes: Studia Geophysica et Geodaetica, **46**, 355–381.

Kristek, J., P. Moczo, E. Chaljub, and M. Kristekova, 2017, An orthorhombic representation of a heterogeneous medium for the finite-difference modelling of seismic wave propagation: Geophysical Journal International, **208**, 1250–1264.

Levander, A. R., 1988, Fourth-order finite-difference P-SV seismograms: Geophysics, **53**, 1425–1436.

Luo, Y., and G. T. Schuster, 1990, Parsimonious staggered grid finite-differencing of the wave equation: Geophysical Research Letters, **17**, 155–158.

Madariaga, R., 1976, Dynamics of an expanding circular fault: Bulletin of the Seismological Society of America, **66**, 639–666.

Masson, Y., 2022, Distributional finite-difference modelling of seismic waves: Geophysical Journal International, **233**, 264–296.

Mittet, R., 2002, Free-surface boundary conditions for elastic staggered-grid modeling schemes: Geophysics, **67**, 1616–1623.

Moczo, P., J. Kristek, and E. Bystrický, 2001, Efficiency and optimization of the 3-d finite-difference modeling of seismic ground motion: Journal of Computational Acoustics, **9**, 593–609.

Moczo, P., J. Kristek, and M. Gális, 2014, The finite-difference modelling of earthquake motions: Waves and ruptures: Cambridge University Press.

New, K. C., K. Watt, C. W. Misner, and J. M. Centrella, 1998, Stable 3-level leapfrog integration in numerical relativity: Physical Review D, **58**, 064022.

Robertsson, J. O. A., 1996, A numerical free-surface condition for elastic/viscoelastic finite-difference modeling in the presence of topography: Geophysics, **61**, 1921–1934.

Symes, W., I. S. Terentyev, and T. Vdovina, 2009, Getting it right without knowing the answer: Quality control in a large seismic modeling project: Presented at the 2009 SEG Annual Meeting, OnePetro.

Symes, W. W., and T. Vdovina, 2009, Interface error analysis for numerical wave propagation: Computational Geosciences, **13**, 363–371.

Tromp, J., D. Komatitsch, and Q. Liu, 2008, Spectral-Element and Adjoint Methods in Seismology: Communications in Computational Physics, **3**, 1–32.

Virieux, J., 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **51**, 889–901.

Virieux, J., H. Calandra, and R.-É. Plessix, 2011, A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging: Geophysical Prospecting, **59**, 794–813.

Wang, M., and S. Xu, 2015, Finite-difference time dispersion transforms for wave propagationtime dispersion transforms: Geophysics, **80**, WD19–WD25.

Yee, K. S., 1966, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media: IEEE Transactions on Antennas and Propagation, **14**, 302–307.

Zhang, W., and X. Chen, 2006, Traction image method for irregular free surface boundaries in finite difference seismic wave simulation: Geophysical Journal International, **167**, 337–353.