Distributional finite-difference modeling of seismic waves

Yder Masson

Geophysical Journal International

Geophys. J. Int. (2022) **00,** 1–33
Advance Access publication 2022 August 18
GJI Seismology

https://doi.org/10.1093/gji/ggac306

Distributional finite-difference modelling of seismic waves

Yder Masson [1,2]

[1] *Inria, Centre de recherche Bordeaux Sud-Ouest, Magique* 3D, 33405 *Talence, Aquitaine, France. E-mail: yder.masson@gmail.com*
[2] *Université de Pau et des Pays de l'Adour, E2S UPPA, CNRS, LFCR, UMR*5150, 64000 *Pau, France*

Accepted 2022 August 5. Received 2022 August 5; in original form 2021 December 1

SUMMARY

This study introduces a distributional finite-difference method (DFDM) for modelling the propagation of elastic waves in heterogeneous media in the time domain. DFDM decomposes the modelling domain into multiple elements that can have arbitrary sizes. When large elements are used, the proposed method closely resembles the finite-difference method because the wavefield is updated using operations involving band diagonal matrices only. Thus DFDM is computationally efficient. When smaller elements are used, DFDM looks closer to the finite-element or the spectral element methods and permits to mesh complicated structures. A complete multidomain algorithm for modelling elastic wave propagation in arbitrarily heterogeneous media is presented. The algorithm's stability is discussed, and the usual Courant condition governs the stability of the proposed scheme. Numerical examples show that the proposed algorithm accurately accounts for free surfaces, solid–fluid interfaces and accommodates non-conformal meshes in their basic form. Seismograms obtained using the proposed method are compared to those computed using analytical solutions and the spectral element method. To achieve comparable accuracy, DFDM requires fewer points per wavelength than the spectral element method, for example.

Key words: Numerical modelling; Computational seismology; Wave propagation.

1 INTRODUCTION

Numerical methods can accurately model seismic wave propagation in a heterogeneous 3-D Earth and have become indispensable in seismology and exploration geophysics. They play a crucial role in the investigation of the Earth's internal structure (Virieux & Operto 2009; French & Romanowicz 2014; Bozdağ *et al.* 2016). A wide variety of methodologies is used for modelling the propagation of elastic waves. The adequacy of a given method depends on the problem considered (Virieux *et al.* 2011).

The finite-difference method (FDM) is among the most popular methods because of its remarkable simplicity (Alterman & Karal 1968; Boore 1972; Alford *et al.* 1974; Kelly *et al.* 1976; Bayliss *et al.* 1986; Virieux 1986; Levander 1988; Igel *et al.* 1995; Zahradník 1995; Graves 1996; Appelö & Petersson 2009). FDM is extremely efficient for modelling seismic wave propagation but it is not straightforward to accurately manage heterogeneity, sharp interfaces and free surfaces with complex geometry. In the past decades, several steps have been taken to overcome this difficulty and improve the accuracy of FDM. Numerical schemes with high-order accuracy in space and time have been proposed (Geller & Takeuchi 1998; Takeuchi & Geller 2000; Liu 2013; Zhou *et al.* 2021; Chen *et al.* 2022). The implementation of the free surface with topography has been extensively studied (Lombard *et al.* 2008; Zhang *et al.* 2012a, b; de la Puente *et al.* 2014; Gao *et al.* 2015; Sun *et al.* 2016; Shragge & Konuk 2020; Zang *et al.* 2021). Accurate treatment of sharp internal material interfaces that do not coincide with the numerical grid (i.e. with subcell resolution) have been developed (Moczo *et al.* 2002, 2014, 2019; Mittet 2017). Wavefield excitation in the context of rupture propagation has been the focus of several studies (Zhang *et al.* 2014, 2016, 2020; Koene *et al.* 2020). Stable discontinuous grid has been obtained first by Kristek *et al.* (2010), it allows to refine the numerical grid and thus increase computational efficiency (Kostin *et al.* 2015; Li *et al.* 2015; Fan *et al.* 2015; Nie *et al.* 2017). Summation-by-parts (SBP) finite-difference operators that are stable, naturally account for the free surface and allow for multidomain modelling have gained more and more attention (Kreiss & Scherer 1974; Strand 1994; Olsson 1995; Mattsson & Nordström 2004; Carpenter *et al.* 2010; Albin & Klarmann 2016; Mattsson & Olsson 2018; Sun *et al.* 2020).

Pseudo-spectral methods (PSM) are regarded as a good alternative to FDM (Fornberg 1987; Carcione 1996). They have a very high spatial accuracy, permit multidomain modelling, and account for the free surface with topography.

The finite-element method (FEM) is very popular and usually comes to mind first when dealing with complex geometries because of its ability to account for arbitrarily complex structures (Lysmer & Drake 1972; Smith 1975; Marfurt 1984). In this case, the mesh generation is generally done separately (Lo 2014). Other methods that rely on an element-wise domain decomposition include spectral element methods

(SEM; Seriani & Priolo 1994; Komatitsch & Vilotte 1998; Komatitsch & Tromp 1999; Nissen-Meyer *et al.* 2014), boundary element methods (BEM; Bouchon & Coutant 1994; Papageorgiou & Pei 1998; Ba & Gao 2017), finite-volume methods (FVM; Dormy & Tarantola 1995; Dumbser *et al.* 2007) and Discontinuous Galerkin methods (DGM; de la Puente *et al.* 2008; Käser *et al.* 2008; Étienne *et al.* 2010; Warburton 2013; Bonnasse-Gahot *et al.* 2018). In the elements methods, the discrete algorithm is constructed from the wave equation's weak formulation, making it easier to account for various boundary conditions accurately, model discontinuities and free surfaces with complex shapes, and perform domain decomposition. One should note that SEM has enabled important advances in seismology (Tape *et al.* 2009; French & Romanowicz 2014; Bozdağ *et al.* 2016) and earthquake physics (Ampuero & Mao 2017; Weng & Ampuero 2019, 2020). In recent years, DGM and Arbitrary high-order DERivative DGM (ADER-DG) have become increasingly popular to model dynamic earthquake rupture (Wilcox *et al.* 2010; Breuer *et al.* 2014). This is because DGM do not impose any field continuity across their boundaries and allows include non-linear interface conditions. It is also possible to implement local time stepping (Uphoff *et al.* 2017) and to use adaptive mesh refinement (Burstedde *et al.* 2011).

In this work, a distributional finite-difference method (DFDM) is introduced. The principal motivation behind this approach is to obtain a method that leads to codes with a simple structure similar to FDM while taking advantage of the accurate treatment of the boundary conditions offered by SEM or DGM. In DFDM, the discrete operators evaluating the partial derivatives are obtained using the weighted residuals method (Salih 2016; Reddy 2019) which is in many ways similar to the weak-formulation approach; they can be seen as discrete equivalents of the distributional derivative

$$f^{(d)}(x) = \int_{-\infty}^{+\infty} f^{(d)}(l)\delta(l-x)\mathrm{d}l. \tag{1a}$$

$$= (-1)^d \int_{-\infty}^{+\infty} f(l)\delta^{(d)}(l-x)\mathrm{d}l, \tag{1b}$$

where the superscript (d) denotes the dth derivative and $\delta(x)$ is the Dirac delta function. To obtain an algorithmic structure similar to FDM, the 1-D operators in DFDM are constructed from basis functions with compact support. They are extended to higher dimensions using a tensor product. The DFDM algorithm is obtained directly from the strong form of the wave equation by replacing the partial derivatives with the discrete differential operators. Because the operators are constructed independently from the discretization of the wave equation in DFDM, one could use those to solve other types of partial differential equations (PDE).

DFDM permits an element-wise domain decomposition, it accurately accounts for the classic boundary conditions, free surface, material discontinuities (e.g. solid–fluid interfaces) and non-conformal interfaces. A significant difference with other element methods is that the elements or domains can be arbitrarily large. One or a few domains can be used when the medium has a simple geometry. Thus, using an external meshing tool is not necessary. However, it is possible whenever complex structures are present in the velocity model. The proposed algorithm has a structure similar to a classic finite-difference algorithm within large elements. This is because the DFD operators depend on basis functions having compact support. In this case, the computations involve operations on band diagonal matrices, this gives the method its efficiency.

An important characteristic of DFDM is that the discrete wavefield is discontinuous between neighbouring elements, as in DGM. This offers more freedom when specifying boundary conditions.

Because DFDM shares the structure of FDM we can envision recycling many existing algorithms. For example, the PMLs that are difficult to adapt to variational methods or the generalized Maxwell body (GMB-EK; Emmerich & Korn 1987; Moczo & Kristek 2005) and the generalized Zener body (Carcione *et al.* 1988a, b) used to model attenuation. This is perhaps the most appealing feature of DFDM.

The manuscript is organized as follows. In Section 2, the distributional finite-difference (DFD) operators are first introduced in their most general form, and the treatment of boundary conditions is discussed. It is shown that adjoint pairs of DFD operators can be constructed by using specific basis functions; this is important to obtain stable numerical schemes. Then, the DFD operators are extended to higher dimensions. Finally, an approximate averaging scheme that allows accounting for heterogeneity in the elastic properties is presented. In Section 3, the DFD algorithm used for modelling wave propagation in the time domain is detailed. The 1-D case is considered first, and the stability of the numerical scheme is discussed. It is shown that the proposed numerical scheme is stable under the classic Courant condition (CFL) when the adjoint DFD operators are used. Then, a complete 2-D algorithm that allows to model wave propagation in complex heterogeneous media is detailed. In Section 4, numerical examples demonstrating the accuracy of the proposed method are presented. The DFDM is benchmarked against analytical solutions, and the results are compared to those obtained with the spectral element method.

2 THE DFD OPERATORS

2.1 DFD operators in one dimension

This section introduces formulae for computing an approximation of the dth derivative $f^{(d)}(x')$ of a known function $f(x')$ that depends on some spatial coordinate x'. Let the function $f(x')$ be defined on domain $\Omega : [x'_-, x'_+]$ and represented in a given basis B as the linear combination

$$f(x') = \sum_{j=1}^{N_B} \mathbf{f}_{B_j} B_j(x'), \tag{2}$$

where $B_j(x')$ $(j = 1, N_B)$ denote the N_B basis functions with support inside Ω and $\mathbf{f}_{\boldsymbol{B}_j}$ $(j = 1, N_B)$ are the expansion coefficients stored in the vertical vector $\mathbf{f}_{\boldsymbol{B}}$. The dth derivative of $f(x')$ is given by

$$f^{(d)}(x') = \sum_{j=1}^{N_B} \mathbf{f}_{\boldsymbol{B}_j} B_j^{(d)}(x'), \tag{3}$$

where $B_j^{(d)}(x')$ is the derivative of order d of the basis function $B_j(x')$. Let the desired approximation $\tilde{f}^{(d)}(x')$ of the exact derivative $f^{(d)}(x')$ be represented in an arbitrary basis β as the linear combination

$$\tilde{f}^{(d)}(x') = \sum_{j=1}^{N_\beta} \mathbf{f}_{\boldsymbol{\beta}_j}^{(d)} \beta_j(x'), \tag{4}$$

where $\beta_j(x')$ denote the N_β basis functions with support inside Ω and $\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)}$ are the expansion coefficients stored in the vertical vector $\mathbf{f}_{\boldsymbol{\beta}}^{(d)}$. To obtain $\tilde{f}^{(d)}(x')$, one needs to determine the expansion coefficients in the vector $\mathbf{f}_{\boldsymbol{\beta}}^{(d)}$. When the approximation is equal to the true derivative, we have

$$\tilde{f}^{(d)}(x') = f^{(d)}(x') \quad \forall\, x' \in \Omega. \tag{5}$$

Assuming that the expansion coefficients $\mathbf{f}_{\boldsymbol{B}_j}$ and $\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)}$ in eqs (3) and (4) are known, it is not straightforward to check that eq. (5) is satisfied because the two functions are represented in different bases. One way to test the equality in eq. (5) is to measure $f^{(d)}(x')$ and $\tilde{f}^{(d)}(x')$ by multiplying both sides of eq. (5) by the same arbitrary test function ϕ_i and integrating over the domain Ω. This gives

$$\int_{x'_-}^{x'_+} \phi_i \tilde{f}^{(d)} \mathrm{d}x' = \int_{x'_-}^{x'_+} \phi_i f^{(d)} \mathrm{d}x' \tag{6a}$$

$$= (-1)^d \int_{x'_-}^{x'_+} \phi_i^{(d)} f \,\mathrm{d}x' + \sum_{p=0}^{d-1} (-1)^{d-p-1} \left[\phi_i^{(d-p-1)} f^{(p)} \right]_{x'_-}^{x'_+}, \tag{6b}$$

where the right-hand side of eq. (6b) is obtained by repeated integration by part, assuming that ϕ and f are continuously differentiable. When the test function ϕ_i is thought as a Dirac delta function $\delta(x' - x'_0)$ with the fundamental property $\int \delta(x' - x'_0) f(x') \mathrm{d}x' = f(x'_0)$, eq. (6) simply verifies that $\tilde{f}^{(d)}(x')$ and $f^{(d)}(x')$ have the same value at point x'_0. If this is true for any point x'_0 in $\Omega : [x'_-, x'_+]$, then eq. (5) is satisfied. Thus, a strategy for computing the approximate derivative $\tilde{f}^{(d)}(x')$ is to search for the optimal expansion coefficients $\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)}$ that satisfy the constraint in eq. (6) for a set of N_ϕ test functions $\phi_i(x')$ $(i = 1, N_\phi)$. Because $\tilde{f}^{(d)}(x')$ is represented using a finite number of unknown coefficients $\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)}$, one only needs to use a finite number of test functions to determine these coefficients. This general approach is called the method of weighted residuals. By substituting $f^{(d)}(x')$ and $\tilde{f}^{(d)}(x')$ in eq. (6) with their expansions in eqs (3) and (4), respectively, we obtain

$$\sum_{j=1}^{N_\beta} \mathbf{f}_{\boldsymbol{\beta}_j}^{(d)} \int_{x'_-}^{x'_+} \phi_i \beta_j \mathrm{d}x' = \sum_{j=1}^{N_B} \mathbf{f}_{\boldsymbol{B}_j} \int_{x'_-}^{x'_+} \phi_i B_j^{(d)} \mathrm{d}x' \tag{7a}$$

$$= (-1)^d \sum_{j=1}^{N_B} \mathbf{f}_{\boldsymbol{B}_j} \int_{x'_-}^{x'_+} \phi_i^{(d)} B_j \mathrm{d}x' + \sum_{p=0}^{d-1} (-1)^{d-p-1} \sum_{j=1}^{N_B} \mathbf{f}_{\boldsymbol{B}_j} \left[\phi_i^{(d-p-1)} B_j^{(p)} \right]_{x'_-}^{x'_+} \tag{7b}$$

or, in matrix form

$$\mathbf{M}\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)} = \mathbf{K}^{(d)} \mathbf{f}_{\boldsymbol{B}} = \left[\mathbf{K}^{*(d)} + \mathbf{Q}^{-(d)} + \mathbf{Q}^{+(d)} \right] \mathbf{f}_{\boldsymbol{B}}, \tag{8}$$

where

$$\mathbf{M} = \mathbf{M}_{\phi,\beta} = (\mathbf{M})_{i,j} = \langle \phi_i, \beta_j \rangle_{x'_-}^{x'_+} \tag{9a}$$

$$\mathbf{K}^{(d)} = \mathbf{M}_{\phi,B}^{(d)} = \left(\mathbf{K}^{(d)}\right)_{i,j} = \langle \phi_i, B_j^{(d)} \rangle_{x'_-}^{x'_+} \tag{9b}$$

$$\mathbf{K}^{*(d)} = \mathbf{K}_{\phi,B}^{*(d)} = \left(\mathbf{K}^{*(d)}\right)_{i,j} = (-1)^d \langle \phi_i^{(d)}, B_j \rangle_{x'_-}^{x'_+} \tag{9c}$$

$$\mathbf{Q}^{\pm(d)} = \mathbf{Q}_{\phi,B}^{\pm(d)} = \left(\mathbf{Q}^{\pm(d)}\right)_{i,j} = \pm \sum_{p=0}^{d-1} (-1)^{d-p-1} \mathbf{\Phi}^{\pm(d-p-1)} \left(\mathbf{q}^{\pm(p)}\right)^T = \pm \sum_{p=0}^{d-1} (-1)^{d-p-1} \phi_i^{(d-p-1)}(x'_\pm) B_j^{(p)}(x'_\pm) \tag{9d}$$

$$\mathbf{\Phi}^{\pm(p)} = \mathbf{\Phi}_{\phi}^{\pm(p)} = \left(\mathbf{\Phi}^{\pm(p)}\right)_i = \phi_i^{(p)}(x'_\pm) \tag{9e}$$

$$\mathbf{q}^{\pm(d)} = \mathbf{q}_{B}^{\pm(d)} = \left(\mathbf{q}^{\pm(d)}\right)_i = B_i^{(d)}(x'_\pm) \tag{9f}$$

$$\mathbf{K}^{(d)} = \mathbf{K}^{*(d)} + \mathbf{Q}^{-(d)} + \mathbf{Q}^{+(d)}, \tag{9g}$$

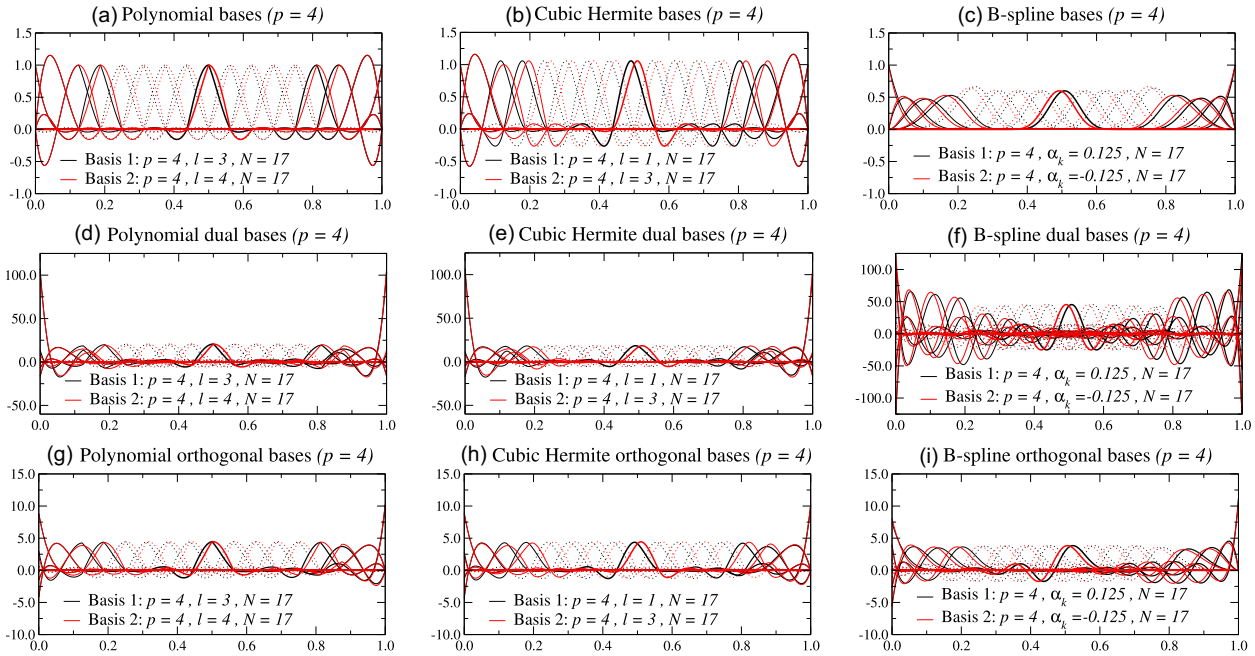

Figure 1. (a) The fourth order polynomial bases obtained following the approach in Section A1 (see Fig. A1a for details). (b) The fourth order Hermite bases obtained following the approach in Section A2 (see Fig. A1b for details). (c) The fourth order B-spline bases obtained following the approach in Section A3(c) (see Fig. A1c for details). (d) The dual bases obtained from the polynomial bases in panel (a) using eq. (25). (e) The dual bases obtained from the Hermite bases in (b) using eq. (25). (f) The dual bases obtained from the B-spline bases in panel (c) using eq. (25). (g) The dual bases obtained from the polynomial bases in panel (a) using eq. (20). (h) The dual bases obtained from the Hermite bases in panel (b) using eq. (20). (i) The dual bases obtained from the B-spline bases in panel (c) using eq. (20).

and

$$\langle f, g\rangle_{x'_-}^{x'_+} = \int_{x'_-}^{x'_+} f(x')g(x')\mathrm{d}x'. \tag{10}$$

In eqs (8) and (9), the matrix $\mathbf{M}$ is often referred to as the mass matrix, the matrices $\mathbf{K}^{(d)}$ and $\mathbf{K}^{*(d)}$ involving the derivatives of the basis functions are usually called stiffness matrices and the matrices $\mathbf{Q}^{-(d)}$ and $\mathbf{Q}^{+(d)}$ account for the left and the right boundary values, respectively. The linear system in eq. (8) can be solved to obtain the expansion coefficient vector

$$\mathbf{f}_{\boldsymbol{\beta}}^{(d)} = \boldsymbol{D}_{\phi,\beta,B}^{(d)}\mathbf{f}_{\boldsymbol{B}}, \tag{11}$$

where the operator

$$\boldsymbol{D}_{\phi,\beta,B}^{(d)} = \mathbf{M}^{-1}\mathbf{K}^{(d)} = \mathbf{M}^{-1}\left[\mathbf{K}^{*(d)} + \mathbf{Q}^{-(d)} + \mathbf{Q}^{+(d)}\right]. \tag{12}$$

is the DFD operator of order d. Provided that the inverse matrix $\mathbf{M}^{-1}$ (or some generalized inverse of the matrix $\mathbf{M}$) exists, the DFD operator in eq. (12) can be used to compute approximations of the dth derivative of $f(x')$. The DFD operator $\boldsymbol{D}_{\phi,\beta,B}^{(d)}$ acts on the expansion vector $\mathbf{f}_{\boldsymbol{B}}$ representing $f(x')$ in basis B and returns the expansion vector $\mathbf{f}_{\boldsymbol{\beta}}^{(d)}$ representing $\tilde{f}^{(d)}(x')$ in basis β.

In this study, the DFD operator $\boldsymbol{D}_{\phi,\beta,B}^{(d)}$ will be constructed using the piecewise polynomial bases presented in the Appendix and pictured in Figs 1(a)–(c) and (A1). All the basis functions considered have compact support (i.e. equal to zero outside a finite interval). It is crucial to obtain efficient algorithms with a finite-difference-like structure. Indeed, when those bases are used, the matrices $\mathbf{K}^{(d)}$ and $\mathbf{K}^{*(d)}$ are band diagonal (i.e. similar to finite-difference matrices but with different coefficients). The matrix $\mathbf{M}$ is band diagonal and symmetric. It can be factorized using Cholesky factorization. Thus the multiplications by the matrices $\mathbf{K}^{(d)}$, $\mathbf{K}^{*(d)}$, and $\mathbf{M}$ can be implemented very efficiently. They have a comparable computational cost. The elements of the vectors $\boldsymbol{\Phi}^{\pm(p)}$ and $\mathbf{q}^{\pm(d)}$ are zero except the first one or the last one. The elements of the matrices $\mathbf{Q}^{\pm(d)}$ are zero except the first one (upper-left) or the last one (bottom-right). It simplifies the treatment of the boundary conditions. Prior ending this section, one should make two informative remarks:

(i) First, when the basis functions $B_j(x')$ and $\beta_j(x')$ are interpolant, that is, when the expansion coefficients are equal to the values of the functions at some interpolation or collocation points, that is we have $\mathbf{f}_{\boldsymbol{B}_j} = f(x'_i)(x'_i = 1, N_B)$ and $\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)} = \tilde{f}^{(d)}(x'_j)$ $(x'_j = 1, N_\beta)$, then the operator in eq. (12) is a true finite-difference operator.

(ii) Secondly, when the test functions ϕ_i are taken as the dual basis functions $\delta_i(x') = \sum_{j=1}^{N_\beta}(\mathbf{M}_\beta^{-1})_{ij}\beta_j(x')$ (e.g. in Fig. 1), where the $(\mathbf{M}_\beta^{-1})_{ij}$ are the components of the inverse of the matrix $\mathbf{M}_\beta = (\mathbf{M}_\beta)_{i,j} = \langle\beta_i, \beta_j\rangle_{x'_-}^{x'_+}$, the expansion coefficients $\mathbf{f}_{\boldsymbol{\beta}_j}^{(d)}$ in eq. (4) can be obtained directly

using $\mathbf{f}^{(d)}_{\beta_j} = \int_{x'_-}^{x'_+} \delta_j f^{(d)} \mathrm{d}x'$. Further, when the basis functions β_i are also interpolant, we have

$$\mathbf{f}^{(d)}_{\beta_j} = f^{(d)}(x'_j) = \int_{x'_-}^{x'_+} \delta_j f^{(d)} \mathrm{d}x', \tag{13}$$

which is a discrete analogue to Dirac's formula in eq. (1b) where the Dirac delta function $\delta(x')$ has been replaced with its approximation $\delta_i(x')$ defined above.

These two observations outline the relationship between the DFD operators and classic finite-difference operators. In the next section, the DFD operator $\boldsymbol{D}^{(d)}_{\phi,\beta,B}$ is modified to account for prescribed boundary conditions.

2.2 DFD operators with prescribed boundary values

In this section, the DFD operator in eq. (12) is reformulated in a way that is more convenient to impose prescribed boundary conditions and to perform domain decomposition. At the limits x'_- and x'_+ of domain Ω, we may express the boundary values of $f^{(d)}(x')$ using the weighted averages

$$f^{(d)}(x'_-) = (1-\gamma_-)\, f^{(d)}(x'_-) + \gamma_-\, \mathrm{f}^{-(d)} \tag{14a}$$

$$f^{(d)}(x'_+) = (1-\gamma_+)\, f^{(d)}(x'_+) + \gamma_+\, \mathrm{f}^{+(d)}, \tag{14b}$$

where γ_- and γ_+ are the left and right weights, respectively, $f^{(d)}(x'_-)$ and $f^{(d)}(x'_+)$ are internal values taken inside domain Ω that can be computed using eq. (2), and, $\mathrm{f}^{-(d)}$ and $\mathrm{f}^{+(d)}$ are external values taken from the domains $\Omega_- : [\ldots, x'_-]$ and $\Omega_+ : [x'_+, \ldots]$ neighbouring domain Ω to the left and to the right, respectively. Assuming that the function $f(x')$ has representations similar to that in eq. (2) in domains Ω_- and Ω_+, the external values may be computed using

$$\mathrm{f}^{-(d)} = \left| f^{(d)}(x'_+) \right|_{\Omega_-} = \left| \sum_{j=1}^{N_B} \mathbf{f}_{\boldsymbol{B}_j}\, B_j(x'_+) \right|_{\Omega_-} = \left| \left(\mathbf{q}^{+(d)}\right)^T \mathbf{f}_{\boldsymbol{B}} \right|_{\Omega_-} \tag{15a}$$

$$\mathrm{f}^{+(d)} = \left| f^{(d)}(x'_-) \right|_{\Omega_+} = \left| \sum_{j=1}^{N_B} \mathbf{f}_{\boldsymbol{B}_j}\, B_j(x'_-) \right|_{\Omega_+} = \left| \left(\mathbf{q}^{-(d)}\right)^T \mathbf{f}_{\boldsymbol{B}} \right|_{\Omega_+}, \tag{15b}$$

where $|\ldots|_\Omega$ specifies that the quantities within the brackets are associated with domain Ω, in particular, $\left|x'_-\right|_\Omega = \left|x'_+\right|_{\Omega_-}$ and $\left|x'_+\right|_\Omega = \left|x'_-\right|_{\Omega_+}$. Starting back from eq. (6) and following the steps thereafter using the substitutions in eq. (14) shows that the expansion coefficients in eq. (11) can be computed using the modified expression

$$\mathbf{f}^{(d)}_{\boldsymbol{\beta}} = \mathbf{D}^{(d)}_{\phi,\beta,B,\gamma_-,\gamma_+} \mathbf{f}_{\boldsymbol{B}} + \sum_{p=0}^{d-1} \left[\mathbf{b}^{-(p)} \mathrm{f}^{-(p)} + \mathbf{b}^{+(p)} \mathrm{f}^{+(p)} \right], \tag{16}$$

where the original DFD operator $\boldsymbol{D}^{(d)}_{\phi,\beta,B}$ in eq. (12) transforms to

$$\mathbf{D}^{(d)}_{\phi,\beta,B,\gamma_-,\gamma_+} = \mathbf{M}^{-1} \left[\mathbf{K}^{*(d)} + (1-\gamma_-)\, \mathbf{Q}^{-(d)} + (1-\gamma_+)\, \mathbf{Q}^{+(d)} \right] = \mathbf{M}^{-1} \left[\mathbf{K}^{(d)} - \gamma_- \mathbf{Q}^{-(d)} - \gamma_+ \mathbf{Q}^{+(d)} \right] \tag{17}$$

and the boundary operators are

$$\mathbf{b}^{\pm(p)} = \mathbf{b}^{\pm(p)}_{\phi,\beta,\gamma_\pm} = \pm\gamma_\pm (-1)^{d-p-1} \mathbf{M}^{-1}\, \boldsymbol{\Phi}^{\pm(d-p-1)}. \tag{18}$$

When setting $\gamma_- = \gamma_+ = 0$ in eqs (17) and (18), eq. (16) reduces to eq. (11) and the DFD operator in eq. (17) is the same as the one in eq. (12). In this case, the boundary values $f^{(d)}(x'_-)$ and $f^{(d)}(x'_+)$ are implicitly specified by the expansion coefficients in vector $\mathbf{f}_{\boldsymbol{B}}$. When taking $\gamma_- = 1$ and/or $\gamma_+ = 1$, the DFD operator in eq. (17) returns the expansion coefficients vector $\mathbf{f}^{(d)}_{\boldsymbol{\beta}}$ assuming that $f^{(d)}(x'_-) = 0$ and/or $f^{(d)}(x'_+) = 0$, respectively. The left and right boundary values can be imposed independently by adding the products $\mathbf{b}^{\pm(p)} \mathrm{f}^{\pm(p)}$ to the vector $\mathbf{f}^{(d)}_{\boldsymbol{\beta}}$. The external values $\mathrm{f}^{-(d)}$ and $\mathrm{f}^{+(d)}$ may be set arbitrarily or computed using eq. (14) knowing the expansion coefficients vectors $\left|\mathbf{f}_{\boldsymbol{B}}\right|_{\Omega_-}$ and $\left|\mathbf{f}_{\boldsymbol{B}}\right|_{\Omega_+}$ in the neighbouring domains Ω_- and Ω_+, respectively.

2.3 Adjoint DFD operators

In this section, we construct adjoint pairs of DFD operators acting on functions represented in the first basis $\hat{B}_1$ and returning their derivative represented in the second basis $\hat{B}_2$, and inversely. Such pairs of DFD operators allow to repeatedly differentiate the field variables (e.g. velocity, displacement or stresses, etc.) represented in different bases, going back and forth from one variable to another in a consistent manner. Further, they maintain the self-adjoint nature of the wave equation when it is discretized, which is essential to achieve numerical reciprocity and obtain

numerical schemes that are conditionally stable (see Section 2.1). This is because adjoint operators lead to a block-antidiagonal first-order linear system that preserves the intrinsic staggered property (Virieux 1986) using dedicated basis function.

Adjoint DFD operators (i.e. where the second operator is the transpose of the first one or its opposite) can be obtained directly from the general expressions in eqs (17) and (18) by using specific basis functions for ϕ, β and B. Let $f(x')$ and $g(x')$ be two functions represented in the bases B_{1_i} $(i = 1, N_1)$ and B_{2_i} $(i = 1, N_2)$ through their expansion coefficient vectors $\mathbf{f_1}$ and $\mathbf{g_2}$, respectively. By constructing and factorizing (e.g. using Cholesky factorization) the mass matrices associated with the bases B_1 and B_2 as

$$\mathbf{M_{11}} = \mathbf{M}_{B_1,B_1} = \mathbf{L_1}\mathbf{L_1}^T \tag{19a}$$

$$\mathbf{M_{22}} = \mathbf{M}_{B_2,B_2} = \mathbf{L_2}\mathbf{L_2}^T, \tag{19b}$$

we can construct the two sets of orthogonal basis functions (illustrated in Fig. 1)

$$\hat{B}_{1_i} = \sum_{j=1}^{N_1} (\mathbf{L_1^{-1}})_{ij} B_{1_j} \quad \forall\ i = 1, N_1 \tag{20a}$$

$$\hat{B}_{2_i} = \sum_{j=1}^{N_2} (\mathbf{L_2^{-1}})_{ij} B_{2_j} \quad \forall\ i = 1, N_2. \tag{20b}$$

By representing the functions $f(x')$ and $g(x')$ in the orthogonal bases $\hat{B}_1$ and $\hat{B}_2$, respectively, we can construct a pair $\mathbf{\hat{D}_{21}^{(d)}}$ / $\mathbf{\hat{D}_{12}^{(d)}}$ of DFD operators and their attached boundary operators $\mathbf{\hat{b}_2^{\pm(p)}}$/$\mathbf{\hat{b}_1^{\pm(p)}}$ such that: $\mathbf{\hat{D}_{21}^{(d)}}$ acts on the vector $\mathbf{\hat{f}_1}$ representing $f(x')$ in basis $\hat{B}_1$ and returns the vector $\mathbf{\hat{f}_2^{(d)}}$ representing the dth derivative of $f(x')$ in basis $\hat{B}_2$, and, $\mathbf{\hat{D}_{12}^{(d)}}$ acts on the vector $\mathbf{\hat{g}_2}$ representing $g(x')$ in basis $\hat{B}_2$ and returns the vector $\mathbf{\hat{g}_1^{(d)}}$ representing the dth derivative of $g(x')$ in basis $\hat{B}_1$. We have:

$$\mathbf{\hat{f}_2^{(d)}} = \mathbf{\hat{D}_{21}^{(d)}}\mathbf{\hat{f}_1} + \sum_{p=0}^{d-1}\left[\mathbf{\hat{b}_2^{-(p)}}\mathrm{f}^{-(p)} + \mathbf{\hat{b}_2^{+(p)}}\mathrm{f}^{+(p)}\right] \tag{21a}$$

$$\mathbf{\hat{g}_1^{(d)}} = \mathbf{\hat{D}_{12}^{(d)}}\mathbf{\hat{g}_2} + \sum_{p=0}^{d-1}\left[\mathbf{\hat{b}_1^{-(p)}}\mathrm{g}^{-(p)} + \mathbf{\hat{b}_1^{+(p)}}\mathrm{g}^{+(p)}\right]. \tag{21b}$$

If needed, the expansion vectors in the original bases B_1 and B_2 may be obtained using

$$\mathbf{f_1} = (\mathbf{L_1^{-1}})^T\mathbf{\hat{f}_1} \quad \mathbf{f_2^{(d)}} = (\mathbf{L_2^{-1}})^T\mathbf{\hat{f}_2^{(d)}} \quad \mathbf{g_2} = (\mathbf{L_2^{-1}})^T\mathbf{\hat{g}_2} \quad \mathbf{g_1^{(d)}} = (\mathbf{L_1^{-1}})^T\mathbf{\hat{g}_1^{(d)}}. \tag{22}$$

To enforce the adjoint property $\mathbf{\hat{D}_{21}^{(d)}} = \pm(\mathbf{\hat{D}_{12}^{(d)}})^T$, the operators $\mathbf{\hat{D}_{21}^{(d)}}$ and $\mathbf{\hat{b}_2^{\pm(p)}}$ are constructed from eqs (17) and (18) using the bases substitutions $B = \hat{B}_1, \phi = \beta = \hat{B}_2$ with weights $\gamma_- = \alpha_-, \gamma_+ = \alpha_+$, and, the operators $\mathbf{\hat{D}_{12}^{(d)}}$ and $\mathbf{\hat{b}_1^{\pm(p)}}$ are constructed from eqs (17) and (18) using the substitutions $B = \hat{B}_2, \phi = \beta = \hat{B}_1$ with complementary weights $\gamma_- = (1 - \alpha_-), \gamma_+ = (1 - \alpha_+)$.

$$\mathbf{\hat{D}_{21}^{(d)}} = \mathbf{D}^{(d)}_{\hat{B}_2,\hat{B}_2,\hat{B}_1,\alpha_-,\alpha_+} = \mathbf{L_2^{-1}}\overset{*}{\mathbf{D}}^{(d)} \quad (\mathbf{L_1^{-1}})^T \tag{23a}$$

$$\mathbf{\hat{D}_{12}^{(d)}} = \mathbf{D}^{(d)}_{\hat{B}_1,\hat{B}_1,\hat{B}_2,(1-\alpha_-),(1-\alpha_+)} = (-1)^d \quad \mathbf{L_1^{-1}}\left(\overset{*}{\mathbf{D}}^{(d)}\right)^T(\mathbf{L_2^{-1}})^T = (-1)^d(\mathbf{\hat{D}_{21}^{(d)}})^T \tag{23b}$$

$$\mathbf{\hat{b}_2^{\pm(p)}} = \mathbf{b}^{\pm(p)}_{\hat{B}_2,\hat{B}_2,\alpha_\pm} = \mathbf{L_2^{-1}}\overset{*}{\mathbf{b}}{}_2^{\pm(p)} \tag{23c}$$

$$\mathbf{\hat{b}_1^{\pm(p)}} = \mathbf{b}^{\pm(p)}_{\hat{B}_1,\hat{B}_1,(1-\alpha_\pm)} = \mathbf{L_1^{-1}}\overset{*}{\mathbf{b}}{}_1^{\pm(p)}, \tag{23d}$$

where

$$\overset{*}{\mathbf{D}}^{(d)} = \overset{*}{\mathbf{D}}{}^{(d)}_{B_2,\overset{*}{B}_2,B_1,\alpha_-,\alpha_+} = \left[\mathbf{K_{21}^{(d)}} - \alpha_-\mathbf{Q_{21}^{-(d)}} - \alpha_+\mathbf{Q_{21}^{+(d)}}\right] \tag{24a}$$

$$= (-1)^d\overset{*}{\mathbf{D}}{}^{(d)}_{B_1,\overset{*}{B}_1,B_2,(1-\alpha_-),(1-\alpha_+)} = (-1)^d\left[\mathbf{K_{12}^{(d)}} - (1-\alpha_-)\mathbf{Q_{12}^{-(d)}} - (1-\alpha_+)\mathbf{Q_{12}^{+(d)}}\right]^T. \tag{24b}$$

$$\overset{*}{\mathbf{b}}{}_2^{\pm(p)} = \overset{*}{\mathbf{b}}{}^{\pm(p)}_{B_2,\overset{*}{B}_2,\alpha_\pm} = \pm\alpha_\pm(-1)^{d-p-1}\mathbf{q_2^{\pm(d-p-1)}} \tag{24c}$$

$$\overset{*}{\mathbf{b}}{}_1^{\pm(p)} = \overset{*}{\mathbf{b}}{}^{\pm(p)}_{B_1,\overset{*}{B}_1,(1-\alpha_\pm)} = \pm(1-\alpha_\pm)(-1)^{d-p-1}\mathbf{q_1^{\pm(d-p-1)}}. \tag{24d}$$

and the matrices $\mathbf{K_{21}^{(d)}} = \mathbf{K}^{(d)}_{B_2,B_1}$, $\mathbf{Q_{21}^{\pm(d)}} = \mathbf{Q}^{\pm(d)}_{B_2,B_1}$, $\mathbf{K_{12}^{(d)}} = \mathbf{K}^{(d)}_{B_1,B_2}$, $\mathbf{Q_{12}^{\pm(d)}} = \mathbf{Q}^{\pm(d)}_{B_1,B_2}$, $\mathbf{q_1^{\pm(d)}} = \mathbf{q}^{\pm(d)}_{B_1}$ and $\mathbf{q_2^{\pm(d)}} = \mathbf{q}^{\pm(d)}_{B_2}$ are computed from the original bases B_1 and B_2 using eq. (9). The operators $\overset{*}{\mathbf{D}}{}^{(d)}$, $\overset{*}{\mathbf{b}}{}_1^{\pm}$ and $\overset{*}{\mathbf{b}}{}_2^{\pm}$ in eq. (24) are themselves DFD operators returning the expansion coefficients $\overset{*}{\mathbf{f}}{}^{(d)}_{\hat{B}_2} = \mathbf{M_{22}}\mathbf{\hat{f}_2^{(d)}}$ and $\overset{*}{\mathbf{g}}{}^{(d)}_{\hat{B}_1} = \mathbf{M_{11}}\mathbf{\hat{g}_1^{(d)}}$ of the dth derivative of $f(x')$ and $g(x')$ in the dual bases (illustrated in Fig. 1)

$$\overset{*}{B}_{1_i} = \sum_{j=1}^{N_1}(\mathbf{M_{11}^{-1}})_{ij}B_{1_j} \quad \forall\ i = 1, N_1 \tag{25a}$$

$$\overset{*}{B}_{2_i} = \sum_{j=1}^{N_2}(\mathbf{M_{22}^{-1}})_{ij}B_{2_j} \quad \forall\ i = 1, N_2, \tag{25b}$$

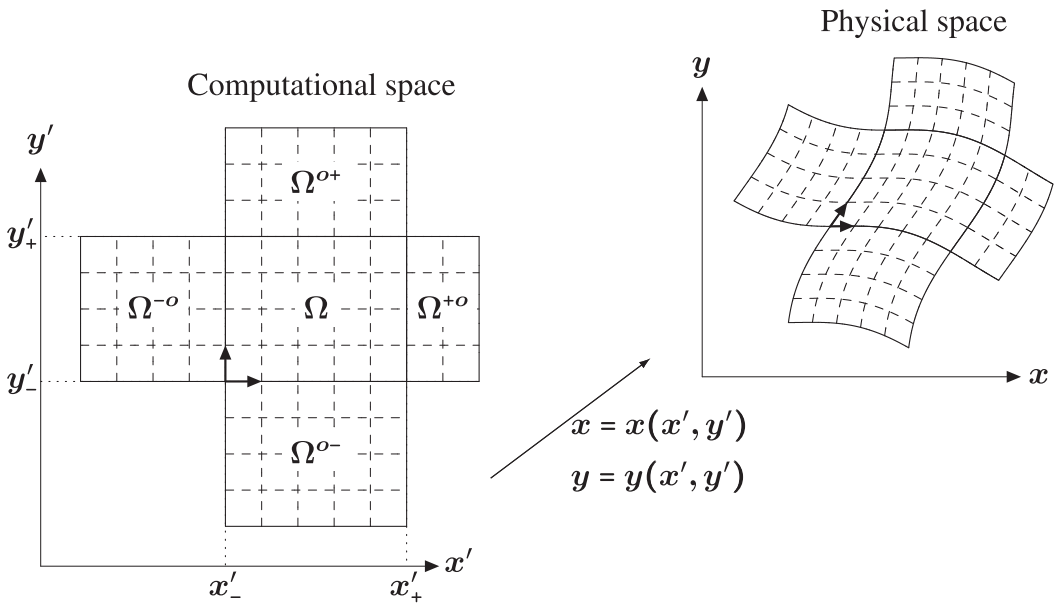

Figure 2. Illustration of the geometrical mapping between the physical space coordinates (x, y) and the computational space coordinates $(x^{'}, y^{'})$.

respectively. Note that choosing the complementary boundary weights $\gamma_- = \alpha_-$, $\gamma_+ = \alpha_+$ for $\hat{\mathbf{D}}_{\mathbf{21}}^{(d)}$ and $\gamma_- = (1 - \alpha_-)$, $\gamma_+ = (1 - \alpha_+)$ for $\hat{\mathbf{D}}_{\mathbf{12}}^{(d)}$ enforces the property $\overset{*}{\mathbf{D}}{}^{(d)}_{B_2,B_2,B_1} = (-1)^d \overset{*}{\mathbf{D}}{}^{(d)}_{B_1,B_1,B_2}$ in eqs (24a) and (24b). We shall see that when modelling wave propagation in a single domain, the condition $\hat{\mathbf{D}}_{\mathbf{12}}^{(d)} = (-1)^d (\hat{\mathbf{D}}_{\mathbf{21}}^{(d)})^T$ is sufficient to obtain stable numerical schemes (see Section 2.1), Neumann or Dirichlet boundary conditions can be accounted for by setting $\alpha_\pm = 0$ or $\alpha_\pm = 1$, while the boundary values $\mathrm{f}^{\pm(p)}$ and $\mathrm{g}^{\pm(p)}$ may be zeroed out or used to impose an external source. When modelling wave propagation in multiple domains Ω_i $(i = 1, N_d)$, the additional constrain $|\alpha_+|_{\Omega_i} = (1 - |\alpha_-|_{\Omega_{i+1}})$ which translates the continuity condition $|f^{(d)}(x'_+)|_{\Omega_i} = |f^{(d)}(x'_-)|_{\Omega_{i+1}}$, is needed to ensure that the global DFD operator (that evaluates the derivatives in all domains) stays adjoint). The operators in eq. (23) are the basic bricks of the proposed algorithm, they can be combined to compute partial derivative in higher dimensions as discussed in Section 2.4.

2.4 DFD operators in higher dimensions

In this section the DFD operators obtained in the previous sections are extended to higher dimensions. A general formula for the DFD operators in two dimension is presented, the operators in higher dimensions can be derived in a similar manner. The objective is to compute the derivatives of a 2-D function $f(x^{'}, y^{'})$ in the rectangular domain (e.g. as pictured in Fig. 2)

$$\Omega := (x', y') : x'_- \leq x' \leq x'_+, \quad y'_- \leq y' \leq y'_+. \tag{26}$$

Given the 1-D sets of basis functions $X_{B_i}(x')$ $(i = 1, N_{X_B})$, $X_{\beta_i}(x')$ $(i = 1, N_{X_\beta})$, $X_{\phi_i}(x')$ $(i = 1, N_{X_\phi})$ defined on the 1-D domain $\Omega_x : [x'_-, x'_+]$ and, $Y_{B_i}(y')$ $(i = 1, N_{Y_B})$, $Y_{\beta_i}(y')$ $(i = 1, N_{Y_\beta})$, $Y_{\phi_i}(y')$ $(i = 1, N_{Y_\phi})$ defined on the 1-D domain $\Omega_y : [y'_-, y'_+]$, we can construct the 2-D sets of basis functions

$$B_{i,j}(x', y') = X_{B_i}(x') Y_{B_j}(y') \qquad \begin{cases} 1 \leq i \leq N_{X_B} \\ 1 \leq j \leq N_{Y_B} \end{cases} \tag{27a}$$

$$\beta_{i,j}(x', y') = X_{\beta_i}(x') Y_{\beta_j}(y') \qquad \begin{cases} 1 \leq i \leq N_{X_\beta} \\ 1 \leq j \leq N_{Y_\beta} \end{cases} \tag{27b}$$

$$\phi_{i,j}(x', y') = X_{\phi_i}(x') Y_{\phi_j}(y') \qquad \begin{cases} 1 \leq i \leq N_{X_\phi} \\ 1 \leq j \leq N_{Y_\phi} \end{cases} . \tag{27c}$$

Following the approach in Section (2.1), we represent the function

$$f(x', y') = \sum_{i=1}^{N_{X_B}} \sum_{j=1}^{N_{Y_B}} \mathbf{F}_{\boldsymbol{B}_{ij}} B_{ij}(x', y') = \sum_{i=1}^{N_{X_B}} X_{B_i}(x') \sum_{j=1}^{N_{Y_B}} Y_{B_j}(y') \mathbf{F}_{\boldsymbol{B}_{ij}} \tag{28}$$

using $N_{X_B} \times N_{Y_B}$ basis functions $B_{ij}(x^{'}, y^{'})$ with support inside the rectangular domain in eq. (26) and expansion coefficients $\mathbf{F}_{\boldsymbol{B}_{ij}}$ stored in matrix $\mathbf{F}_{\boldsymbol{B}}$. The exact partial derivatives of $f(x^{'}, y^{'})$ are

$$f^{(d_x,d_y)}(x', y') = \sum_{i=1}^{N_{X_B}} \sum_{j=1}^{N_{Y_B}} \mathbf{F}_{\boldsymbol{B}_{ij}} B_{ij}^{(d_x,d_y)}(x', y') = \sum_{i=1}^{N_{X_B}} X_{B_i}^{(d_x)}(x') \sum_{j=1}^{N_{Y_B}} Y_{B_j}^{(d_y)}(y') \, \mathbf{F}_{\boldsymbol{B}_{ij}}, \tag{29}$$

where $B_{ij}^{(d_x,d_y)}(x', y') = \frac{\partial^{d_x+d_y}}{\partial_x{}^{d_x} \partial_y{}^{d_y}} B_{ij}(x', y') = \frac{\partial^{d_x}}{\partial_x{}^{d_x}} X_{B_i}(x') \frac{\partial^{d_y}}{\partial_y{}^{d_y}} Y_{B_j}(y') = X_{B_i}^{(d_x)}(x') Y_{B_j}^{(d_y)}(y')$ denotes the partial derivative of the basis function $B_{ij}(x^{'}, y^{'})$, and, d_x and d_y are the orders of the partial derivatives in direction x and y, respectively. We want to obtain the approximation

$$\tilde{f}^{(d_x,d_y)}(x', y') = \sum_{i=1}^{N_{X_\beta}} \sum_{j=1}^{N_{Y_\beta}} \tilde{\mathbf{f}}_{\boldsymbol{\beta}_{ij}}^{(d_x,d_y)} \beta_{ij}(x', y') = \sum_{i=1}^{N_{X_\beta}} X_{\beta_i}(x') \sum_{j=1}^{N_{Y_\beta}} Y_{\beta_j}(y') \, \tilde{\mathbf{f}}_{\boldsymbol{\beta}_{ij}}^{(d_x,d_y)} \tag{30}$$

of $f^{(d_x,d_y)}(x', y')$ represented using $N_{X_\beta} \times N_{Y_\beta}$ basis functions $\beta_{ij}(x^{'}, y^{'})$ with support inside the rectangular domain in eq. (26) and unknown expansion coefficients $\tilde{\mathbf{f}}_{\boldsymbol{\beta}_{ij}}^{(d_x,d_y)}$ stored in matrix $\tilde{\mathbf{f}}_{\boldsymbol{\beta}}^{(d_x,d_y)}$. Starting with the target equality

$$\tilde{f}^{(d_x,d_y)}(x', y') = f^{(d_x,d_y)}(x', y'), \tag{31}$$

multiplying both sides of eq. (31) by the test function $\phi_{kl} = \phi_{kl}^{(0,0)}$ in eq. (27c) and integrating over the rectangular domain in eq. (26)

$$\int_{x'_-}^{x'_+} \int_{y'_-}^{y'_+} \phi_{kl} \tilde{f}^{(d_x,d_y)} \mathrm{d}x' \mathrm{d}y' = \int_{x'_-}^{x'_+} \int_{y'_-}^{y'_+} \phi_{kl} f^{(d_x,d_y)} \mathrm{d}x' \mathrm{d}y', \tag{32}$$

and replacing $f^{(d_x,d_y)}$ and $\tilde{f}^{(d_x,d_y)}$ with their expansions in eqs (29) and (30), respectively, we obtain

$$\sum_{i=1}^{N_{X_\beta}} \int_{x'_-}^{x'_+} X_{\phi_k} X_{\beta_i} \mathrm{d}x' \sum_{j=1}^{N_{Y_\beta}} \int_{y'_-}^{y'_+} Y_{\phi_l} Y_{\beta_j} \mathrm{d}y' \tilde{\mathbf{f}}_{\boldsymbol{\beta}_{ij}}^{(d_x,d_y)} = \sum_{i=1}^{N_{X_B}} \int_{x'_-}^{x'_+} X_{\phi_k} X_{B_i}^{(d_x)} \mathrm{d}x' \sum_{j=1}^{N_{Y_B}} \int_{y'_-}^{y'_+} Y_{\phi_l} Y_{B_j}^{(d_y)} \mathrm{d}y' \mathbf{F}_{\boldsymbol{B}_{ij}}, \tag{33}$$

or, in matrix form

$$\mathbf{M}_{\mathbf{x}} \tilde{\mathbf{f}}_{\boldsymbol{\beta}}^{(d_x,d_y)} (\mathbf{M}_{\mathbf{y}})^T = \mathbf{K}_{\mathbf{x}}^{(d_x)} \mathbf{F}_{\boldsymbol{B}} (\mathbf{K}_{\mathbf{y}}^{(d_y)})^T, \tag{34}$$

where the matrices $\mathbf{M}_{\mathbf{x}} = \mathbf{M}_{X_\phi, X_\beta}$, $\mathbf{M}_{\mathbf{y}} = \mathbf{M}_{Y_\phi, Y_\beta}$, $\mathbf{K}_{\mathbf{x}}^{(d_x)} = \mathbf{K}_{X_\phi, X_B}^{(d_x)}$ and $\mathbf{K}_{\mathbf{y}}^{(d_y)} = \mathbf{K}_{Y_\phi, Y_B}^{(d_y)}$ are similar to those in the 1-D case and can be computed using eqs (9a) and (9b). The expansion coefficients $\tilde{\mathbf{F}}_{\beta_{ij}}$ can be computed by solving the linear system in eq. (34), and the 2-D DFD operator $\boldsymbol{D}_{\phi,\beta,B}^{(d_x,d_y)}$ can be defined using

$$\tilde{\mathbf{f}}_{\boldsymbol{\beta}}^{(d_x,d_y)} = \boldsymbol{D}_{\phi,\beta,B}^{(d_x,d_y)} \mathbf{F}_{\boldsymbol{B}}. \tag{35}$$

By defining the action of the tensor product between two matrices $\mathbf{A}$ and $\mathbf{B}$ on a matrix $\mathbf{X}$ as

$$(\mathbf{A} \otimes \mathbf{B}) \mathbf{X} = \mathbf{A} \mathbf{X} \mathbf{B}^T = \left(\mathbf{B} (\mathbf{A} \mathbf{X})^T \right)^T, \tag{36}$$

where $\otimes$ denotes the tensor product, and, using the identity

$$(\mathbf{A} \otimes \mathbf{B}) = (\mathbf{A} \otimes \mathbf{I}) (\mathbf{I} \otimes \mathbf{B}) = (\mathbf{I} \otimes \mathbf{B}) (\mathbf{A} \otimes \mathbf{I}), \tag{37}$$

where $\mathbf{I}$ is the identity matrix, we can express the 2-D DFD operator in eq. (35) as

$$\boldsymbol{D}_{\phi,\beta,B}^{(d_x,d_y)} = \left(\mathbf{M}_{\mathbf{x}}{}^{-1} \mathbf{K}_{\mathbf{x}}^{(d_x)} \right) \otimes \left(\mathbf{M}_{\mathbf{y}}{}^{-1} \mathbf{K}_{\mathbf{y}}^{(d_y)} \right) \tag{38a}$$

$$= \boldsymbol{D}_{X_\phi, X_\beta X_B}^{(d_x)} \otimes \boldsymbol{D}_{Y_\phi, Y_\beta, Y_B}^{(d_y)} \tag{38b}$$

$$= \left(\boldsymbol{D}_{X_\phi, X_\beta X_B}^{(d_x)} \otimes \mathbf{I} \right) \left(\mathbf{I} \otimes \boldsymbol{D}_{Y_\phi, Y_\beta, Y_B}^{(d_y)} \right) \tag{38c}$$

$$= \left(\mathbf{I} \otimes \boldsymbol{D}_{Y_\phi, Y_\beta, Y_B}^{(d_y)} \right) \left(\boldsymbol{D}_{X_\phi, X_\beta X_B}^{(d_x)} \otimes \mathbf{I} \right) \tag{38d}$$

which is simply the tensor product between the 1-D DFD operators $\boldsymbol{D}_{X_\phi, X_\beta X_B}^{(d_x)}$ and $\boldsymbol{D}_{Y_\phi, Y_\beta, Y_B}^{(d_y)}$. When acting on a matrix $\mathbf{X}$ the operator $(\mathbf{A} \otimes \mathbf{I})$ in eq. (37) applies the matrix $\mathbf{A}$ to all columns of matrix $\mathbf{X}$ while the operator $(\mathbf{I} \otimes \mathbf{B})$ applies the matrix $\mathbf{B}$ to all rows of matrix $\mathbf{X}$. Thus the 2-D DFD operators simply apply the 1-D DFD operators $\boldsymbol{D}_{Y_\phi, Y_\beta, Y_B}^{(d_y)}$ and $\boldsymbol{D}_{X_\phi, X_\beta X_B}^{(d_x)}$ to all the rows and to all the columns of the matrix $\mathbf{F}_{\boldsymbol{B}}$, respectively. When isolating the boundary values as in Section 2.2, in place of eq. (35), we obtain the following expression for computing the partial derivatives of $f(x^{'}, y^{'})$

$$\mathbf{F}^{(d_x,d_y)} = \sum_{p_x=0}^{d_x-1}\sum_{p_y=0}^{d_y-1}(\mathbf{b}_{\mathbf{x}}^{-(p_x)} \otimes \mathbf{b}_{\mathbf{y}}^{-(p_y)})\mathrm{f}^{--(p_x,p_y)} + \sum_{p_x=0}^{d_x-1}(\mathbf{b}_{\mathbf{x}}^{-(p_x)} \otimes \mathbf{D}_{\mathbf{y}}^{-o(d_y)})\mathbf{f}^{-o(p_x,0)} + \sum_{p_x=0}^{d_x-1}\sum_{p_y=0}^{d_y-1}(\mathbf{b}_{\mathbf{x}}^{-(p_x)} \otimes \mathbf{b}_{\mathbf{y}}^{+(p_y)})\mathrm{f}^{-+(p_x,p_y)} \tag{39a}$$

$$+ \sum_{p_y=0}^{d_y-1}(\mathbf{D}_{\mathbf{x}}^{o-(d_x)} \otimes \mathbf{b}_{\mathbf{y}}^{-(p_y)})\mathbf{f}^{o-(0,p_y)} + (\mathbf{D}_{\mathbf{x}}^{oo(d_x)} \otimes \mathbf{D}_{\mathbf{y}}^{oo(d_y)})\mathbf{F}^{oo} + \sum_{p_y=0}^{d_y-1}(\mathbf{D}_{\mathbf{x}}^{o+(d_x)} \otimes \mathbf{b}_{\mathbf{y}}^{+(p_y)})\mathbf{f}^{o+(p_y,0)} \tag{39b}$$

$$+ \sum_{p_x=0}^{d_x-1}\sum_{p_y=0}^{d_y-1}(\mathbf{b}_{\mathbf{x}}^{+(p_x)} \otimes \mathbf{b}_{\mathbf{y}}^{-(p_y)})\mathrm{f}^{+-(p_x,p_y)} + \sum_{p_x=0}^{d_x-1}(\mathbf{b}_{\mathbf{x}}^{+(p_x)} \otimes \mathbf{D}_{\mathbf{y}}^{+o(d_y)})\mathbf{f}^{+o(0,p_x)} + \sum_{p_x=0}^{d_x-1}\sum_{p_y=0}^{d_y-1}(\mathbf{b}_{\mathbf{x}}^{+(p_x)} \otimes \mathbf{b}_{\mathbf{y}}^{+(p_y)})\mathrm{f}^{++(p_x,p_y)}, \tag{39c}$$

where $\mathbf{F}^{oo} = \mathbf{F}_{\mathcal{B}}$ and the superscripts $(--, -o, -+, o-, oo, o+, +-, +o, ++)$ identify the domains from which the values are computed, as illustrated in Fig. 2. The 1-D operators involved in the computations are

$$\mathbf{b}_{\mathbf{x}}^{\pm(p_x)} = \mathbf{b}_{X_\phi,X_\beta,\gamma_\pm^x}^{\pm(p_x)} \tag{40a}$$

$$\mathbf{b}_{\mathbf{y}}^{\pm(p_y)} = \mathbf{b}_{Y_\phi,Y_\beta,\gamma_\pm^y}^{\pm(p_y)} \tag{40b}$$

$$\mathbf{D}_{\mathbf{x}}^{ab(d_x)} = \mathbf{D}_{X_\phi,X_\beta,X_B^{ab},\gamma_-^x,\gamma_+^x}^{\pm(p_x)} \tag{40c}$$

$$\mathbf{D}_{\mathbf{y}}^{ab(d_y)} = \mathbf{D}_{Y_\phi,Y_\beta,Y_B^{ab},\gamma_-^y,\gamma_+^y}^{\pm(p_y)}. \tag{40d}$$

The external values of $f^{(d_x,d_y)}(x', y')$ at the bottom-left $(--)$, bottom-right $(+-)$, top-left $(-+)$ and top-right $(++)$ corners of domain Ω are defined as

$$f^{(d_x,d_y)}(x'_-,y'_-) = \mathrm{f}^{--(d_x,d_y)} \qquad f^{(d_x,d_y)}(x'_-,y'_+) = \mathrm{f}^{-+(d_x,d_y)} \tag{41a}$$

$$f^{(d_x,d_y)}(x'_+,y'_-) = \mathrm{f}^{+-(d_x,d_y)} \qquad f^{(d_x,d_y)}(x'_+,y'_+) = \mathrm{f}^{++(d_x,d_y)}. \tag{41b}$$

The values of $f^{(d_x,d_y)}(x', y')$ along the left $(-o)$, right $(+o)$, bottom $(o-)$ and top $(o+)$ faces of domain Ω represented in the 1-D bases Y^{-o}, Y^{+o}, X^{o-} and X^{o+}, respectively, are represented as

$$f^{(d_x,0)}(x'_-,y') = \sum_{j=1}^{N_Y^{-o}} \mathbf{f}_j^{-o(d_x,0)} Y_j^{-o}(y') \qquad f^{(d_x,0)}(x'_+,y') = \sum_{j=1}^{N_Y^{+o}} \mathbf{f}_j^{+o(d_x,0)} Y_j^{+o}(y') \tag{42a}$$

$$f^{(0,d_y)}(x',y'_-) = \sum_{j=1}^{N_X^{o-}} \mathbf{f}_j^{o-(0,d_y)} X_j^{o-}(x') \qquad f^{(0,d_x)}(x',y'_+) = \sum_{j=1}^{N_X^{o+}} \mathbf{f}_j^{o+(0,d_x)} X_j^{o+}(x'). \tag{42b}$$

Note that the bases representing the external values do not need to be the same as those representing $f(x', y')$ inside domain Ω which is useful for non-conformal multidomain modelling. In higher dimension the DFD operators can be obtained directly by taking the tensor product of the 1-D operators, for example, in three dimension, we have

$$\boldsymbol{D}_{\phi,\beta,B}^{(d_x,d_y,d_z)} = \boldsymbol{D}_{X_\phi,X_\beta X_B}^{(d_x)} \otimes \boldsymbol{D}_{Y_\phi,Y_\beta,Y_B}^{(d_y)} \otimes \boldsymbol{D}_{Z_\phi,Z_\beta Z_B}^{(d_z)}. \tag{43}$$

where the 1-D operators $\boldsymbol{D}_{A,B,C}^{(d)}$ are given by eq. (12).

2.5 Accounting for material heterogeneity using weighted DFD operators

This section introduces an approximation to account for heterogeneity in the medium when modelling wave propagation without modifying the DFD operators introduced in the previous sections. To solve the wave equation numerically in heterogeneous media, we need to compute products of the form

$$p(x') = h(x')f(x') \tag{44}$$

between some spatially varying physical property $h(x')$ (e.g. density, elastic moduli, etc.) and some field variable $f(x')$ (e.g. velocities, displacements, stresses, etc.). Let $p(x')$, $h(x')$ and $f(x')$ be represented as the linear combinations

$$f(x') = \sum_{j=1}^{N_{\hat{B}}} \hat{\mathbf{f}}_{\hat{\boldsymbol{B}}_j} \hat{B}_j(x') \tag{45a}$$

$$p(x') = \sum_{j=1}^{N_{\hat{B}}} \hat{\mathbf{p}}_{\hat{\boldsymbol{B}}_j} \hat{B}_j(x') \tag{45b}$$

$$h(x') = \sum_{j=1}^{N_H} \mathbf{h}_j H_j(x'), \tag{45c}$$

where $p(x')$ and $f(x')$ are represented in the same orthogonal basis $\hat{B}$ (i.e. as defined in eq. 20) with expansion coefficients $\hat{\mathbf{p}}_{\hat{B}_j}$ and $\hat{\mathbf{f}}_{\hat{B}_j}$ stored in vectors $\hat{\mathbf{p}}_{\hat{B}}$ and $\hat{\mathbf{f}}_{\hat{B}}$, respectively. $h(x')$ is represented in an arbitrary basis H with expansion coefficients $\mathbf{h}_j$ stored in vector $\mathbf{h}$. Our objective is to determine the expansion coefficient vector $\hat{\mathbf{p}}_{\hat{B}}$ that gives an accurate approximation of the product in eq. (44). To do so, we proceed as we did for deriving the DFD operators. By multiplying both sides of eq. (44) by the same test function $\phi_i(x') = \hat{B}_i(x')$ $(i = 1, N_{\hat{B}})$ and integrating over the domain of interest Ω we obtain the linear system

$$\mathbf{M}\,\hat{\mathbf{p}}_{\hat{B}} = \mathbf{I}\,\hat{\mathbf{p}}_{\hat{B}} = \hat{\mathbf{p}}_{\hat{B}} = \mathbf{M}_{\mathrm{h}}\,\mathbf{f}_{\hat{B}}, \tag{46}$$

where

$$\mathbf{M} = \mathbf{I} = \mathrm{M}_{i,j} = \langle \hat{B}_i, \hat{B}_j \rangle_{x'_-}^{x'_+} = \delta_{ij} \tag{47a}$$

$$\mathbf{M}_h = (\mathrm{M}_h)_{i,j} = \langle h\hat{B}_i, \hat{B}_j \rangle_{x'_-}^{x'_+} = \langle \hat{B}_i, h\hat{B}_j \rangle_{x'_-}^{x'_+}. \tag{47b}$$

Note that an obvious advantage when working with the orthogonal basis in eq. (20) is that the mass matrix in eq. (47a) is the identity matrix. In this case, the expansion coefficient vector $\mathbf{p}_{\hat{B}}$ is obtained by multiplying the discrete field variable $\mathbf{f}_{\hat{B}}$ by the weighted mass matrix $\mathbf{M}_h$. This matrix may be pre-computed and stored prior to running a simulation. Though this is not a problem in 1-D, storing the matrix $\mathbf{M}_h$ requires lots of memory, and it dramatically increases the computational burden in 2-D and 3-D. To overcome this issue, one can substitute the mass matrix $\mathbf{M_h}$ by the diagonal matrix $\overline{\mathbf{M}}_\mathbf{h}$ obtained by summing all line coefficients of $\mathbf{M_h}$ onto the diagonal. We have:

$$\overline{\mathbf{M}}_h = \mathrm{diag}\left(\langle\langle \mathbf{h} \rangle\rangle\right), \tag{48}$$

where the vector

$$\langle\langle \mathbf{h} \rangle\rangle = \overline{\mathbf{W}}\,\mathbf{h} \tag{49}$$

can be obtained directly from the vector $\mathbf{h}$ representing the physical properties using the matrix

$$\mathbf{W} = \mathbf{W}_{\hat{B},H} = \left(\mathbf{W}_{\hat{B},H}\right)_{ij} = \int_{x'_-}^{x'_+} \hat{B}_i(x') H_j(x') \sum_{k=1}^{N_{\hat{B}}} \hat{B}_k(x') \mathrm{d}x' \tag{50}$$

that is independent of the physical properties. It follows that the vector $\hat{\mathbf{p}}_{\hat{B}_j}$ can be approximated using

$$\hat{\mathbf{p}}_{\hat{B}} \approx \langle\langle \mathbf{h} \rangle\rangle \circ \hat{\mathbf{f}}_{\hat{B}}, \tag{51}$$

where $\circ$ denotes the element-wise or Hadamard product. The approximation $\overline{\mathbf{M}}_h \approx \mathbf{M}_h$ is a common practice in finite-element modelling called mass lumping. Note, however, that performing the mass lumping on the matrix $\mathbf{M}_h$ constructed from the orthogonal basis $\hat{B}$ guarantees that one obtains an exact result when the weight function $h(x')$ is constant. This is not true when the mass lumping is applied to the regular mass matrices $\mathbf{M_{11}}\,\mathbf{M_{22}}$ in eq. (19), for example. That aspect is key in the proposed approach and gives excellent approximations when the physical properties are varying smoothly and the basis functions in $\hat{B}_i$ are spatially localized. As for the DFD operator, we can extend this approach to higher dimension using the tensor product. For example, the 2-D counterpart of the vector in eq. (49) is

$$\langle\langle \mathbf{H} \rangle\rangle = \left(\overline{\mathbf{W}}_\mathbf{x} \otimes \overline{\mathbf{W}}_\mathbf{y}\right)\,\mathbf{H}, \tag{52}$$

where the matrix $\mathbf{H}$ contains the expansion coefficients of $h(x', y')$ represented in basis $H(x', y') = X_H(x')Y_H(y')$, and $\overline{\mathbf{W}}_\mathbf{x}$ and $\overline{\mathbf{W}}_\mathbf{y}$ can be computed using eq. (50) with the substitutions $\hat{B}_i = \hat{X}_i$, $\hat{H}_j = \hat{X_{H_j}}$ and $\hat{B}_i = \hat{Y}_i$, $\hat{H}_j = \hat{Y_{H_j}}$, assuming that $\hat{X}_i$ and $\hat{Y}_i$ are used to represent the field variables.

3 DFD MODELLING IN THE TIME DOMAIN

3.1 1-D wave propagation modelling and stability analysis

This section presents a 1-D DFD scheme for modelling wave propagation in the time domain. It is shown that when the adjoint DFD operators introduced in Section 2.3 are used to discretize the wave equation, we obtain a numerical scheme that is stable under the classic Courant (CFL) condition.

In one dimension the wave equation can be written using the displacement–stress formulation

$$\rho\ddot{u} = \frac{\partial}{\partial_x}\sigma \tag{53a}$$

$$\sigma = k\frac{\partial}{\partial_x}u. \tag{53b}$$

where u is the displacement, $\ddot{u}$ is the acceleration, σ is the 1-D stress, ρ is the density and k is an elastic constant. Eq. (53) can be discretized on a single domain by representing the displacement u and the stress σ in the orthogonal bases $\hat{B}_1$ and $\hat{B}_2$, respectively, and, the density ρ and the elastic constant k in an arbitrary basis H. In this case, the discrete wave equation is obtained directly by substituting the spatial

partial derivatives with the operators in eqs (23a) and (23b) computed using $d = 1$. Remember that those operators encompass the boundary conditions which are specified by the coefficients $\alpha_{\pm}$. The time derivative can be approximated using the explicit second order finite-difference scheme and the material properties can be replaced by the matrices $[\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_1]$ and $[\langle\!\langle \mathbf{k} \rangle\!\rangle_2]$ computed using eq. (49). We obtain the following numerical scheme for solving the wave equation in the time domain

$$[\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_1] \frac{\hat{\mathbf{u}}^{n+1} - 2\hat{\mathbf{u}}^{n} + \hat{\mathbf{u}}^{n-1}}{\Delta_t} = -\hat{\mathbf{D}}_{\mathbf{21}}^{T} \hat{\boldsymbol{\sigma}}^{n} \tag{54a}$$

$$\hat{\boldsymbol{\sigma}}^{n} = [\langle\!\langle \mathbf{k} \rangle\!\rangle_2] \, \hat{\mathbf{D}}_{\mathbf{21}} \mathbf{u}^{n}. \tag{54b}$$

At each time step, the stress is computed first using eq. (54b). Then, knowing the displacement at the time steps n and $n-1$, eq. (54a) is used to obtain the displacement at time step $n+1$. When the operators $\hat{\mathbf{D}}_{\mathbf{21}} = \mathbf{L_2}^{-1} \overset{*}{\mathbf{D}}^{(1)} (\mathbf{L_1}^{-1})^T$ and $\hat{\mathbf{D}}_{\mathbf{21}}^{T} = \mathbf{L_1}^{-1} (\overset{*}{\mathbf{D}}^{(1)})^T (\mathbf{L_2}^{-1})^T$ are computed using the bases presented in the Appendix, the matrix $\overset{*}{\mathbf{D}}^{(1)}$ is band diagonal and the matrices $\mathbf{L_2}$ and $\mathbf{L_2}$ are lower triangular band diagonal matrices. The computational cost associated with the operator $\overset{*}{\mathbf{D}}^{(1)}$ is the same as that of a standard finite-difference matrix (i.e. when the bandwidths of the matrices are the same). The operators $\mathbf{L_1}^{-1}$, $\mathbf{L_2}^{-1}$, $(\mathbf{L_1}^{-1})^T$ and $(\mathbf{L_2}^{-1})^T$ can be efficiently implemented using forward and backward substitution algorithms. Their computational cost is about half that of the operator $\overset{*}{\mathbf{D}}^{(1)}$. Thus the total computational cost of the algorithm in eq. (54) is about twice that of a comparable finite-difference algorithm (when the bandwidth of the FD matrix is the same as that of $\overset{*}{\mathbf{D}}^{(1)}$).

To discuss the stability of the numerical scheme, it is convenient to write it using the compact form

$$\mathbf{M} \frac{\mathbf{u}^{n+1} - 2\mathbf{u}^{n} + \mathbf{u}^{n-1}}{\Delta_t} = -\mathbf{K}\mathbf{u}^{n}, \tag{55}$$

where

$$\mathbf{M} = [\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_1] \tag{56a}$$

$$\mathbf{K} = \hat{\mathbf{D}}_{\mathbf{21}}^{T} \, [\langle\!\langle \mathbf{k} \rangle\!\rangle_2] \, \hat{\mathbf{D}}_{\mathbf{21}}, \tag{56b}$$

that is the general form of the discrete wave equation (Fichtner 2010; Hughes 2012). It is known that the time domain numerical scheme in eq. (55) is conditionally stable if $\mathbf{M}$ is a diagonal positive definite matrix and $\mathbf{K}$ is a symmetric positive definite matrix (Samarskii 2001; Mattsson & Nordström 2004; Dovgilovich & Sofronov 2015). In this case, the numerical scheme is stable when it satisfies an analogous to the classic Courant–Friedrichs–Lewy (CFL) convergence condition

$$\Delta_t \leq \Delta_t^{\mathrm{CFL}} \approx C \min \left[\frac{\Delta_x}{v} \right], \tag{57}$$

where Δ_t^{CFL} denotes the largest time step for which the scheme is stable, C is a small constant, Δ_x is the grid spacing and v is the wave velocity. Because the DFD operators are adjoint, the matrix $\mathbf{K}$ has the general form $\mathbf{A}^T\mathbf{B}\mathbf{A}$ where $\mathbf{B} = [\langle\!\langle \mathbf{k} \rangle\!\rangle_2]$ is a diagonal matrix by construction. Therefore, $\mathbf{K}$ is positive semidefinite if $\mathbf{B}$ is positive. The matrix $\mathbf{M} = [\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_1]$ is diagonal by construction. Consequently, the numerical scheme in eq. (54) is conditionally stable if $[\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_1]$ and $[\langle\!\langle \mathbf{k} \rangle\!\rangle_2]$ are positive matrices. This is usually the case for realistic smoothly varying distributions, further, the positiveness of these matrices can be easily verified prior running a simulation.

So far, we have seen that the adjoint DFD operators in Section 2.3 directly lead to a stable numerical scheme when a single domain is used. When modelling wave propagation using multiple domains, the boundary terms (i.e. the sum terms in eq. 21) need to be accounted for when computing the partial derivatives of the wavefield. In this case, the boundary coefficients $\alpha_{\pm}$ in eq. (24) need to be set consistently to ensure that the total DFD operators (that evaluate the derivative in the whole computational domain) stay adjoint. For example, if the computational domain is divided in N intervals $\Omega_i = [x_i', x_{i+1}']$ $(i = 0, N)$, a continuity condition of the following type needs to be enforced

$$\alpha_{-[i]} = 1 - \alpha_{+[i-1]} = \alpha_i \qquad \text{for all} \qquad i = 1, N. \tag{58}$$

Here, the subscript $[i]$ is used to indicate the domain Ω_i in which the parameters $\alpha_{\pm}$ are defined. From now on, a similar notation is used for the other variables. The condition in eq. (58) translates the fact that the values of wavefield are assumed to be uniquely defined at the boundaries x_i' between the different domains, that is

$$u(x_i') = u(x_+')_{[i-1]} = u(x_-')_{[i]} = \alpha_i \, u_{[i-1]}^{+} + (1 - \alpha_i) \, u_{[i]}^{-} \tag{59a}$$

$$\sigma(x_i') = \sigma(x_+')_{[i-1]} = \sigma(x_-')_{[i]} = (1 - \alpha_i) \, \sigma_{[i-1]}^{+} + \alpha_i \, \sigma_{[i]}^{-}, \tag{59b}$$

where

$$u_{[i]}^{\pm} = (\mathbf{q_1}^{\pm})_{[i]}^{T} (\mathbf{L_1}^{-1})_{[i]}^{T} \hat{\mathbf{u}}_{[i]} \tag{59c}$$

$$\sigma_{[i]}^{\pm} = (\mathbf{q_2}^{\pm})_{[i]}^{T} (\mathbf{L_2}^{-1})_{[i]}^{T} \hat{\boldsymbol{\sigma}}_{[i]} \tag{59d}$$

are the boundary values computed in the domain Ω_i from the expansion coefficients vector $\hat{\mathbf{u}}_{[i]}$ and $\hat{\boldsymbol{\sigma}}_{[i]}$. It is important to note that the continuity conditions $u(x_+')_{[i-1]} = u(x_-')_{[i]}$ and $\sigma(x_+')_{[i-1]} = \sigma(x_-')_{[i]}$ are not satisfied by the numerical solution that is discontinuous between two consecutive elements. In general, we have $u_{[i-1]}^{+} \neq u_{[i]}^{-}$ and $\sigma_{[i-1]}^{+} \neq \sigma_{[i]}^{-}$. However, eq. (59) should be used when evaluating the wavefield

at the interfaces between the elements. When evaluating the partial derivatives in eq. (53) using eq. (21) with the substitutions $\hat{\mathbf{f}}_\mathbf{1} \to \hat{\mathbf{u}}$, $\hat{\mathbf{g}}_\mathbf{2} \to \hat{\boldsymbol{\sigma}}, f^\pm \to u^\pm$ and $g^\pm \to \sigma^\pm$, the numerical scheme in eq. (54) becomes

$$[\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_\mathbf{1}]_{[i]} \frac{\hat{\mathbf{u}}_{[i]}^{n+1} - 2\hat{\mathbf{u}}_{[i]}^{n} + \hat{\mathbf{u}}_{[i]}^{n-1}}{\Delta_t} = \left[\hat{\mathbf{D}}_{\mathbf{21}_{[i]}}^T \hat{\boldsymbol{\sigma}}_{[i]}^n - (\mathbf{B}_{[i-1]}^+)^T \hat{\boldsymbol{\sigma}}_{[i-1]}^n + (\mathbf{B}_{[i+1]}^-)^T \hat{\boldsymbol{\sigma}}_{[i+1]}^n\right] \tag{60a}$$

$$\hat{\boldsymbol{\sigma}}_{[i]} = [\langle\!\langle \mathbf{k} \rangle\!\rangle_\mathbf{2}]_{[i]} \left[\hat{\mathbf{D}}_{\mathbf{21}_{[i]}} \hat{\mathbf{u}}_{[i]}^n - \mathbf{B}_{[i]}^- \hat{\mathbf{u}}_{[i-1]}^n + \mathbf{B}_{[i]}^+ \hat{\mathbf{u}}_{[i+1]}^n\right], \tag{60b}$$

where

$$\mathbf{B}_{[i]}^- = \alpha_i \, \mathbf{L}_{\mathbf{2}_{[i]}}^{-1} \mathbf{q}_{\mathbf{2}_{[i]}}^- (\mathbf{q}_\mathbf{1}^+)_{[i-1]}^T (\mathbf{L}_\mathbf{1}^{-1})_{[i-1]}^T \tag{60c}$$

$$\mathbf{B}_{[i]}^+ = (1 - \alpha_{i+1}) \, \mathbf{L}_{\mathbf{2}_{[i]}}^{-1} \mathbf{q}_{\mathbf{2}_{[i]}}^+ (\mathbf{q}_\mathbf{1}^-)_{[i+1]}^T (\mathbf{L}_\mathbf{1}^{-1})_{[i+1]}^T. \tag{60d}$$

In each domain Ω_i, the wavefield can be updated by first evaluating eq. (60b) knowing the displacement in the neighbouring domains, and then evaluating eq. (60a) knowing the stress in the neighbouring domains. Note that the continuity condition in eq. (58) ensures that the boundary operators $\mathbf{B}_{[i]}^\pm$ in eq. (60a) are the transpose of those in eq. (60b). Practically, to compute the stress we take the following steps in all domains:

(i) Solve $\mathbf{L}_{\mathbf{1}_{[i]}}^T \mathbf{a}_{[i]} = \hat{\mathbf{u}}_{[i]}^n$.
(ii) Compute $\mathbf{b}_{[i]} = \overset{*}{\mathbf{D}}_{[i]}^{(1)} \mathbf{a}_{[i]}$.
(iii) Update the first and last elements of $\mathbf{b}$ using $\begin{cases} b_{1_{[i]}} \Leftarrow b_{1_{[i]}} + \alpha_i \, b_{N_{[i-1]}} \\ b_{N_{[i]}} \Leftarrow b_{N_{[i]}} + (1 - \alpha_{i+1}) \, b_{1_{[i+1]}} \end{cases}$.
(iv) Solve $\mathbf{L}_{\mathbf{2}_{[i]}} \mathbf{c}_{[i]} = \mathbf{b}_{[i]}$.
(v) Compute $\hat{\boldsymbol{\sigma}}_{[i]}^n = [\langle\!\langle \mathbf{k} \rangle\!\rangle_\mathbf{2}]_{[i]} \, \mathbf{c}_{[i]}$.

Similarly, to update the displacement we take the following steps in all domains:

(i) Solve $\mathbf{L}_{\mathbf{2}_{[i]}}^T \mathbf{a}_{[i]} = \hat{\boldsymbol{\sigma}}_{[i]}^n$.
(ii) Compute $\mathbf{b}_{[i]} = \left(\overset{*}{\mathbf{D}}_{[i]}^{(1)}\right)^T \mathbf{a}_{[i]}$.
(iii) Update the first and last elements of $\mathbf{b}$ using $\begin{cases} b_{1_{[i]}} \Leftarrow b_{1_{[i]}} + (1 - \alpha_i) \, b_{N_{[i-1]}} \\ b_{N_{[i]}} \Leftarrow b_{N_{[i]}} + \alpha_{i+1} \, b_{1_{[i+1]}} \end{cases}$.
(iv) Solve $\mathbf{L}_{\mathbf{1}_{[i]}} \mathbf{c} = \mathbf{b}_{[i]}$.
(v) Compute $\hat{\mathbf{u}}_{[i]}^{n+1} = 2\hat{\mathbf{u}}_{[i]}^n - \hat{\mathbf{u}}_{[i]}^{n-1} + \Delta_t \, [\langle\!\langle \boldsymbol{\rho} \rangle\!\rangle_\mathbf{1}]_{[i]}^{-1} \, \mathbf{c}_{[i]}$.

By assembling the linear system in eq. (60), we obtain the update equation for the complete computational domain. We have:

$$\mathbf{M}_g \frac{\hat{\mathbf{u}}_g^{n+1} - 2\hat{\mathbf{u}}_g^n + \hat{\mathbf{u}}_g^{n-1}}{\Delta_t} = -\mathbf{D}_g^T \mathbf{K}_g \mathbf{D}_g \hat{\mathbf{u}}_g, \tag{61a}$$

where the aggregated displacement vector is

$$\hat{\mathbf{u}}_g = \begin{bmatrix} \hat{\mathbf{u}}_{[0]} \ldots\ldots\ldots \hat{\mathbf{u}}_{[N]} \end{bmatrix}^T, \tag{61b}$$

the global DFD operator is

$$\mathbf{D}_g = \begin{bmatrix} \hat{\mathbf{D}}_{\mathbf{21}_{[0]}} & \mathbf{B}_{[0]}^+ & & & & \\ \mathbf{B}_{[1]}^- & \hat{\mathbf{D}}_{\mathbf{21}_{[1]}} & \mathbf{B}_{[1]}^+ & & (0) & \\ & \mathbf{B}_{[3]}^- & \ddots & \ddots & & \\ & & \ddots & \ddots & \mathbf{B}_{[N-2]}^+ & \\ & (0) & & \mathbf{B}_{[N-1]}^- & \hat{\mathbf{D}}_{\mathbf{21}_{[N-1]}} & \mathbf{B}_{[N-1]}^+ \\ & & & & \mathbf{B}_{[N]}^- & \hat{\mathbf{D}}_{\mathbf{21}_{[N]}} \end{bmatrix} \tag{61c}$$

and the matrices that contain the distributions of physical properties in the different domains are

$$\mathbf{M}_g = \begin{bmatrix} [\langle\langle \boldsymbol{\rho} \rangle\rangle_1]_{[0]} & & \\ & \ddots & \\ & & [\langle\langle \boldsymbol{\rho} \rangle\rangle_1]_{[N]} \end{bmatrix} \tag{61d}$$

$$\mathbf{K}_g = \begin{bmatrix} [\langle\langle \mathbf{k} \rangle\rangle_1]_{[0]} & & \\ & \ddots & \\ & & [\langle\langle \mathbf{k} \rangle\rangle_1]_{[N]} \end{bmatrix}. \tag{61e}$$

We see that the global DFD operators $\mathbf{D}_g$ and $\mathbf{D}_g^T$ are still adjoint when the continuity condition in eq. (58) is satisfied. Therefore, the multidomain scheme in eq. (61a) is also stable under the classic Courant condition in eq. (57). Practically, the linear system in eq. (61a) does not need to be assembled, and eq. (60) should be used to update the wavefield in the different domains. From a computational perspective, in eq. (60), it is better to factorize the multiplications by the matrices $\mathbf{L_{1/2}}$ and $\mathbf{L_{1/2}}^T$ as detailed in the next section. The computational cost of the boundary operators is negligible as they simply communicate the first and last expansion coefficients between neighbouring elements (when the operators are constructed using the bases described in the Appendix). Thus the computational cost of the multidomain algorithm in eq. (60) is marginally higher than that of the single-domain algorithm in eq. (54) for a given problem dimension. The 1-D algorithm is extended to 2-D in the next section.

3.2 2-D wave propagation modelling

The governing equations for modelling wave propagation in a 2-D linear isotropic elastic media can be written using the displacement–stress formulation

$$\rho \ddot{u}_x = \partial_x \sigma_{xx} + \partial_y \sigma_{xy} + f_x \tag{62a}$$

$$\rho \ddot{u}_y = \partial_x \sigma_{xy} + \partial_y \sigma_{yy} + f_y \tag{62b}$$

$$\sigma_{xx} = (\lambda + 2\mu)\partial_x u_x + \lambda \partial_y u_y \tag{62c}$$

$$\sigma_{yy} = \lambda \partial_x u_x + (\lambda + 2\mu)\partial_y u_y \tag{62d}$$

$$\sigma_{xy} = \mu(\partial_x u_y + \partial_y u_x), \tag{62e}$$

where u_i is the displacement in direction i, $\ddot{u}_i$ is the acceleration in direction i, σ_{ij} are the stresses, ρ is the density, λ and μ are the Lamé parameters and ∂_i denotes the partial derivative with respect to the spatial direction i. Using the smooth coordinate transformation

$$x = x(x', y') \quad y = y(x', y') \tag{63}$$

illustrated in Fig. 2, we can write eq. (62) in the conservative form (Thompson *et al.* 1982; Appelö & Petersson 2009; Dovgilovich & Sofronov 2015)

$$|\mathbf{J}| \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} \ddot{u}_x \\ \ddot{u}_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x'} & \frac{\partial}{\partial y'} & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial x'} & \frac{\partial}{\partial y'} \end{bmatrix} |\mathbf{J}| \begin{bmatrix} \frac{\partial x'}{\partial x} & 0 & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & 0 & \frac{\partial y'}{\partial y} \\ 0 & \frac{\partial x'}{\partial y} & \frac{\partial x'}{\partial x} \\ 0 & \frac{\partial y'}{\partial y} & \frac{\partial y'}{\partial x} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} + |\mathbf{J}| \begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{64a}$$

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{bmatrix}=\begin{bmatrix}\lambda+2\mu & \lambda & 0\\ \lambda & \lambda+2\mu & 0\\ 0 & 0 & \mu\end{bmatrix}\begin{bmatrix}\frac{\partial x'}{\partial x} & \frac{\partial y'}{\partial x} & 0 & 0\\ 0 & 0 & \frac{\partial x'}{\partial y} & \frac{\partial y'}{\partial y}\\ \frac{\partial x'}{\partial y} & \frac{\partial y'}{\partial y} & \frac{\partial x'}{\partial x} & \frac{\partial y'}{\partial x}\end{bmatrix}\begin{bmatrix}\frac{\partial}{\partial x'} & 0\\ \frac{\partial}{\partial y'} & 0\\ 0 & \frac{\partial}{\partial x'}\\ 0 & \frac{\partial}{\partial y'}\end{bmatrix}\begin{bmatrix}u_x\\ u_y\end{bmatrix}, \tag{64b}$$

where $|\mathbf{J}|$ denotes the determinant of the Jacobian matrix. We have:

$$\mathbf{J}(x',y')=\begin{bmatrix}\frac{\partial x}{\partial x'} & \frac{\partial x}{\partial y'}\\ \frac{\partial y}{\partial x'} & \frac{\partial y}{\partial y'}\end{bmatrix}\qquad \mathbf{J}^{-1}(x',y')=\begin{bmatrix}\frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y}\\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y}\end{bmatrix}\quad |\mathbf{J}|=\frac{\partial x}{\partial x'}\frac{\partial y}{\partial y'}-\frac{\partial y}{\partial x'}\frac{\partial x}{\partial y'} \tag{65a}$$

In order to discretize eq. (64), within each domain (e.g. domain Ω in Fig. 2) the fields variables, the physical properties and the geometric variables are first represented as linear combinations similar to that in eq. (28) using the 2-D basis functions

$$B_{1_{i,j}}(x',y')=X_{1_i}(x')Y_{1_j}(y')\quad 1\le i\le N_{X_1}\quad 1\le j\le N_{Y_1} \tag{66a}$$

$$B_{2_{i,j}}(x',y')=X_{2_i}(x')Y_{2_j}(y')\quad 1\le i\le N_{X_2}\quad 1\le j\le N_{Y_2} \tag{66b}$$

$$H_{i,j}(x',y')=X_{H_i}(x')Y_{H_j}(y')\quad 1\le i\le N_{X_K}\quad 1\le j\le N_{Y_K}, \tag{66c}$$

where X_1, X_2, X_H and, Y_1, Y_2, Y_H are 1-D bases defined on domain $[x'_-, x'_+]$ and $[y'_-, y'_+]$, respectively. The displacements u_x, u_y are represented in basis B_1 with their expansions coefficients stored in the matrices $\mathbf{U_x}$ and $\mathbf{U_y}$, respectively, and, the stresses σ_{xx}, σ_{yy} and σ_{xy} are represented in basis B_2 with their expansions coefficients in the matrices $\boldsymbol{\sigma}_{\mathbf{xx}}$, $\boldsymbol{\sigma}_{\mathbf{yy}}$ and $\boldsymbol{\sigma}_{\mathbf{xy}}$, respectively. We assume that all other variables are represented in some arbitrary basis H that may be different for each variable. By computing and factorizing the 1-D mass matrices [i.e. using eq. (9a) followed by a Cholesky factorization]

$$\mathbf{M}_{\mathbf{x_1}}=\mathbf{M}_{X_1,X_1}=\mathbf{L}_{\mathbf{x_1}}\mathbf{L}_{\mathbf{x_1}}^T\quad \mathbf{M}_{\mathbf{x_2}}=\mathbf{M}_{X_2,X_2}=\mathbf{L}_{\mathbf{x_2}}\mathbf{L}_{\mathbf{x_2}}^T\quad \mathbf{M}_{\mathbf{x_1}}=\mathbf{M}_{Y_1,Y_1}=\mathbf{L}_{\mathbf{y_1}}\mathbf{L}_{\mathbf{y_1}}^T\quad \mathbf{M}_{\mathbf{y_2}}=\mathbf{M}_{Y_2,Y_2}=\mathbf{L}_{\mathbf{y_2}}\mathbf{L}_{\mathbf{y_2}}^T \tag{67a}$$

we can construct the orthogonal bases

$$\hat{B}_{1_{i,j}}(x',y')=\hat{X}_{1_i}(x')\hat{Y}_{1_j}(y')\quad \hat{X}_{1_i}(x')=\sum_{j=1}^{N_{X_1}}(\mathbf{L}_{\mathbf{x_1}}^{-1})_{ij}X_{1_j}(x')\quad \hat{Y}_{1_i}(y')=\sum_{j=1}^{N_{Y_1}}(\mathbf{L}_{\mathbf{y_1}}^{-1})_{ij}Y_{1_j}(y') \tag{68a}$$

$$\hat{B}_{2_{i,j}}(x',y')=\hat{X}_{2_i}(x')\hat{Y}_{2_j}(y')\quad \hat{X}_{2_i}(x')=\sum_{j=1}^{N_{X_2}}(\mathbf{L}_{\mathbf{x_2}}^{-1})_{ij}X_{2_j}(x')\quad \hat{Y}_{2_i}(y')=\sum_{j=1}^{N_{Y_2}}(\mathbf{L}_{\mathbf{y_2}}^{-1})_{ij}Y_{2_j}(y') \tag{68b}$$

and the dual bases

$$\overset{*}{B}_{1_{i,j}}(x',y')=\overset{*}{X}_{1_i}(x')\overset{*}{Y}_{1_j}(y')\quad \overset{*}{X}_{1_i}(x')=\sum_{j=1}^{N_{X_1}}(\mathbf{M}_{\mathbf{x_1}}^{-1})_{ij}X_{1_j}(x')\quad \overset{*}{Y}_{1_i}(y')=\sum_{j=1}^{N_{Y_1}}(\mathbf{M}_{\mathbf{y_1}}^{-1})_{ij}Y_{1_j}(y') \tag{69a}$$

$$\overset{*}{B}_{2_{i,j}}(x',y')=\overset{*}{X}_{2_i}(x')\overset{*}{Y}_{2_j}(y')\quad \overset{*}{X}_{2_i}(x')=\sum_{j=1}^{N_{X_2}}(\mathbf{M}_{\mathbf{x_2}}^{-1})_{ij}X_{2_j}(x')\quad \overset{*}{Y}_{2_i}(y')=\sum_{j=1}^{N_{Y_2}}(\mathbf{M}_{\mathbf{y_2}}^{-1})_{ij}Y_{2_j}(y'). \tag{69b}$$

Using these bases, the displacements u_x, u_y may be equally represented in the bases $\hat{B}_1$ or $\overset{*}{B}_1$ using the expansion coefficient matrices $\hat{\mathbf{U}}_\mathbf{x}$ and $\hat{\mathbf{U}}_\mathbf{y}$ or $\overset{*}{\mathbf{U}}_x$ and $\overset{*}{\mathbf{U}}_y$, respectively, and, the stresses σ_{xx}, σ_{yy} and σ_{xy} may be equally represented in the bases $\hat{B}_2$ or $\overset{*}{B}_2$ using the expansion coefficient matrices $\hat{\boldsymbol{\sigma}}_{\mathbf{xx}}$, $\hat{\boldsymbol{\sigma}}_{\mathbf{yy}}$ and $\hat{\boldsymbol{\sigma}}_{\mathbf{xy}}$ or $\overset{*}{\boldsymbol{\sigma}}_{\mathbf{xx}}$, $\overset{*}{\boldsymbol{\sigma}}_{\mathbf{yy}}$ and $\overset{*}{\boldsymbol{\sigma}}_{\mathbf{xy}}$, respectively. By approximating the time derivatives $\ddot{u}_i$ using a second order centred finite-difference, and, by substituting the spatial partial derivatives in eq. (64) with the discrete operator in eq. (39), where the 1-D DFD operators are computed using eq. (23), we obtain the discrete linear system

$$\hat{\boldsymbol{\sigma}}_{\mathbf{xx}}=\langle\!\langle\lambda+2\mu\rangle\!\rangle_2\circ\left[\left\langle\!\left\langle\frac{\partial x'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{x}^{n^{(1,0)}}+\left\langle\!\left\langle\frac{\partial y'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{x}^{n^{(0,1)}}\right]+\langle\!\langle\lambda\rangle\!\rangle_2\circ\left[\left\langle\!\left\langle\frac{\partial x'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{y}^{n^{(1,0)}}+\left\langle\!\left\langle\frac{\partial y'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{y}^{n^{(0,1)}}\right] \tag{70a}$$

$$\hat{\boldsymbol{\sigma}}_{\mathbf{yy}}=\langle\!\langle\lambda\rangle\!\rangle_2\circ\left[\left\langle\!\left\langle\frac{\partial x'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{x}^{n^{(1,0)}}+\left\langle\!\left\langle\frac{\partial y'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{x}^{n^{(0,1)}}\right]+\langle\!\langle\lambda+2\mu\rangle\!\rangle_2\circ\left[\left\langle\!\left\langle\frac{\partial x'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{y}^{n^{(1,0)}}+\left\langle\!\left\langle\frac{\partial y'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{y}^{n^{(0,1)}}\right] \tag{70b}$$

$$\hat{\boldsymbol{\sigma}}_{\mathbf{xy}}=\langle\!\langle\mu\rangle\!\rangle_2\circ\left[\left\langle\!\left\langle\frac{\partial x'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{x}^{n^{(1,0)}}+\left\langle\!\left\langle\frac{\partial y'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{x}^{n^{(0,1)}}\right]+\langle\!\langle\mu\rangle\!\rangle_2\circ\left[\left\langle\!\left\langle\frac{\partial x'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{y}^{n^{(1,0)}}+\left\langle\!\left\langle\frac{\partial y'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\mathbf{U}}_\mathbf{y}^{n^{(0,1)}}\right] \tag{70c}$$

$$\hat{\boldsymbol{S}}_{\mathbf{xx}}=\langle\!\langle|\mathbf{J}|\rangle\!\rangle_2\circ\left(\left\langle\!\left\langle\frac{\partial x'}{\partial x}\right\rangle\!\right\rangle_2\circ\hat{\boldsymbol{\sigma}}_{\mathbf{xx}}+\left\langle\!\left\langle\frac{\partial x'}{\partial y}\right\rangle\!\right\rangle_2\circ\hat{\boldsymbol{\sigma}}_{\mathbf{xy}}\right) \tag{71a}$$

$$\hat{\mathbf{S}}_{\mathbf{xy}} = \langle\langle |\mathbf{J}| \rangle\rangle_{\mathbf{2}} \circ \left(\left\langle\left\langle \frac{\partial y'}{\partial x} \right\rangle\right\rangle_{\mathbf{2}} \circ \hat{\boldsymbol{\sigma}}_{\mathbf{xx}} + \left\langle\left\langle \frac{\partial y'}{\partial y} \right\rangle\right\rangle_{\mathbf{2}} \circ \hat{\boldsymbol{\sigma}}_{\mathbf{xy}} \right) \tag{71b}$$

$$\hat{\mathbf{S}}_{\mathbf{yx}} = \langle\langle |\mathbf{J}| \rangle\rangle_{\mathbf{2}} \circ \left(\left\langle\left\langle \frac{\partial x'}{\partial y} \right\rangle\right\rangle_{\mathbf{2}} \circ \hat{\boldsymbol{\sigma}}_{\mathbf{yy}} + \left\langle\left\langle \frac{\partial x'}{\partial x} \right\rangle\right\rangle_{\mathbf{2}} \circ \hat{\boldsymbol{\sigma}}_{\mathbf{xy}} \right) \tag{71c}$$

$$\hat{\mathbf{S}}_{\mathbf{yy}} = \langle\langle |\mathbf{J}| \rangle\rangle_{\mathbf{2}} \circ \left(\left\langle\left\langle \frac{\partial y'}{\partial y} \right\rangle\right\rangle_{\mathbf{2}} \circ \hat{\boldsymbol{\sigma}}_{\mathbf{yy}} + \left\langle\left\langle \frac{\partial y'}{\partial x} \right\rangle\right\rangle_{\mathbf{2}} \circ \hat{\boldsymbol{\sigma}}_{\mathbf{xy}} \right) \tag{71d}$$

$$\hat{\mathbf{U}}_{\mathbf{x}}^{n+1} = 2\hat{\mathbf{U}}_{\mathbf{x}}^{n} - \hat{\mathbf{U}}_{\mathbf{x}}^{n-1} + \Delta_t^2 \langle\langle \rho \rangle\rangle_{\mathbf{1}}^{-1} \circ \left[\langle\langle |\mathbf{J}| \rangle\rangle_{\mathbf{1}}^{-1} \circ \left(\hat{\boldsymbol{S}}_{\mathbf{xx}}^{(1,0)} + \hat{\boldsymbol{S}}_{\mathbf{xy}}^{(0,1)} \right) + \mathbf{F}_{\mathbf{x}} \right] \tag{72a}$$

$$\hat{\mathbf{U}}_{\mathbf{y}}^{n+1} = 2\hat{\mathbf{U}}_{\mathbf{y}}^{n} - \hat{\mathbf{U}}_{\mathbf{y}}^{n-1} + \Delta_t^2 \langle\langle \rho \rangle\rangle_{\mathbf{1}}^{-1} \circ \left[\langle\langle |\mathbf{J}| \rangle\rangle_{\mathbf{1}}^{-1} \circ \left(\hat{\boldsymbol{S}}_{\mathbf{yx}}^{(1,0)} + \hat{\boldsymbol{S}}_{\mathbf{yy}}^{(0,1)} \right) + \mathbf{F}_{\mathbf{y}} \right], \tag{72b}$$

where $\langle\langle\rangle\rangle_{\mathbf{1}}$ and $\langle\langle\rangle\rangle_{\mathbf{2}}$ denote the averaged properties of the quantities within the brackets computed using eq. (52), with $\hat{B} = \hat{B}_1$ and $\hat{B} = \hat{B}_2$, respectively.

The solution to the discrete linear system is obtained recursively. At each time step, knowing the displacements $\hat{\mathbf{U}}^{n-1}$ and $\hat{\mathbf{U}}^{n}$ at times $(n-1)\Delta_t$ and $n\Delta_t$, the displacement $\hat{\mathbf{U}}^{\mathbf{n+1}}$ at time $(n+1)\Delta_t$ is computed by evaluating eqs (70), (71) and (72) successively. In that sequence, most of the computational burden lies in the computation of the partial derivatives $\hat{\mathbf{U}}_{\mathbf{x}}^{n(0,1)}$, $\hat{\mathbf{U}}_{\mathbf{x}}^{n(1,0)}$, $\hat{\mathbf{U}}_{\mathbf{y}}^{n(0,1)}$, $\hat{\mathbf{U}}_{\mathbf{y}}^{n(1,0)}$, $\hat{\boldsymbol{S}}_{\mathbf{xx}}^{(1,0)}$, $\hat{\boldsymbol{S}}_{\mathbf{xy}}^{(0,1)}$, $\hat{\boldsymbol{S}}_{\mathbf{yx}}^{(1,0)}$ and $\hat{\boldsymbol{S}}_{\mathbf{yy}}^{(0,1)}$. These partial derivatives are obtained by applying two types of operators to the field variables:

(i) Operators of the form $(\mathbf{A}_{\mathbf{x}} \otimes \mathbf{I})$, $(\mathbf{A}_{\mathbf{x}} \otimes \mathbf{I})^T = \left(\mathbf{A}_{\mathbf{x}}{}^T \otimes \mathbf{I}\right)$, $\left(\mathbf{I} \otimes \mathbf{A}_{\mathbf{y}}\right)$ and $\left(\mathbf{I} \otimes \mathbf{A}_{\mathbf{y}}\right)^T = \left(\mathbf{I} \otimes \mathbf{A}_{\mathbf{y}}{}^T\right)$ which apply either a matrix $\mathbf{A}_{\mathbf{x}}$ (or its transpose) to all columns of the matrix variables or a matrix $\mathbf{A}_{\mathbf{y}}$ (or its transpose) to all rows of the matrix variables. When the matrices $\mathbf{A}_{\mathbf{x}}$ and $\mathbf{A}_{\mathbf{y}}$ are band diagonal matrices (as in our examples), these operators can be thought as classic finite-difference operators. They perform similar operations and have the same computational cost when the bandwidth of the matrices is the same as the length of the finite-difference stencil. These operators are constructed from the matrices (vectors are taken as single column matrices)

$$\mathbf{q}_{\mathbf{x_1}}^{\pm} = \mathbf{q}_{X_1}^{\pm(1)} \qquad \mathbf{q}_{\mathbf{x_2}}^{\pm} = \mathbf{q}_{X_2}^{\pm(1)} \qquad \mathbf{q}_{\mathbf{y_1}}^{\pm} = \mathbf{q}_{Y_1}^{\pm(1)} \qquad \mathbf{q}_{\mathbf{y_2}}^{\pm} = \mathbf{q}_{Y_2}^{\pm(1)} \tag{73a}$$

$$\overset{*}{\mathbf{b}}_{\mathbf{x_1}}^{\pm} = \pm(1-\alpha_{\pm}^{x})\mathbf{q}_{\mathbf{x_1}}^{\pm} \qquad \overset{*}{\mathbf{b}}_{\mathbf{x_2}}^{\pm} = \pm\alpha_{\pm}^{x}\mathbf{q}_{\mathbf{x_2}}^{\pm} \qquad \overset{*}{\mathbf{b}}_{\mathbf{y_1}}^{\pm} = \pm(1-\alpha_{\pm}^{y})\mathbf{q}_{\mathbf{y_1}}^{\pm} \qquad \overset{*}{\mathbf{b}}_{\mathbf{y_2}}^{\pm} = \pm\alpha_{\pm}^{y}\mathbf{q}_{\mathbf{y_2}}^{\pm} \tag{73b}$$

$$\overset{*}{\mathbf{D}}_{\mathbf{x}}^{(0)} = \mathbf{K}_{X_2,X_1}^{(0)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{x_1}}^{o-(0)} = \mathbf{K}_{X_1,X_2^{o-}}^{o-(0)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{x_2}}^{o-(0)} = \mathbf{K}_{X_2,X_1^{o-}}^{o-(0)} \tag{73c}$$

$$\overset{*}{\mathbf{D}}_{\mathbf{y}}^{(0)} = \mathbf{K}_{Y_2,Y_1}^{(0)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{x_1}}^{o+(0)} = \mathbf{K}_{X_1,X_2^{o+}}^{o+(0)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{x_2}}^{o+(0)} = \mathbf{K}_{X_2,X_1^{o+}}^{o+(0)} \tag{73d}$$

$$\overset{*}{\mathbf{D}}_{\mathbf{x}}^{(1)} = \mathbf{K}_{X_2,X_1}^{(1)} - \alpha_{-}^{x}\mathbf{Q}_{X_2,X_1}^{-(1)} - \alpha_{+}^{x}\mathbf{Q}_{X_2,X_1}^{+(1)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{y_1}}^{-o(0)} = \mathbf{K}_{Y_1,Y_2^{-o}}^{-o(0)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{y_2}}^{-o(0)} = \mathbf{K}_{Y_2,Y_1^{-o}}^{-o(0)} \tag{73e}$$

$$\overset{*}{\mathbf{D}}_{\mathbf{y}}^{(1)} = \mathbf{K}_{Y_2,Y_1}^{(1)} - \alpha_{-}^{y}\mathbf{Q}_{Y_2,Y_1}^{-(1)} - \alpha_{+}^{y}\mathbf{Q}_{Y_2,Y_1}^{+(1)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{y_1}}^{+o(0)} = \mathbf{K}_{Y_1,Y_2^{+o}}^{+o(0)} \qquad \overset{*}{\mathbf{D}}_{\mathbf{y_2}}^{+o(0)} = \mathbf{K}_{Y_2,Y_1^{+o}}^{+o(0)} \tag{73f}$$

that are computed using eq. (9) and stored prior running the simulation.

(ii) Operators of the form $\left(\mathbf{L}_{\mathbf{x}}^{-1} \otimes \mathbf{I}\right)$, $\left(\mathbf{L}_{\mathbf{x}}^{-1} \otimes \mathbf{I}\right)^T = \left((\mathbf{L}_{\mathbf{x}}^{-1})^T \otimes \mathbf{I}\right)$, $\left(\mathbf{I} \otimes \mathbf{L}_{\mathbf{y}}^{-1}\right)$ and $\left(\mathbf{I} \otimes \mathbf{L}_{\mathbf{y}}^{-1}\right)^T = \left(\mathbf{I} \otimes (\mathbf{L}_{\mathbf{y}}^{-1})^T\right)$ which apply either an inverse matrix $\mathbf{L}_{\mathbf{x}}^{-1}$ (or its transpose) to all columns of the matrix variables or an inverse matrix $\mathbf{L}_{\mathbf{y}}^{-1}$ (or its transpose) to all rows of the matrix variables. These operators can be thought as recursive pre- and post-filters that are applied prior and after differentiating the field variables using the operators belonging to the first class above. They can be implemented efficiently using forward and backward substitution and their computational cost is half that of the operators of the first class because they involve upper and lower triangular band diagonal matrices. These operators are constructed from the matrices in eq. (67) which are computed and stored prior running the simulation.

Practically, the partial derivatives $\hat{\mathbf{U}}_{\mathbf{x}}^{n(1,0)}$, $\hat{\mathbf{U}}_{\mathbf{x}}^{n(0,1)}$, $\hat{\mathbf{U}}_{\mathbf{y}}^{n(1,0)}$ and $\hat{\mathbf{U}}_{\mathbf{y}}^{n(0,1)}$ in eq. (70) are computed as follows. The displacements $\hat{\mathbf{U}}_{\mathbf{x}}^{n}$ and $\hat{\mathbf{U}}_{\mathbf{y}}^{n}$ represented in basis $\hat{B}_1$ are first transformed using 1-D backward substitutions to obtain the displacements $\mathbf{U}_{\mathbf{x}}^{n}$ and $\mathbf{U}_{\mathbf{y}}^{n}$ in basis B_1. We have:

$$\mathbf{U}_{\mathbf{x}}^{n} = \left(\mathbf{L}_{\mathbf{x_1}}^{-1} \otimes \mathbf{I}\right)^T \left(\mathbf{I} \otimes \mathbf{L}_{\mathbf{y_1}}^{-1}\right)^T \hat{\mathbf{U}}_{\mathbf{x}}^{n} \qquad \mathbf{U}_{\mathbf{y}}^{n} = \left(\mathbf{L}_{\mathbf{x_1}}^{-1} \otimes \mathbf{I}\right)^T \left(\mathbf{I} \otimes \mathbf{L}_{\mathbf{y_1}}^{-1}\right)^T \hat{\mathbf{U}}_{\mathbf{y}}^{n}. \tag{74a}$$

At this point, because the wavefield is represented in the basis B_1 that has compactly supported basis functions, it can be easily evaluated at arbitrary receivers locations. Then, the boundary vector variables

$$\mathbf{u}_{\mathbf{x}}^{-o} = \left(\left(\mathbf{q}_{\mathbf{x_1}}^{-}\right)^T \otimes \mathbf{I}\right)\mathbf{U}_{\mathbf{x}}^{n} \qquad \mathbf{u}_{\mathbf{y}}^{-o} = \left(\left(\mathbf{q}_{\mathbf{x_1}}^{-}\right)^T \otimes \mathbf{I}\right)\mathbf{U}_{\mathbf{y}}^{n} \tag{75a}$$

$$\mathbf{u}_{\mathbf{x}}^{+o} = \left(\left(\mathbf{q}_{\mathbf{x_1}}^{+}\right)^T \otimes \mathbf{I}\right)\mathbf{U}_{\mathbf{x}}^{n} \qquad \mathbf{u}_{\mathbf{y}}^{+o} = \left(\left(\mathbf{q}_{\mathbf{x_1}}^{+}\right)^T \otimes \mathbf{I}\right)\mathbf{U}_{\mathbf{y}}^{n} \tag{75b}$$

$$\mathbf{u}_{\mathbf{x}}^{o-} = \left(\mathbf{I} \otimes \left(\mathbf{q}_{\mathbf{y_1}}^{-}\right)^T\right)\mathbf{U}_{\mathbf{x}}^{n} \qquad \mathbf{u}_{\mathbf{y}}^{o-} = \left(\mathbf{I} \otimes \left(\mathbf{q}_{\mathbf{y_1}}^{-}\right)^T\right)\mathbf{U}_{\mathbf{y}}^{n} \tag{75c}$$

$$\mathbf{u}_{\mathbf{x}}^{o+} = \left(\mathbf{I} \otimes \left(\mathbf{q}_{\mathbf{y_1}}^{+}\right)^T\right)\mathbf{U}_{\mathbf{x}}^{n} \qquad \mathbf{u}_{\mathbf{y}}^{o+} = \left(\mathbf{I} \otimes \left(\mathbf{q}_{\mathbf{y_1}}^{+}\right)^T\right)\mathbf{U}_{\mathbf{y}}^{n} \tag{75d}$$

are computed and sent to/exchanged with the neighbouring domains/elements (in the case of multidomain modelling). Note that, when the inner basis functions vanish at points x_- and x_+ (i.e. when $X_{1_i}(x_-) = X_{1_i}(x_+) = 0$ for $i = 2, N_{X_1} - 1$ and $Y_{1_i}(x_-) = Y_{1_i}(x_+) = 0$ for $i = 2, N_{Y_1} - 1$, as in Figs A1 and 1), the equations above simply pick up the first/last row/column of the matrices $\mathbf{U}^n$. As the boundary vector variables become available, the partial derivatives can be computed in the dual basis $\overset{*}{B}_2$ using the finite-difference-like scheme

$$\overset{*}{\mathbf{U}}{}^{n(1,0)}_{\mathbf{x}} = \big(\overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{x}} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{y}}\big)\mathbf{U}^n_{\mathbf{x}} + \big(\overset{*}{\mathbf{b}}{}^{-}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{-o^{(0)}}_{\mathbf{y}_2}\big)\mathbf{u}^{-o}_{\mathbf{x}} + \big(\overset{*}{\mathbf{b}}{}^{+}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{+o^{(0)}}_{\mathbf{y}_2}\big)\mathbf{u}^{+o}_{\mathbf{x}} \tag{76a}$$

$$\overset{*}{\mathbf{U}}{}^{n(0,1)}_{\mathbf{x}} = \big(\overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{x}} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{y}}\big)\mathbf{U}^n_{\mathbf{x}} + \big(\overset{*}{\mathbf{D}}{}^{o-^{(0)}}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{-}_{\mathbf{y}_2}\big)\mathbf{u}^{o-}_{\mathbf{x}} + \big(\overset{*}{\mathbf{D}}{}^{o+^{(0)}}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{+}_{\mathbf{y}_2}\big)\mathbf{u}^{o+}_{\mathbf{x}} \tag{76b}$$

$$\overset{*}{\mathbf{U}}{}^{n(1,0)}_{\mathbf{y}} = \big(\overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{x}} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{y}}\big)\mathbf{U}^n_{\mathbf{y}} + \big(\overset{*}{\mathbf{b}}{}^{-}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{-o^{(0)}}_{\mathbf{y}_2}\big)\mathbf{u}^{-o}_{\mathbf{y}} + \big(\overset{*}{\mathbf{b}}{}^{+}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{+o^{(0)}}_{\mathbf{y}_2}\big)\mathbf{u}^{+o}_{\mathbf{y}} \tag{76c}$$

$$\overset{*}{\mathbf{U}}{}^{n(0,1)}_{\mathbf{y}} = \big(\overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{x}} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{y}}\big)\mathbf{U}^n_{\mathbf{y}} + \big(\overset{*}{\mathbf{D}}{}^{o-^{(0)}}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{-}_{\mathbf{y}_2}\big)\mathbf{u}^{o-}_{\mathbf{y}} + \big(\overset{*}{\mathbf{D}}{}^{o+^{(0)}}_{\mathbf{x}_2} \otimes \mathbf{I}\big) \quad \big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{+}_{\mathbf{y}_2}\big)\mathbf{u}^{o+}_{\mathbf{y}}. \tag{76d}$$

Note that one may start to compute the first term to the right-hand side in the above equation prior receiving the boundary vector variables. When using conformal basis functions (i.e. that are the same in all domains/elements), we have $\overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{x}} = \overset{*}{\mathbf{D}}{}^{o-^{(0)}}_{\mathbf{x}} = \overset{*}{\mathbf{D}}{}^{o+^{(0)}}_{\mathbf{x}}$ and $\overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{y}} = \overset{*}{\mathbf{D}}{}^{-o^{(0)}}_{\mathbf{y}} = \overset{*}{\mathbf{D}}{}^{+o^{(0)}}_{\mathbf{y}}$. Thus, if the inner basis functions vanish at the boundaries [i.e. when $X_{2_i}(x_-) = X_{2_i}(x_+) = 0$ for $i = 2, N_{X_2} - 1$ and $Y_{2_i}(x_-) = Y_{2_i}(x_+) = 0$ for $i = 2, N_{Y_2} - 1$], the operators involving the vectors $\overset{*}{\mathbf{b}}$ simply add the vector $\mathbf{u}$ to the first/last column/row of the result matrix. Finally, the partial derivatives are transformed using 1-D forward substitutions to obtain their representation in basis $\hat{B}_2$. We have:

$$\hat{\mathbf{U}}^{n(1,0)}_{\mathbf{x}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)\overset{*}{\mathbf{U}}{}^{n(1,0)}_{\mathbf{x}} \qquad \hat{\mathbf{U}}^{n(1,0)}_{\mathbf{y}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)\overset{*}{\mathbf{U}}{}^{n(1,0)}_{\mathbf{y}} \tag{77a}$$

$$\hat{\mathbf{U}}^{n(0,1)}_{\mathbf{x}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)\overset{*}{\mathbf{U}}{}^{n(0,1)}_{\mathbf{x}} \qquad \hat{\mathbf{U}}^{n(0,1)}_{\mathbf{y}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)\overset{*}{\mathbf{U}}{}^{n(0,1)}_{\mathbf{y}}. \tag{77b}$$

The partial derivatives $\hat{\mathbf{S}}^{(1,0)}_{\mathbf{xx}}$, $\hat{\mathbf{S}}^{(0,1)}_{\mathbf{xy}}$, $\hat{\mathbf{S}}^{(1,0)}_{\mathbf{yx}}$ and $\hat{\mathbf{S}}^{(0,1)}_{\mathbf{yy}}$ in eq. (72) are computed in a similar manner. The variables $\hat{\mathbf{S}}_{\mathbf{xx}}$, $\hat{\mathbf{S}}_{\mathbf{xy}}$, $\hat{\mathbf{S}}_{\mathbf{yx}}$ and $\hat{\mathbf{S}}_{\mathbf{yy}}$ represented in basis $\hat{B}_2$ are first transformed using 1-D backward substitutions to obtain their representation in basis B_2. We have

$$\mathbf{S}_{\mathbf{xx}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)^T\hat{\mathbf{S}}_{\mathbf{xx}} \qquad \mathbf{S}_{\mathbf{xy}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)^T\hat{\mathbf{S}}_{\mathbf{xy}} \tag{78a}$$

$$\mathbf{S}_{\mathbf{yx}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)^T\hat{\mathbf{S}}_{\mathbf{yx}} \qquad \mathbf{S}_{\mathbf{yy}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_2} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_2}\big)^T\hat{\mathbf{S}}_{\mathbf{yy}}. \tag{78b}$$

Then, the boundary vector variables

$$\mathbf{s}^{-o}_{\mathbf{xx}} = \Big(\big(\mathbf{q}^{-}_{\mathbf{x}_2}\big)^T \otimes \mathbf{I}\Big)\mathbf{S}_{\mathbf{xx}} \qquad \mathbf{s}^{o-}_{\mathbf{xy}} = \Big(\mathbf{I} \otimes \big(\mathbf{q}^{-}_{\mathbf{y}_2}\big)^T\Big)\mathbf{S}_{\mathbf{xy}} \qquad \mathbf{s}^{-o}_{\mathbf{yx}} = \Big(\big(\mathbf{q}^{-}_{\mathbf{x}_2}\big)^T \otimes \mathbf{I}\Big)\mathbf{S}_{\mathbf{yx}} \qquad \mathbf{s}^{o-}_{\mathbf{yy}} = \Big(\mathbf{I} \otimes \big(\mathbf{q}^{-}_{\mathbf{y}_2}\big)^T\Big)\mathbf{S}_{\mathbf{yy}} \tag{79a}$$

$$\mathbf{s}^{+o}_{\mathbf{xx}} = \Big(\big(\mathbf{q}^{-}_{\mathbf{x}_2}\big)^T \otimes \mathbf{I}\Big)\mathbf{S}_{\mathbf{xx}} \qquad \mathbf{s}^{o+}_{\mathbf{xy}} = \Big(\mathbf{I} \otimes \big(\mathbf{q}^{-}_{\mathbf{y}_2}\big)^T\Big)\mathbf{S}_{\mathbf{xy}} \qquad \mathbf{s}^{+o}_{\mathbf{yx}} = \Big(\big(\mathbf{q}^{-}_{\mathbf{x}_2}\big)^T \otimes \mathbf{I}\Big)\mathbf{S}_{\mathbf{yx}} \qquad \mathbf{s}^{o+}_{\mathbf{yy}} = \Big(\mathbf{I} \otimes \big(\mathbf{q}^{-}_{\mathbf{y}_2}\big)^T\Big)\mathbf{S}_{\mathbf{yy}}. \tag{79b}$$

are computed and sent to/exchanged with the neighbouring domains/elements. Here, we assume that the domains neighbouring domain Ω in Fig. 2 are oriented similarly. When the neighbouring elements are rotated, the boundary vectors in eq. (79) need to be rotated accordingly. In turn, the partial derivatives are computed in the dual basis $\overset{*}{B}_1$ using

$$\overset{*}{\mathbf{S}}{}^{(1,0)}_{\mathbf{xx}} = -\big(\overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{x}} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{y}}\big)^T\mathbf{S}_{\mathbf{xx}} + \big(\overset{*}{\mathbf{b}}{}^{-}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{-o^{(0)}}_{\mathbf{y}_1}\big)\mathbf{s}^{-o}_{\mathbf{xx}} + \big(\overset{*}{\mathbf{b}}{}^{+}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{+o^{(0)}}_{\mathbf{y}_1}\big)\mathbf{s}^{+o}_{\mathbf{xx}} \tag{80a}$$

$$\overset{*}{\mathbf{S}}{}^{(0,1)}_{\mathbf{xy}} = -\big(\overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{x}} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{y}}\big)^T\mathbf{S}_{\mathbf{xy}} + \big(\overset{*}{\mathbf{D}}{}^{o-^{(0)}}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{-}_{\mathbf{y}_1}\big)\mathbf{s}^{o-}_{\mathbf{xy}} + \big(\overset{*}{\mathbf{D}}{}^{o+^{(0)}}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{+}_{\mathbf{y}_1}\big)\mathbf{s}^{o+}_{\mathbf{xy}} \tag{80b}$$

$$\overset{*}{\mathbf{S}}{}^{(1,0)}_{\mathbf{yx}} = -\big(\overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{x}} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{y}}\big)^T\mathbf{S}_{\mathbf{yx}} + \big(\overset{*}{\mathbf{b}}{}^{-}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{-o^{(0)}}_{\mathbf{y}_1}\big)\mathbf{s}^{-o}_{\mathbf{yx}} + \big(\overset{*}{\mathbf{b}}{}^{+}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{+o^{(0)}}_{\mathbf{y}_1}\big)\mathbf{s}^{+o}_{\mathbf{yx}} \tag{80c}$$

$$\overset{*}{\mathbf{S}}{}^{(0,1)}_{\mathbf{yy}} = -\big(\overset{*}{\mathbf{D}}{}^{(0)}_{\mathbf{x}} \otimes \mathbf{I}\big)^T\big(\mathbf{I} \otimes \overset{*}{\mathbf{D}}{}^{(1)}_{\mathbf{y}}\big)^T\mathbf{S}_{\mathbf{yy}} + \big(\overset{*}{\mathbf{D}}{}^{o-^{(0)}}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{-}_{\mathbf{y}_1}\big)\mathbf{s}^{o-}_{\mathbf{yy}} + \big(\overset{*}{\mathbf{D}}{}^{o+^{(0)}}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \overset{*}{\mathbf{b}}{}^{+}_{\mathbf{y}_1}\big)\mathbf{s}^{o+}_{\mathbf{yy}}. \tag{80d}$$

Finally, the partial derivatives are transformed using 1-D forward substitutions to obtain their representation in basis $\hat{B}_1$. We have:

$$\hat{\mathbf{S}}^{(1,0)}_{\mathbf{xx}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_1}\big)\overset{*}{\mathbf{S}}{}^{(1,0)}_{\mathbf{xx}} \qquad \hat{\mathbf{S}}^{(0,1)}_{\mathbf{xy}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_1}\big)\overset{*}{\mathbf{S}}{}^{(0,1)}_{\mathbf{xy}} \tag{81a}$$

$$\hat{\mathbf{S}}^{(1,0)}_{\mathbf{yx}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_1}\big)\overset{*}{\mathbf{S}}{}^{(1,0)}_{\mathbf{yx}} \qquad \hat{\mathbf{S}}^{(0,1)}_{\mathbf{yy}} = \big(\mathbf{L}^{-1}_{\mathbf{x}_1} \otimes \mathbf{I}\big)\big(\mathbf{I} \otimes \mathbf{L}^{-1}_{\mathbf{y}_1}\big)\overset{*}{\mathbf{S}}{}^{(0,1)}_{\mathbf{yy}}. \tag{81b}$$

Overall, from a computational perspective, the proposed algorithm is very similar to classic finite-difference schemes; because the main part of the computations in eqs (76) and (80) are finite-difference like operations (with different coefficients). The pre-processing steps in eqs (74) and (78) and the post-processing steps in eqs (77) and (81) modestly add up to the computational burden and give the DFD operator an enhanced accuracy. The explicit treatment of the boundary values in eqs (75) and (79) permits to specify boundary conditions easily and to account for sharp discontinuities as well as solid–fluid interfaces, as discussed in Section 4.

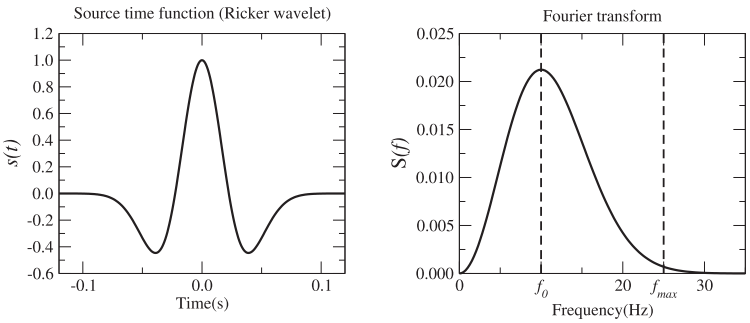

Figure 3. The source–time function (Ricker wavelet) used in the numerical simulations (left-hand panel) and its Fourier transform (right-hand panel). f_0 denotes the peak frequency at which the amplitude of the Fourier transform is maximum and $f_{\max} \approx 2.5 f_0$ is the estimated maximum frequency above which the amplitude of the Fourier transform becomes negligible. In this particular example $f_0 = 10$ Hz and $f_{\max} = 25$ Hz.

Figure		Simulation type	Number of elements	Grid dimension	Δ_t	Points per P-wavelength		Points per S-wavelength
						Solid	Fluid	
Fig. 7	(a)	DFDM	15	168780	8.5×10^{-4} s	12.2		7
	(b)	SEM	10368	166753	8.5×10^{-4} s	12.2		7
	(c)	DFDM	15	42630	1.7×10^{-3} s	6.1		3.5
	(d)	SEM	2592	41905	1.7×10^{-3} s	6.1		3.5
Fig. 10	(a)	DFDM	12	252444	8.5×10^{-4} s	12.2	5.4	7
	(b)	SEM	15552	249841	8.5×10^{-4} s	12.2	5.4	7
	(c)	DFDM	12	64020	1.7×10^{-3} s	6.1	2.7	3.5
	(d)	SEM	3888	62713	1.7×10^{-3} s	6.1	2.7	3.5

Figure 4. Summary of the parameters that take a different value in the simulations in Figs 7 and 10.

4 NUMERICAL EXAMPLES

In this section, numerical examples demonstrating the flexibility and accuracy of the DFD algorithm are presented. Domain decomposition, surface wave modelling and the solid–fluid interface are addressed in separate examples. The following comments apply to all the examples discussed after that:

All the DFD simulations use the algorithm described in Section 3.2. The displacements u_x and u_y are represented in the same basis B_1 defined in eq. (66a). The 1-D bases X_1 and Y_1 are always of the same kind. They can be either one of the left-skewed bases pictured in black in Figs A1 and 1(a)–(c) (with a different N). The stresses σ_{xx}, σ_{yy} and σ_{xy} are represented in the same basis B_2 defined in eq. (66b). The 1-D bases X_2 and Y_2 are always of the same kind as X_1 and Y_1. They can be either one of the right-skewed bases pictured in red in Figs A1 and 1(a)–(c). The usage of the left- and right-skewed basis functions is analogous to the MacCormack-type scheme in the finite-difference method (see e.g. Hixon 1998; Hixon & Turkel 2000; Zhang & Chen 2006). It efficiently prevents grid decoupling phenomena. All the physical properties λ, μ, ρ and the quantities associated with the curvilinear mapping are represented in the same basis H defined in eq. (66c). That basis is a 2-D second-order B-spline basis. The parameters $\alpha_{\pm}^{x/y}$ in eq. (73) have been set as follows:

(i) $\alpha_{\pm}^{x/y} = 0$ at all interfaces surrounding the computational domain to model a free surface.
(ii) $\alpha_{\pm}^{x/y} = 1/2$ at all interfaces separating two domains that are both solid/elastic ($\mu \neq 0$) or fluid/acoustic ($\mu = 0$).
(iii) $\alpha_{\pm}^{x/y} = 0$ inside the solid/elastic domain and $\alpha_{\pm}^{x/y} = 1$ inside the fluid/acoustic domain, at all solid/fluid interfaces.

Other values could be used for the solid–solid and fluid–fluid interfaces. It was observed that taking $\alpha_{\pm}^{x/y} = 0$ or 1 at the solid–fluid interfaces efficiently prevent spurious modes from appearing. As a shortcut, domains' dimensions are given as the number of gridpoints (instead of the number of basis functions or expansion coefficients in the different spatial directions). It makes the comparison with other methods easier. By extension, the average number of expansion coefficients per wavelength is given as the number of gridpoints per wavelength.

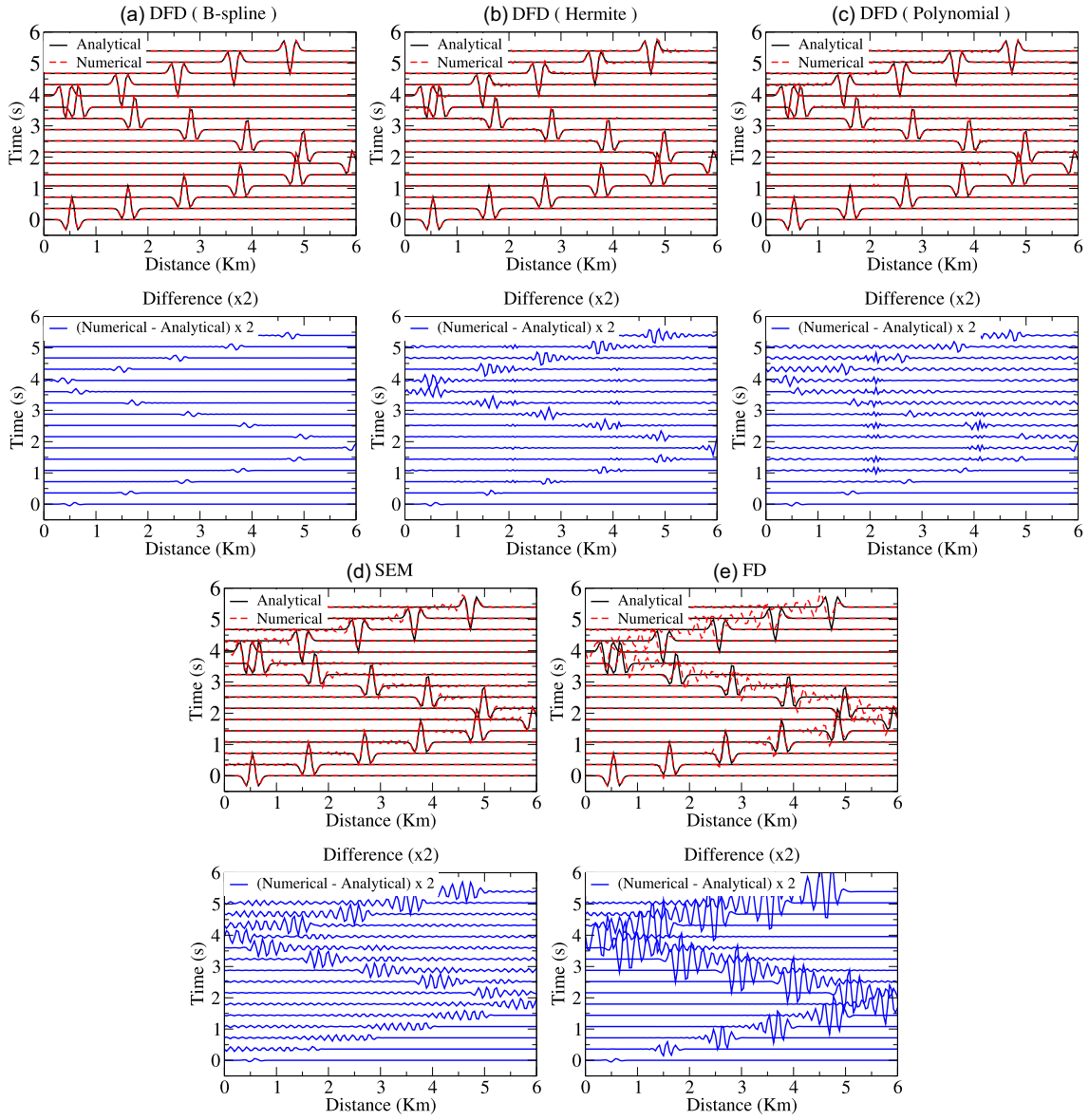

Figure 5. 1-D snapshots (i.e. showing the displacement as a function of distance at a given time) in five different simulations (3 DFD simulations with the fourth-order bases in Figs 1(a)–(c), 1 FD simulation, and 1 SEM simulation). The dashed red lines show the computed wavefield in the top panels, and the solid black lines show the analytical solution. The difference between the numerical and analytical solutions (multiplied by 2) is represented in the bottom panels. Panel (a) shows the result obtained for a DFD simulation using 3 subdomains having the same dimension and the B-spline basis in Fig. 1(c). Panel (b) shows the result obtained for a DFD simulation using 3 subdomains having the same dimension and the Hermite basis in Fig. 1(b). Panel (c) shows the result obtained for a DFD simulation using 3 subdomains having the same dimension and the Polynomial basis in Fig. 1(a). Panel (d) shows the result obtained for an SEM simulation using 39 elements having the same dimension and polynomial order $p = 4$. Panel (e) shows the result obtained for an FD simulation with 4th-order precision in space and 157 gridpoints.

Because the principal focus of the paper is introducing the DFDM algorithm, we adopt a conservative approach and present difference-seismograms when testing the numerical solution against analytical solution and other modelling methods. An in-depth accuracy analysis of DFDM should rely on more advanced techniques (e.g. Kristeková *et al.* 2006, 2009).

We first present numerical examples showing wave propagation in one dimension. The accuracy of DFDM is tested against SEM, FDM and an analytical solution. We then illustrate the flexibility of DFDM for domain decomposition by modelling 2-D wave propagation using elements with different shapes and sizes. The last examples investigate the relative accuracy of DFDM and SEM. The results are compared with analytical solutions. We first discuss surface wave modelling, followed by the case of a solid–fluid interface with and without topography.

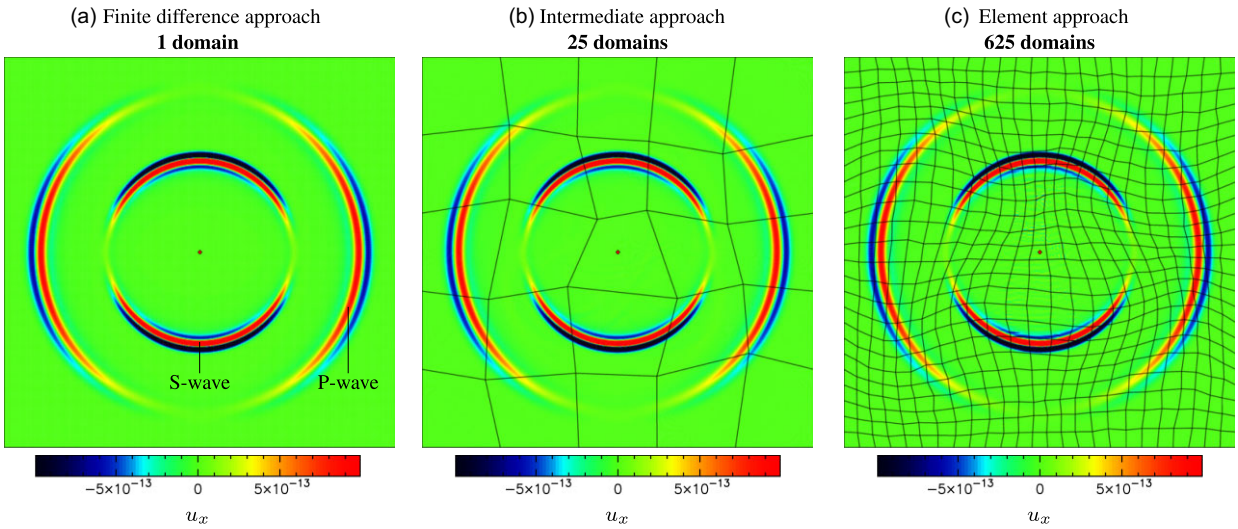

Figure 6. Snapshots showing the horizontal displacement in three different simulations at time $t = 1.4 \times 10^{-4}$. The outermost wave is a *P* wave, while the innermost wave is an *S* wave. The velocity model is a square meter domain with *P*-wave velocity $v_p = 3400\ \mathrm{m\,s^{-1}}$, *S*-wave velocity $v_s = 1963\ \mathrm{m\,s^{-1}}$ and density $\rho = 2500\ \mathrm{kg\,m^{-3}}$. The source is a horizontal point force, and the red diamond marks its location. The source time function is a Ricker wavelet with peak frequency $f_0 = 60$ kHz. The time step is the same for the three simulations, and the grid dimension (i.e. number of degrees of freedom) is almost identical. We have about 5.6 points per *P*-wavelength and 3.3 points per *S*-wavelength. In panel (a), the simulation is performed using a single domain and resembles a finite-difference simulation. The simulation uses the fourth-order polynomial basis. Panel (b) is similar to panel (a) but the velocity model is divided in 25 domains (delimited by the solid black lines). The wavefield is represented using a different basis in each domain. Panel (c) is similar to panel (b) but the velocity model is divided in 625 domains. Note that the three snapshots are visually identical. No numerical dispersion is observed despite the small number of points per -wavelength and the variety of bases used to represent the wavefield in the different domains. These simulations illustrate the ability of the DFDM to adapt/refine the meshing depending on the complexity of the velocity model. In the large and relatively homogeneous region, one can use one or a few big domains. The size of the domains can be adapted, and small elements can be used to mesh complex structures in the velocity model.

4.1 Wave propagation modelling in one dimension

In Fig. 5, 1-D simulations in a bounded homogeneous medium are presented to highlight the amount of numerical dispersion obtained when modelling wave propagation by implementing the DFD algorithm in Section 3.1 with the three bases in Figs 1(a)–(c). A small number of points per wavelength is used to emphasize the dispersion of the different numerical schemes. The results correspond to three simulations based on the DFD algorithm with the bases in Figs 1(a)–(c) plus one FD and one SEM references simulations. The FD simulation is fourth-order accurate in space and second-order accurate in time (Levander 1988).

In all simulations, the length of the velocity model is 6 km. The wave speed is $v = 3000\ \mathrm{m\,s^{-1}}$ and the density is $\rho = 2500\ \mathrm{kg\,m^{-3}}$. The time step $\Delta_t = 3.85 \times 10^{-2}$ s is the same for the five simulations and is small enough to minimize the error associated with time discretization. A Ricker wavelet with peak frequency $f_0 = 10$ Hz is generated by imposing the displacement's initial values. The grid dimension is the same in all simulations; there are 157 gridpoints or basis functions in all simulations. There are 3 points per wavelength at the estimated maximum frequency $f_{\max} = 25$ (Hz). Dirichlet and Neumann boundary conditions are imposed to the left and to the right of the computational domain, respectively. (For the FD simulation, the boundary conditions are imposed using the image method, that is the displacement is assumed to be antisymmetric at the left boundary and symmetric at the right boundary.) The numerical error is estimated by taking the difference between the computed wavefield and an analytical solution.

One first emphasizes that the discontinuous nature of the wavefield is not visible in the DFDM simulations because the jump discontinuities between neighbouring elements are very small.

The DFD algorithm based on the B-spline basis in Fig. 1(c) has the smallest numerical error followed by the DFD algorithm based on the Hermite basis in Fig. 1(b), the DFD algorithm based on the Polynomial basis in Fig. 1(a), the SEM algorithm and the FD algorithm. This is likely due to the higher convergence rate of the B-spline basis. The error is about similar in the DFD simulations based on the Hermite and the Polynomial bases in Fig. 5(b) and in Fig. 5(c). Interestingly, the DFD simulation based on the Polynomial basis in Fig. 5(c) exhibits some slight stationary noise at the interfaces between the DFD domains located around 2 and 4 km. This noise is not present in the DFD simulations based on the Hermite and the B-spline bases. This is attributed to the Polynomial basis's lower continuity and suggests that highly continuous bases such as the B-spline basis are desirable. All the DFD simulations have a smaller numerical error than the SEM and the FD simulations. This suggests that the DFD operators require fewer points per wavelength to achieve an accuracy comparable to the SEM or the FD approaches.

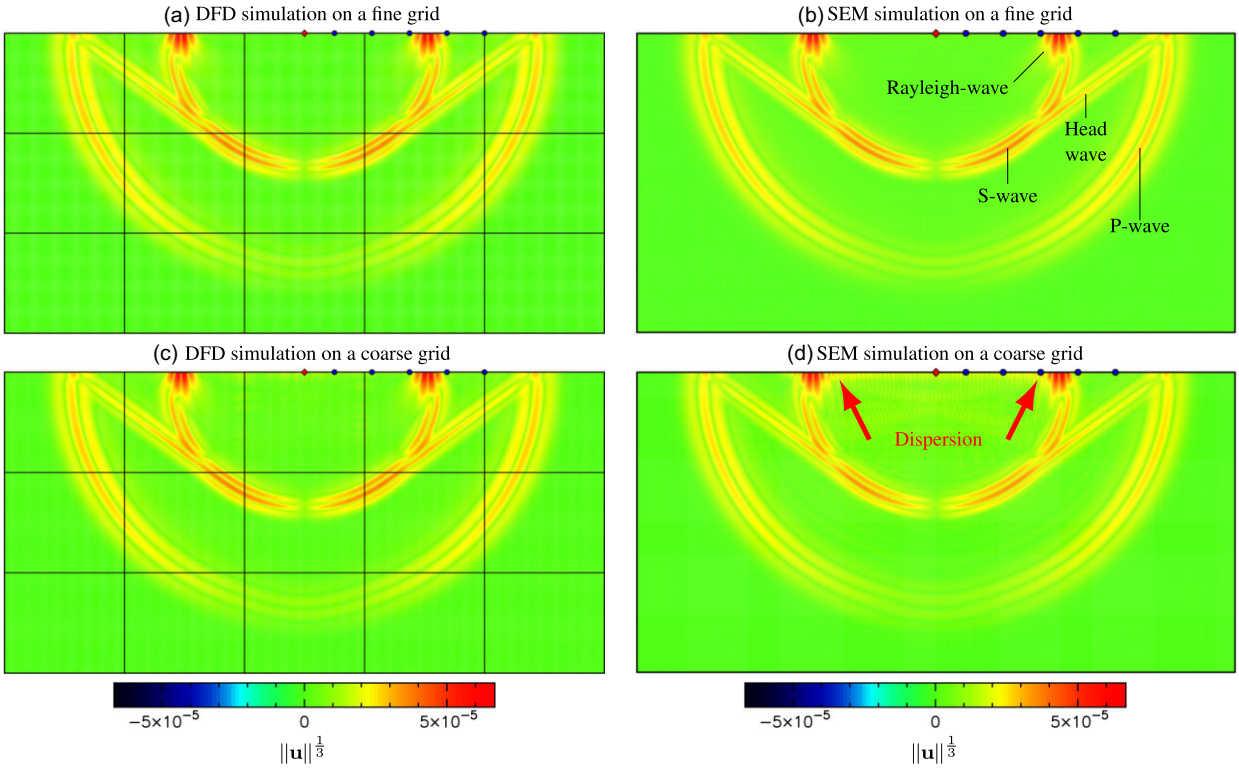

Figure 7. The snapshots show the magnitude of the displacement vector for four different simulations at time $t = 0.85$ s. The magnitude of the displacement is raised to the power 1/3 to enhance waves having a small amplitude. The velocity model has dimensions 6400 m by 3200 m with *P*-wave velocity $v_p = 3400\,\mathrm{m\,s^{-1}}$, *S*-wave velocity $v_s = 1963\,\mathrm{m\,s^{-1}}$ and density $\rho = 2500\,\mathrm{kg\,m^{-3}}$. The waves are generated by an explosive source placed at the position marked by a red diamond. The source time function is a Ricker wavelet with peak frequency $f_0 = 10$ Hz. The receivers are represented using blue circles. The seismograms recorded in the simulations (a, b) and (c, d) are plotted in Figs 8 and 9, respectively. The different seismic waves are indicated in panel (b). Panel (a) shows the wavefield obtained for a DFD simulation with 15 domains. The grid has 168 780 gridpoints. The wavefield is represented with the fourth-order polynomial basis in Fig. 1(a). There are 12.2 points per *P*-wavelength and and 7 points per *S*-wavelength. The time step is $\Delta_t = 8.5 \times 10^{-4}$. Panel (b) is similar to (a) but for a SEM simulation with 10 368 elements and 166 753 gridpoints. The number of points per wavelength and the time step are the same in panels (a) and (b). Panel (c) is similar to panel (a) but the grid is coarser. The grid has 42 630 points. The wavefield is represented using the fourth-order B-spline basis in Fig. 1(c); There are 6.1 points per P-wavelength and and 3.5 points per S-wavelength. The time step is $\Delta_t = 1.7 \times 10^{-3}$. Panel (d) is similar to panel (c) but for a SEM simulation with 2592 elements and 41 905 gridpoints. The number of points per wavelength and the time step are the same in panels (d) and (c). In panels (a) and (b), there are enough points per wavelength. The *P* wave and the *S* wave are modelled accurately. There is no visible numerical dispersion, and the wavefields obtained using DFDM and SEM are identical. In panels (c) and (d), the number of points per wavelength and the time step are divided by two with respect to panels (a) and (b). In panel (d), we see that the SEM simulation exhibits significant numerical dispersion for the surface wave, as indicated by the red arrows. In panel (c), we observe no numerical dispersion with the DFDM, and the wavefield looks identical to panel (a).

4.2 Wave propagation modelling using different element/domain sizes

To highlight the flexibility of the DFD algorithm, three similar simulations with different domain decompositions are presented. The same elastic model and seismic source are used in all simulations while the elements's sizes vary. The results are illustrated in Fig. 6. The velocity model is a 1 m^2 domain with *P*-wave velocity $v_p = 3400\,\mathrm{m\,s^{-1}}$, *S*-wave velocity $v_s = 1963\,\mathrm{m\,s^{-1}}$ and density $\rho = 2500\,\mathrm{kg\,m^{-3}}$. The source time function is a Ricker wavelet similar to that in Fig. 3 with peak frequency $f_0 = 60$ kHz and an estimated maximum frequency $f_{\max} = 150$ kHz. The time step is $\Delta_t = 1.18 \times 10^{-7}$ s.

Fig. 6(a) presents the first simulation. It uses a single domain, and the basis function used is the fourth-order polynomial basis in Fig. 1(a). The grid dimension is $250 \times 250 = 62\,500$. There are 5.6 points per *P*-wavelength and 3.3 points per *S*-wavelength. Note that this is well below the recommended value of five points per wavelength for a spectral element simulation (with polynomial order $p = 4$).

Fig. 6(b) presents the second simulation. The velocity model is divided into 25 domains with random shapes. Each domain uses a different basis chosen randomly (from those in Figs A1 and 1a–c). The grid dimensions of each domain are adapted to have 5.6 points per *P*-wavelength and 3.3 points per *S*-wavelength. The grid dimension is 7629. Most interfaces are non-conformal. The number of basis functions, their type, and polynomial order can be different in neighbouring elements. More precisely, there are three conformal interfaces where the bases exactly match and 37 non-conformal interfaces (i.e. where the basis functions are different on each side of the interface). There are 5, 3, 3, 5, 4 and 5 domains using the basis in Figs A1(a), 1(a), A1(b), 1(b), A1(c) and 1(c), respectively.

Fig. 6(c) presents the third simulation which was designed similarly to the second simulation. The only difference is that the model is divided into 625 elements. The grid dimension is 7472. There are 124 conformal interfaces and 1076 non-conformal interfaces. There are 108, 110, 121, 99, 94 and 93 domains using the bases in Figs A1(a), 1(a), A1(b), 1(b), A1(c) and 1(c), respectively.

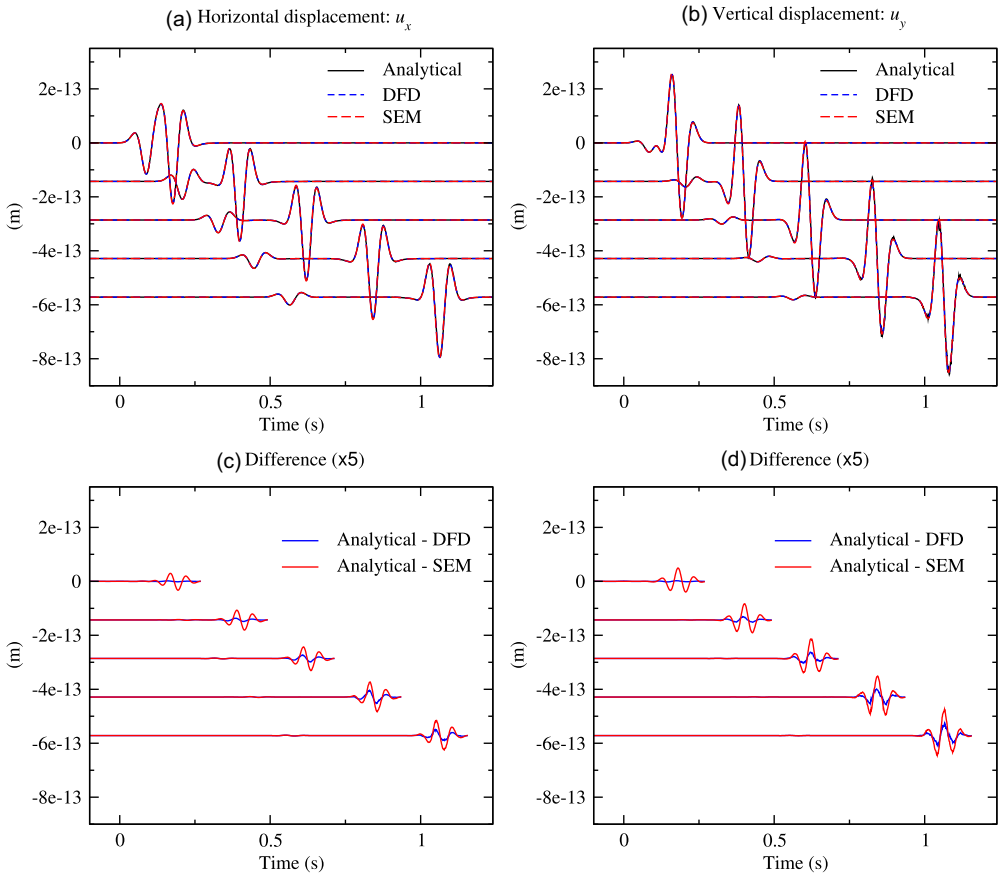

Figure 8. Comparison between the seismograms obtained using DFDM, SEM and the analytical solution when modelling surface wave propagation on a fine grid, as detailed in Figs 7(a) and (b). Panels (a) and (b) show the horizontal and the vertical displacements, respectively. The recordings correspond to the five receivers represented by blues circles in Fig. 7. Panels (c) and (d) show the difference (multiplied by 5) between the computed seismograms and the analytical solution. The differences in panels (c) and (d) are cut when the analytical solution predicts no more waves. As expected, the numerical error increases with the distance to the source because of the accumulation of the numerical dispersion. The maximum difference between the analytical solution and the computed seismograms is about 2.5 and 5 per cent of the signals' amplitude for the DFD and SEM seismograms, respectively.

When comparing the three simulations in Fig. 6, we observe no differences in the modelled wavefield. The DFDM accurately handles the non-conformal interfaces, mixed polynomial orders and multiple basis types. Despite the relatively small number of points per wavelength (for the *S* wave), we observe no visible dispersion; this emphasizes the method's accuracy. The apparent advantage of the DFDM is that the domains' sizes can be adapted depending on the complexity of the velocity model. In large regions where the elastic properties are smoothly varying, one can proceed as in Fig. 6(a), and use one or a few large domains. When needed, one can use much smaller elements as in Fig. 6(c). In this case, we can mesh complex structures and, the DFDM resembles more to the finite/spectral element method. It is also possible to transition from large to small elements using standard mesh refinement techniques.

4.3 Surface wave modelling on a fine grid

In this series of examples, wave propagation in a 2-D homogeneous isotropic half-space model with a free surface is addressed. The results are presented in Figs 7(a), (b) and 8. It is important to note that the DFD algorithm allows to explicitly include the stress conditions at the interface and put the shear velocity to zero in the fluid domain. This is possible because the proposed algorithm is based on the first-order PDEs.

The computational domain has dimensions 6400 m by 3200 m, the *P*-wave velocity is $v_p = 3400\ \mathrm{m\,s^{-1}}$, the *S*-wave velocity is $v_s = 1963\ \mathrm{m\,s^{-1}}$ and the density is $\rho = 2500\ \mathrm{kg\,m^{-3}}$. An explosive source is placed at the centre of the free surface, and the wavefield is recorded at five receivers equally spaced at the surface. The source time function is a Ricker wavelet similar to that in Fig. 3 with peak frequency $f_0 = 10$ Hz and an estimated maximum frequency $f_{\max} = 25$ Hz. The distance from the source to the receivers ranges from 320 to 1920 m.

A DFD simulation has been run for each one of the fourth-order bases in Figs 1(a)–(c), to compare their accuracy. In these simulations, the model is divided into 15 domains with dimensions 1280 m by 1067 m. Each domain contains 116×97 gridpoints. In total, the wavefield is represented using $15 \times 116 \times 97 = 168\,780$ gridpoints. All the interfaces are conformal in these simulations.

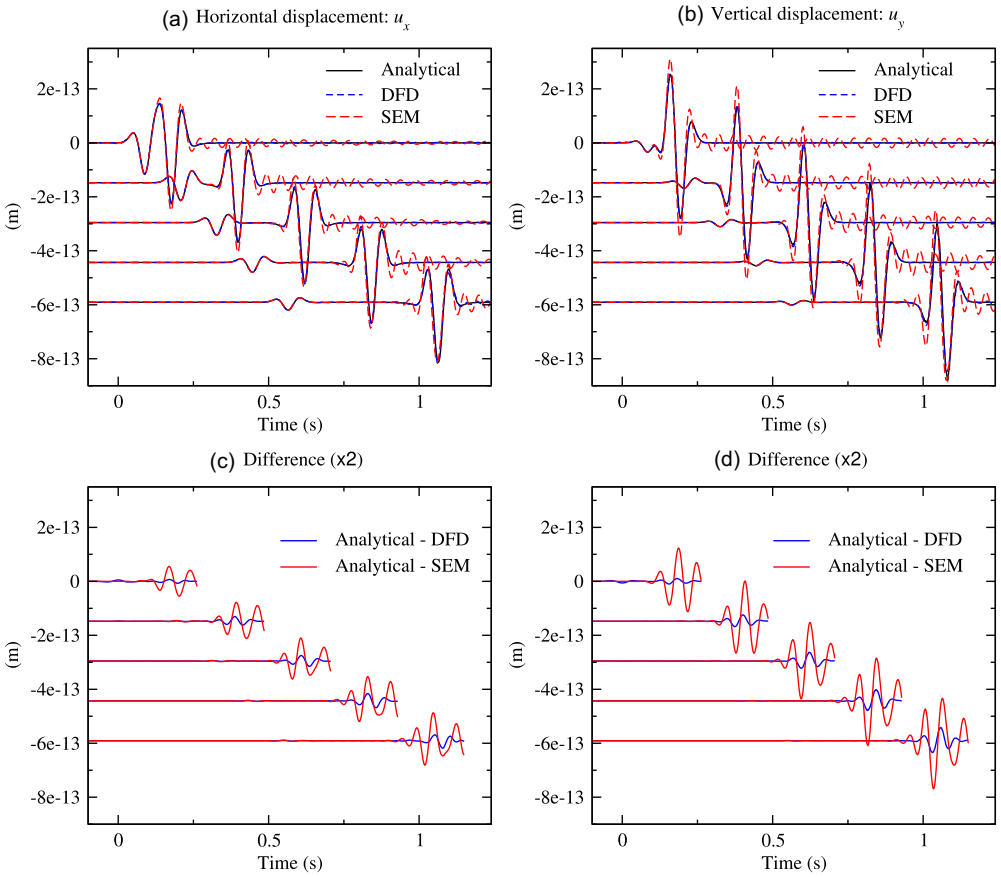

Figure 9. Comparison between the seismograms obtained using the DFDM, SEM and the analytical solution when modelling surface wave propagation on a coarse grid, as detailed in Figs 7(b) and (c). Panels (a) and (b) show the horizontal and the vertical displacements, respectively. The recordings correspond to the five receivers represented by blues circles in Fig. 7. The seismograms obtained using the DFDM (dashed blue lines) match perfectly the analytical solution (solid black line). The seismograms obtained using the SEM (dashed red lines) exhibit some spurious ringing in the surface waves because the number of points per wavelength is insufficient. Panels (c) and (d) show the difference (multiplied by 2) between the computed seismograms and the analytical solution. The difference in panels (c) and (d) are cut when the analytical solution predicts no more waves. The maximum difference between the analytical solution and the computed seismograms is about 8 and 25 per cent of the signals' amplitude for the DFD and SEM seismograms, respectively.

A spectral element simulation has been run using the Specfem code (Komatitsch & Tromp 1999). This simulation divides the model into 144×72 elements with polynomial order 4, producing a grid with 166 753 points.

The number of points per wavelength is similar in the DFD and the SEM simulations. We have 12 points per wavelength for the P wave and 7 points per wavelength for the S wave (these estimates are based on $f_{\max} = 25$ Hz). For a fair comparison, the time step is fixed to $\Delta_t = 8.5 \times 10^{-4}$ s in all simulations.

The three DFD simulations produced nearly indistinguishable results and only the wavefield obtained using the fourth polynomial basis is presented in Fig. 7(a). The wavefield obtained from the SEM simulation is pictured in Fig. 7(b). We observe no visual difference between the two snapshots. The P wave, S wave, Rayleigh-wave and Head-wave emerge as expected, as indicated in Fig. 7(b). One notes no significant numerical dispersion. The seismograms recorded at the five receivers for the DFD and the SEM simulation are shown in Figs 8(a) and (b). These seismograms are compared to an analytical solution evaluated using the Gar6more software (Ezziani *et al.* 2010). This software implements the Cagniard-de Hoop method. The computed seismograms perfectly match the analytical solution because there is a sufficient number of points per wavelength. Not surprisingly, the error pictured in Figs 8(c) and (d) is increasing with distance because of the accumulation of the numerical dispersion. The Rayleigh wave that has the smallest number of points per wavelength is the most affected. Conversely, the P wave is sampled by the largest number of gridpoints shows almost no error. The maximum error observed in these examples is about 5 per cent of the signals' amplitude for the SEM simulation and 2.5 per cent for the DFD simulations. Thus, the DFDM appears to be more accurate than SEM when using the same polynomial order and the same number of points per wavelength.

4.4 Surface wave modelling on a coarse grid

Examples similar to those in the previous section but using a coarser grid are presented. The grid spacing has been divided by two, and the time step was adapted accordingly; we have $\Delta_t = 1.7 \times 10^{-3}$ s. The parameters that are different from the fine-grid case are listed in Fig. 4. In this situation, we expect to observe a lot more numerical dispersion. The results are presented in Figs 7(c), (d) and 9.

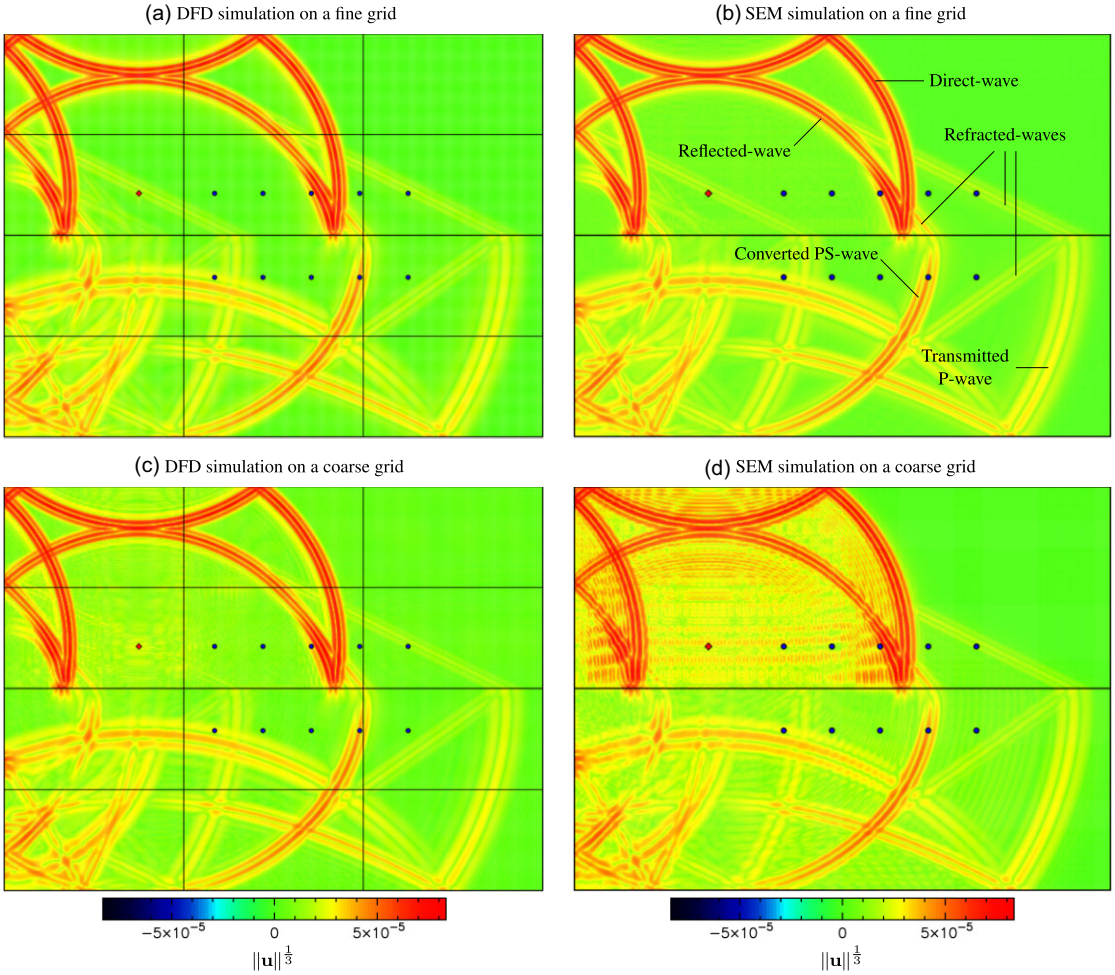

Figure 10. The snapshots show the magnitude of the displacement vector for four different simulations at time $t = 1, 7$ s. The magnitude of the displacement is raised to the power 1/3 to enhance waves with a small amplitude. The velocity model has dimensions 6400 m by 3200 m and is divided into two horizontal layers. The lower layer is a 1600 m high elastic region with *P*-wave velocity $v_p = 3400\ \mathrm{m\,s^{-1}}$, *S*-wave velocity $v_s = 1963\ \mathrm{m\,s^{-1}}$ and density $\rho = 2500\ \mathrm{kg\,m^{-3}}$. The upper layer is a 1600 m high acoustic region with *P*-wave velocity $v_p = 1500\ \mathrm{m\,s^{-1}}$ and density $\rho = 1020\ \mathrm{kg\,m^{-3}}$. The waves are generated by an explosive source placed at the position marked by a red diamond. The source time function is a Ricker wavelet with peak frequency $f_0 = 10$ Hz. The receivers are represented using blue circles and the seismograms recorded in the simulations (a, b) and (c, d) are plotted in Figs 11 and 12, respectively. The principal seismic waves are indicated in panel (b). Panel (a) shows the wavefield obtained for a DFD simulation with 12 domains. The grid has 252444 points. The wavefield is represented using the fourth-order B-spline basis in Fig. 1(a). There are 12.2 points per *P*-wavelength and 7 points per *S*-wavelength in the elastic layer. There are 5.4 points per *P*-wavelength in the acoustic layer. The time step is $\Delta_t = 8.5 \times 10^{-4}$ s. Panel (b) is similar to panel (a) but for a SEM simulation with 15 552 elements and 249 841 gridpoints. The number of points per wavelength and the time step are the same in panels (a) and (b). Panel (c) is similar to panel (a) but is using a coarser grid. The grid size is 64 020 points. the wavefield is represented using the fourth-order B-spline basis in Fig. 1(c). There are 6.1 points per *P*-wavelength and 3.5 points per *S*-wavelength in the elastic layer. There are 2.7 points per *P*-wavelength in the acoustic layer. The time step is $\Delta_t = 1.7 \times 10^{-3}$ s. Panel (d) is similar to panel (c) but for a SEM simulation with 3888 elements and 62 713 gridpoints. The number of points per wavelength and the time step are the same in panels (d) and (c). There are enough points per wavelength in panels (a) and (b) to model wave propagation accurately. No numerical dispersion is observed, and the wavefields obtained using DFDM and SEM are identical. In panels (c) and (d), the number of points per wavelength and the time step are divided by two with respect to panels (a) and (b). In panel (d), we see that the SEM simulation exhibits lots of numerical noise in the acoustic region because there is not enough points per *P*-wavelength. In panel (c), we observe no numerical dispersion with the DFDM, and the wavefield looks identical to panel (a).

In the DFD simulations, the 15 domains have dimensions 58×49 gridpoints and there is $15 \times 58 \times 49 = 42\,630$ gridpoints in total. In the SEM simulation, the model is divided into 72×36 elements. This produces a grid with 41 905 points. There is the same number of points per wavelength in the SEM and the DFD simulations. We have 6.1 points per *P*-wavelength and 3.5 points per *S*-wavelength.

The seismograms obtained for the three DFD simulations are almost identical, and only the results obtained with the B-spline basis are presented. The error in the computed seismograms is more prominent than in the fine grid case because there are fewer points per wavelength. The maximum error is about 25 per cent of the signals' amplitude for the SEM simulation and 8 per cent for the DFD simulations. Most importantly, the increase in the error is more considerable for the SEM simulation. A significant amount of numerical dispersion affects the surfaces wave (as indicated by the red arrows in Fig. 7d). This numerical dispersion appears as high-frequency oscillations in the SEM seismograms (the dashed red lines in Figs 9a and b). On the contrary, the DFD seismograms (the dashed blues lines in Figs 9a and b) match remarkably well the analytical solution.

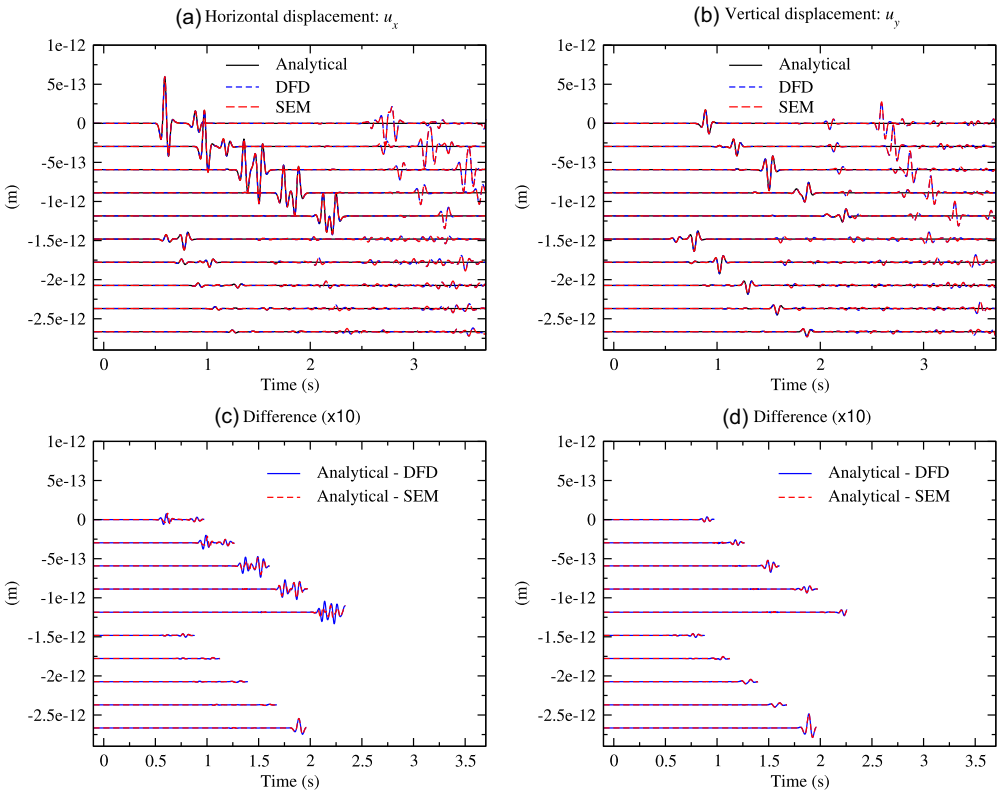

Figure 11. Comparison between the seismograms obtained using DFDM, SEM and the analytical solution when modelling wave propagation near a solid–fluid interface on a fine grid, as detailed in Figs 10(a) and (b). Panels (a) and (b) show the horizontal and the vertical displacements, respectively. The recordings correspond to the ten receivers represented by blues circles in Fig. 10. The first five receivers are placed in the acoustic region, and the last five receivers are placed in the elastic region. Panels (c) and (d) show the difference (multiplied by 10) between the computed seismograms and the analytical solution. The differences in panels (c) and (d) are cut when the analytical solution predicts no more waves. The arrivals observed later in the computed seismograms come from reflections on the free surfaces at the sides of the domain. The maximum difference between the analytical solution and the computed seismograms is about 5 per cent of the signals' amplitude in both simulations.

These examples confirm that the DFDM is more accurate when using the same number of points per wavelength and polynomial order for the basis functions.

4.5 Wave propagation near a fluid–solid interface using a fine grid

A situation that is notoriously delicate to model is when waves propagate through media containing fluid–solid interfaces. In this case, spurious modes may arise, and the modelling algorithms often need to be adapted (see, e.g. Komatitsch *et al.* 2000). The case of two horizontal layers separated by a flat interface (Komatitsch *et al.* 2000) is presented to demonstrate that the DFD is accurate in this delicate situation. The results are shown in Figs 10(a), (b) and 11.

The velocity model has dimensions 6400 m by 3200 m. The bottom layer is a 1600 m high elastic region with P-wave velocity $v_p = 3400\ \mathrm{m\,s^{-1}}$, S-wave velocity $v_s = 1963\ \mathrm{m\,s^{-1}}$ and density $\rho = 2500\ \mathrm{kg\,m^{-3}}$. The top layer is a 1600 m high acoustic region with P-wave velocity $v_p = 1500\ \mathrm{m\,s^{-1}}$ and density $\rho = 1020\ \mathrm{kg\,m^{-3}}$. The waves are generated by an explosive source placed in the acoustic region. The source time function is a Ricker wavelet with peak frequency $f_0 = 10$ Hz and an estimated maximum frequency $f_{\max} = 25$ Hz. The wavefield is recorded using five receivers in the acoustic region and five receivers in the elastic region. The receivers in the acoustic region measure the direct, reflected and refracted/head waves. The receivers in the elastic region record the converted and transmitted waves. Two simulations have been run.

The first simulation is a DFD simulation using the fourth-order B-spline basis in Fig. 1(a). The velocity model is divided into 12 domains with dimensions $193 \times 109 = 21\,037$ gridpoints. The total grid has 252 444 points. The second simulation is using the SEM software Specfem (Komatitsch & Vilotte 1998), the velocity model is divided into $144 \times 108 = 15\,552$ elements with polynomial order 4 and has 249 841 gridpoints in total.

In both simulations, the time step is $\Delta_t = 8.5 \times 10^{-4}$ and the number of points per wavelength is the same. There are 12.2 points per P-wavelength and 7 points per S-wavelength in the elastic layer. There are 5.4 points per P-wavelength in the acoustic layer. In the DFD simulation, the solid–fluid interface is simply handled by setting $\alpha^y_+ = 0$ in eq. (73) inside the three elastic domains neighbouring the interface

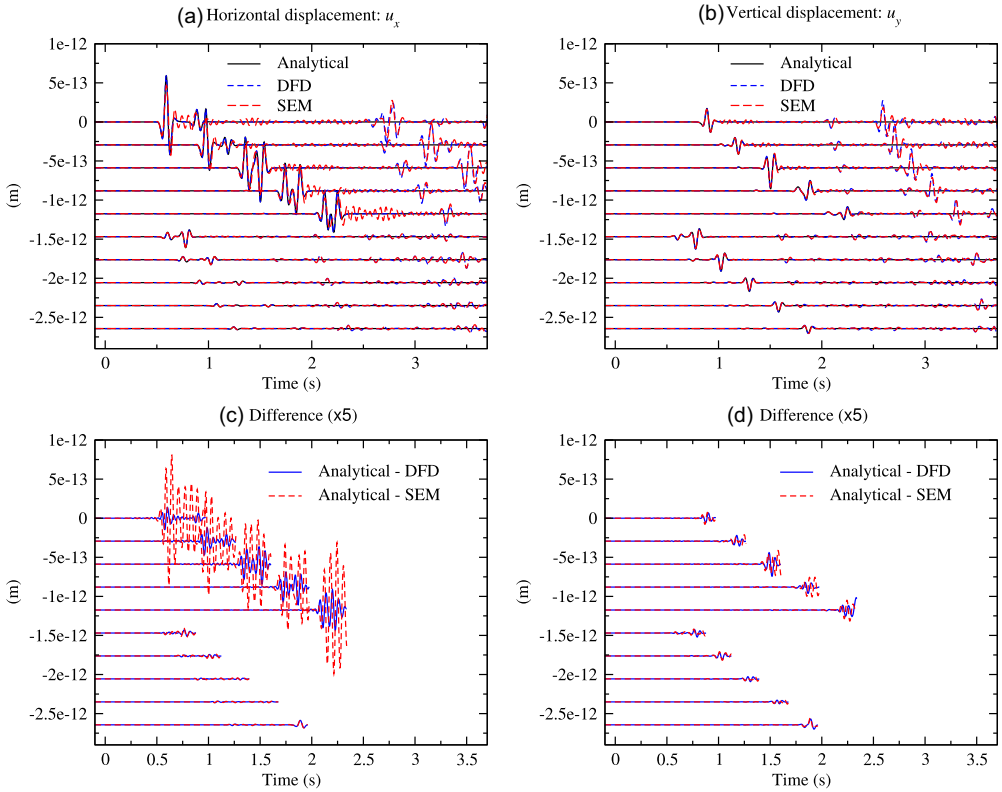

Figure 12. Comparison between the seismograms obtained using DFDM, SEM, and the analytical solution when modelling wave propagation near a solid–fluid interface on a coarse grid, as detailed in Figs 10(b) and (c). Panels (a) and (b) show the horizontal and the vertical displacements, respectively. The recordings correspond to the 10 receivers represented by blues circles in Fig. 10. The first five receivers are placed in the acoustic region, and the last five receivers are placed in the elastic region. The seismograms obtained using the DFDM (dashed blue lines) match the analytical solution (solid black line). The seismograms obtained using SEM (dashed red lines) exhibit some spurious high-frequency noise generated in the acoustic region where the number of points per wavelength is insufficient. Panels (c) and (d) show the difference (multiplied by 5) between the computed seismograms and the analytical solution. The difference in panels (c) and (d) are cut when the analytical solution predicts no more waves. The maximum difference between the analytical solution and the computed seismograms is about 20 per cent of the signals' amplitude for the DFD seismograms and 60 per cent for SEM seismograms.

and $\alpha_{-}^{y} = 1$ inside the three acoustic domains. In the SEM simulation, the solid–fluid interface is handled using the method in Komatitsch *et al.* (2000) where problem is formulated in terms of displacement in elastic regions and a velocity potential in acoustic regions.

In both simulations, we observe no numerical artefact, and no spurious modes emerge. The DFD and the SEM snapshots in Figs 10(a) and (b) are visually identical. The corresponding seismograms in Fig. 11 perfectly agree with the analytical solution (Ezziani *et al.* 2010). The error does not exceed 5 per cent of the signals' amplitude. The error is almost similar for DFD and SEM and is more significant for the receivers placed in the acoustic region because it has fewer points per wavelength. These examples demonstrate that the DFDM accurately models wave propagation in the vicinity of a solid–fluid interface. No modification to the basic algorithm is needed in that situation.

4.6 Wave propagation near a fluid–solid interface using a coarse grid

The following examples are similar to those in the previous section but use a coarser grid to investigate the robustness of the DFDM. All the parameters are identical except the grid dimensions and the time step, as summarized in Fig. 4. The results are presented in Figs 10(c), (d) and 12.

In the DFD simulation, the dimensions of the 15 domains are $97 \times 55 = 5335$ gridpoints. The complete grid has 64 020 gridpoints. In the SEM simulation, the velocity model is divided into $72 \times 54 = 3888$ elements. The grid contains 62 713 points. In both simulations, the time step is $\Delta_t = 1.7 \times 10^{-3}$ s, and the number of points per wavelength is the same. There are 6.1 points per P-wavelength and 3.5 points per S-wavelength in the elastic layer, and 2.7 points per P-wavelength in the acoustic layer.

When comparing the snapshots in Figs 10(c) and (d) to those in Figs 10(a) and (b), we see that the wavefield obtained using the SEM simulation exhibits a significant level of high-frequency noise in the acoustic region where the number of points per wavelength is the smallest. For the DFDM, the snapshot in Fig. 10(c) looks almost similar to that in Fig. 10(a). It emphasizes the superior accuracy of the DFDM when a few points per wavelength are being used. This effect is visible in the seismograms pictured in Fig. 12. The SEM seismograms (plotted using dashed red lines) contain high-frequency oscillations which are not visible in the DFD seismograms (represented using dashed blue lines). These latter match the analytical solution relatively well. In Figs 12(c) and (d), we see that the measured error is about 20 per cent of the

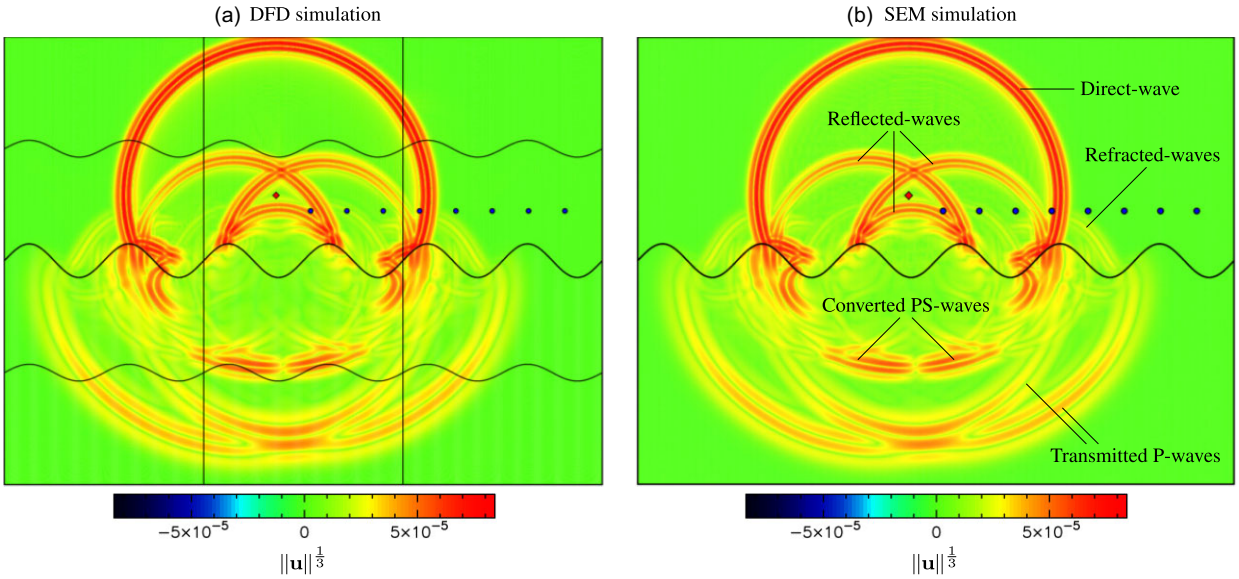

Figure 13. The snapshots show the magnitude of the displacement vector for four different simulations at time $t = 1.2$ s. The magnitude of the displacement is raised to the power 1/3 to enhance waves having a small amplitude. The velocity model is that of (Komatitsch *et al.* 2000). It has dimensions 6400 m by 3200 m and is divided into two layers separated by a sinusoidal interface. The lower layer is an elastic region with P-wave velocity $v_p = 3400\ \mathrm{m\,s^{-1}}$, S-wave velocity $v_s = 1963\ \mathrm{m\,s^{-1}}$ and density $\rho = 2500\ \mathrm{kg\,m^{-3}}$. The upper layer is an acoustic region with P-wave velocity $v_p = 1500\ \mathrm{m\,s^{-1}}$ and density $\rho = 1020\ \mathrm{kg\,m^{-3}}$. The waves are generated by an explosive source placed at the position marked by a red diamond. The source time function is a Ricker wavelet with peak frequency $f_0 = 10$ Hz. The receivers are represented using blue circles The seismograms recorded in the simulations are plotted in Fig. 14. The principal seismic waves observed are indicated in panel (b). Panel (a) shows the wavefield obtained for a DFD simulation with 12 domains. The six domains in the elastic region contain $147 \times 83 = 12\,201$ gridpoints. The six domains in the acoustic region contain $193 \times 109 = 21\,037$ gridpoints. Thus, the grid at the sinusoidal interface is non-conformal. The total grid has 199 428 points. The wavefield is represented using the fourth order polynomial basis in Fig. 1(a). There is 9.3 points per P-wavelength and 5.4 points per S-wavelength in the elastic region. There are 5.4 points per P-wavelength in the acoustic region. The time step is $\Delta_t = 5 \times 10^{-4}$ s. Panel (b) is similar to panel (a) but for a SEM simulation with 15 552 elements and 249 841 gridpoints. There are 12.2 points per P-wavelength and 7.1 points per S-wavelength in the elastic region. There are 5.4 points per P-wavelength in the acoustic region; The grid is conformal at the interface. The time step is the same in panels (a) and (b). The wavefield in panels (a) and (b) look identical.

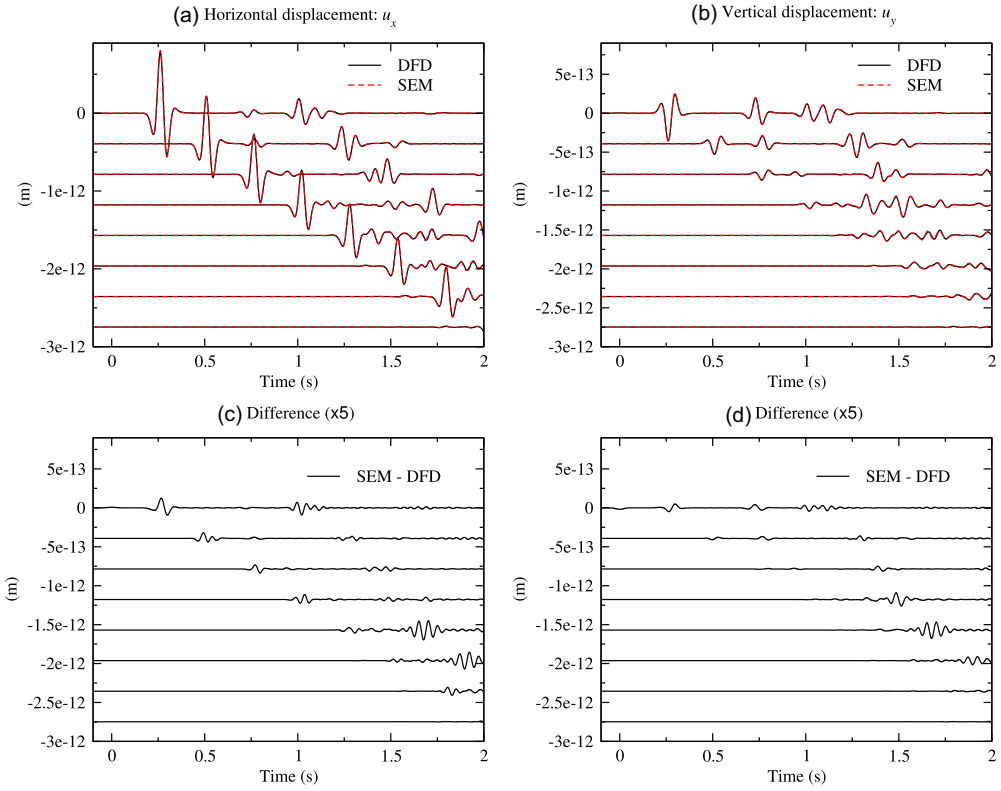

Figure 14. Comparison between the seismograms obtained using DFDM and SEM when modelling wave propagation near a solid–fluid interface with surface topography, as detailed in Figs 13(b) and (c). Panels (a) and (b) show the horizontal and the vertical displacements, respectively. The recordings correspond to the eight receivers represented by blues circles in Fig. 13. The seismograms obtained using the DFDM (solid black lines) match those obtained with SEM (dashed red line). Panels (c) and (d) show the difference (multiplied by 5) between the SEM and DFDM seismograms. The maximum difference between the two sets of seismograms is about 6 per cent of the signals' amplitude, and both methods are in good agreement.

signals' amplitude for the DFDM and goes as high as 60 per cent for SEM. Even though the DFD seismograms are also affected by a small amount of numerical dispersion, all the wave packets observed correspond to the arrival of physical waves propagating through the model.

4.7 Wave propagation near a fluid–solid interface with surface topography

As a tribute to Dimitri Komatitsch, the last example considers wave propagation near a fluid–solid interface with surface topography and is adapted from the famous example in Komatitsch *et al.* (2000). This example is significant to demonstrate the ability of the DFDM to accurately model wave propagation when the domains are deformed, for example to account for curved discontinuities. It also confirms that the DFDM can accurately account for heterogeneity within the domains. Because the same approximation is used to account for the curvilinear mapping and the heterogeneity in the material properties. The results are presented in Figs 13 and 14.

The velocity model is almost similar to the one introduced in the previous section. It has dimensions 6400 m by 3200 m and consists of two horizontal layers separated by a sinusoidal interface. The sinusoidal interface is centred vertically. Its height varies between 1420 and 1780 m, and its periodicity is 1067 m. The properties of the two layers are the same as in the previous example. The source parameters are also identical except its position. The wavefield is recorded using eight receivers in the acoustic region.

A DFD simulation and a SEM simulation have been run. The DFD simulation uses the fourth-order B-spline basis in Fig. 1(a), and the velocity model is divided into 12 domains. The domains have different dimensions in the elastic and acoustic regions. Thus, the mesh is non-conformal at the interface. In the acoustic region, the domains have dimensions $193 \times 109 = 21\,037$ gridpoints, and in the elastic region, the domains have dimensions $147 \times 83 = 12\,201$ gridpoints. The total grid has 199 428 points. There are 9.3 points per P-wavelength and 5.4 points per S-wavelength in the elastic layer and 5.4 points per P-wavelength in the acoustic layer. For the SEM simulation, the velocity model is divided into $144 \times 108 = 15\,552$ elements with polynomial order 4. The grid has 249 841 points. There are 12.2 points per P-wavelength and 7 points per S-wavelength in the elastic layer and 5.4 points per P-wavelength in the acoustic layer. In both simulations the time step is $\Delta_t = 8.5 \times 10^{-4}$.

The snapshots in Fig. 13 show that the presence of the sinusoidal interface produces a complex wavefield where multiple reflected and transmitted waves emerge and pass at the receiver's locations at different times. There is no visual difference between the wavefields computed using DFDM and SEM. The seismograms recorded at the receivers are plotted in Fig. 14 and match perfectly. Since no analytical solution was readily available for an interface with topography, the signals are compared by subtracting the DFD seismograms from the SEM seismograms. The maximum difference does not exceed 6 per cent of the seismograms' amplitude. In this example, both DFDM and SEM give sufficiently accurate results. As discussed in the previous section, other tests using the same configuration but a coarser grid have been performed. These tests led to the same conclusion.

5 DISCUSSION

Prior ending our discussion, let's outline the principal characteristics of DFDM:

(i) The computational domain is decomposed into multiple subdomains or elements similarly to what is done in FEM, SEM or DGM. In this study, rectangular/hexahedral elements are considered. However, the proposed approach could be extended to account for triangular/tetrahedral elements.

(ii) The subdomains can be arbitrarily large, and we may use multiple elements depending on the velocity model. This is different from most element methods, where the element size is usually small and fixed. PSM and SBP-FDM also allow for arbitrary element size. Thus in DFDM, we can do the domain decomposition by hand when a few large elements are used, as in FDM. Using an external meshing tool is also possible whenever complex structures and interfaces need to be represented accurately, as in FEM, SEM or DGM.

(iii) Within a subdomain, the structure of the proposed algorithm is relatively simple and similar to FDM because the DFD operators are constructed from 1-D basis functions having compact support using a tensor product. This makes DFDM efficient, because all the computations rely on 1-D operations involving band diagonal matrices.

(iv) In DFDM, the CPU timescales with the problem's dimension as in FDM and is largely independent of the element size. This is different from spectral methods such as PSM or SEM, for example.

(v) Inside the subdomains, DFDM can account for smooth and discontinuous material heterogeneity. In this work, smooth heterogeneity is incorporated by lumping the mass matrix. The latter is diagonal in DFDM when the material is homogeneous. As opposed to traditional mass lumping in FEM, this does not significantly deteriorate the accuracy of DFDM. It is also possible to adapt some FDM schemes with a subcell resolution to account for internal discontinuities. This approach is treated in a separate publication.

(vi) The free surface can be accounted for naturally in DFDM because the discrete differential operators are derived using a weighted residuals method that is analogous to taking the weak form of the wave equation. Thus we obtain an accuracy similar to FEM, SEM, DGM and SBP-FDM. Surface topography can be accommodated using boundary conforming curvilinear meshes.

(vii) In the case of multidomain modelling, the global linear system does not need to be assembled similarly to FEM, SEM and DGM. The boundary values are exchanged between neighbouring elements, allowing for efficient parallel computations.

(viii) As opposed to FEM and SEM and similarly to DGM, the numerical wavefield is discontinuous. Both the displacement and the stress can be explicitly defined at the surface of the elements. In this respect, DFDM is closer to DGM, which could make it possible to include non-linear interface conditions, arising, for example in dynamic earthquake rupture problems. This is not considered in this paper.

(ix) Nonconformal interfaces, where the basis functions are different in two neighbouring subdomains, are accounted for naturally in DFDM. Thus, the discontinuous grid is a natural feature of DFDM. This includes solid–fluid interfaces that require specific treatment in SEM, for example.

(x) The proposed DFDM algorithm relies on the first-order PDEs and uses the same equations in the fluid and the solid regions. This is similar to FDM on staggered grids. It differs from SEM which is based on the second-order PDEs and uses acoustic and elastic equations in fluid and solid domains, respectively. Acoustic wave propagation is simply achieved by putting the shear velocity to zero in the fluid domains.

(xi) In DFDM, the field variables are represented using orthonormal bases constructed from basis functions having compact support. The orthogonal basis functions are not compactly supported and span the entire width of the elements. This gives the discrete differential operators in DFDM a spectral aspect that is reminiscent of PSM and ensures good accuracy.

(xii) An important aspect of DFDM is that the stress and the displacement components are represented in different bases. This gives some flexibility to annihilate non-physical, parasitic modes. In this paper, left- and right-skewed basis functions are used to mimic the MacCormack-type scheme used in FDM (see e.g. Hixon 1998; Hixon & Turkel 2000; Zhang & Chen 2006). One may also use shifted basis functions analogous to the staggered grids in FDM (see e.g. Yee 1966; Virieux 1986); this is addressed in a separate publication.

(xiii) In this paper, a displacement–stress formulation is used which avoids recording the stress versus time (see e.g. Luo & Schuster 1990). However, the velocity–stress formulation that is necessary for simulating rupture propagation, for example, can also be used and will be detailed in a separate publication.

(xiv) The DFD operators are adjoint, and the wave equation's self-adjoint nature is preserved at the discrete level. This critical property ensures the numerical reciprocity and the stability (subject to the CFL condition) of the proposed algorithm.

The points above put DFDM in perspective with other modelling methods. It should help evaluating the adequacy of DFDM for addressing specific geophysical problems.

6 CONCLUSION

In summary, an explicit time-stepping DFD scheme for modelling wave propagation in heterogeneous elastic media has been introduced. The originality of this work is that the DFD differential operators are derived independently from the governing equation describing wave propagation, using a variational approach. This permits discretizing the wave equation directly from its strong form by substituting the partial derivatives with the DFD differential operators. This is similar to what is done in the finite-difference method.

Adjoint pairs of DFD operators have been constructed and are needed to ensure the numerical stability of the DFD algorithm. Another important aspect of this work is the introduction of an averaging scheme to account for heterogeneity in the elastic properties of the medium. This without increasing the computational complexity of the algorithm. The computational structure of the proposed algorithm is similar to that of the finite-difference method.

Because the modelling domain can be decomposed into domains or elements having arbitrary sizes, the DFDM is very flexible and can be used like a finite-difference method by using a few large elements, or, like a finite/spectral element method where complex structures in the velocity model are meshed using small elements. The proposed formalism also allows representing the wavefield using different basis functions, grid spacing (i.e. non-conformal meshes), and polynomial order in the various domains or elements, which adds to its flexibility.

Finally, the DFDM accurately models the most common situations, including free surfaces and solid–fluid interfaces, and can thus be used to address a wide variety of geophysical problems.

ACKNOWLEDGMENTS

I would like to thank the grant I-SITE E2S-ANR-THERMapp and the France-Berkeley Fund (Project #10-2019) for their support. I wish to thank the editor Sidao Ni as well as Peng Yong and Peter Moczo for their insightful and constructive comments that helped me improve the quality of this paper. I thank Julien Diaz for providing his code and, Jean Virieux and Yann Capdeville for enthusiastic discussions about DFDM.

DATA AVAILABILITY

The data underlying this paper will be shared on reasonable request to the corresponding author (yder_masson@berkeley.edu).

REFERENCES

Albin, N. & Klarmann, J., 2016. An algorithmic exploration of the existence of high-order summation by parts operators with diagonal norm, *J. Sci. Comput.,* **69**(2), 633–650.

Alford, R., Kelly, K. & Boore, D.M., 1974. Accuracy of finite-difference modeling of the acoustic wave equation, *Geophysics,* **39**(6), 834–842.

Alterman, Z. & Karal, F. Jr, 1968. Propagation of elastic waves in layered media by finite difference methods, *Bull. seism. Soc. Am.,* **58**(1), 367–398.

Ampuero, J.P. & Mao, X., 2017. Upper limit on damage zone thickness controlled by seismogenic depth, in *Fault Zone Dynamic Processes: Evolution of Fault Properties During Seismic Rupture,* Chapter 13, Geophysical Monograph Series, Geophysical Monograph 227, pp. 243–253, eds Thomas, M.Y., Mitchell, T.M. & Bhat, H.S., John Wiley & Sons, Inc.

Appelö, D. & Petersson, N.A., 2009. A stable finite difference method for the elastic wave equation on complex geometries with free surfaces, *Commun. Comput. Phys.,* **5**(1), 84–107.

Ba, Z. & Gao, X., 2017. Soil-structure interaction in transversely isotropic layered media subjected to incident plane SH waves, *Shock Vib.,* **2017**(2), 1–3.

Bayliss, A., Jordan, K., LeMesurier, B. & Turkel, E., 1986. A fourth-order accurate finite-difference scheme for the computation of elastic waves, *Bull. seism. Soc. Am.,* **76**(4), 1115–1132.

Bonnasse-Gahot, M., Calandra, H., Diaz, J. & Lanteri, S., 2018. Hybridizable discontinuous Galerkin method for the 2-D frequency-domain elastic wave equations, *Geophys. J. Int.,* **213**(1), 637–659.

Boore, D.M., 1972. Finite difference methods for seismic wave propagation in heterogeneous materials, *Methods Comput. Phys.,* **11,** 1–37.

Bouchon, M. & Coutant, O., 1994. Calculation of synthetic seismograms in a laterally varying medium by the boundary element-discrete wavenumber method, *Bull. seism. Soc. Am.,* **84**(6), 1869–1881.

Bozdağ, E., Peter, D., Lefebvre, M., Komatitsch, D., Tromp, J., Hill, J., Podhorszki, N. & Pugmire, D., 2016. Global adjoint tomography: first-generation model, *Geophys. J. Int.,* **207**(3), 1739–1766.

Breuer, A., Heinecke, A., Rettenberger, S., Bader, M., Gabriel, A.-A. & Pelties, C., 2014. Sustained Petascale Performance of Seismic Simulations with SeisSol on SuperMUC, in *International Supercomputing Conference, ISC 2014. Lecture Notes in Computer Science,* Vol. 8488, pp.1–18, Kunkel, J.M., Ludwig, T. & Meuer, H.W., Springer.

Burstedde, C., Wilcox, L.C. & Ghattas, O., 2011. p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees, *SIAM J. Sci. Comput.,* **33**(3), 1103–1133.

Carcione, J., 1996. A 2D chebyshev differential operator for the elastic wave equation, *Comput. Methods Appl. Mech. Eng.,* **130**(1–2), 33–45.

Carcione, J.M., Kosloff, D. & Kosloff, R., 1988a. Wave-propagation simulation in an elastic anisotropic (transversely isotropic) solid, *Q. J. Mech. Appl. Math.,* **41**(3), 319–346.

Carcione, J.M., Kosloff, D. & Kosloff, R., 1988b. Wave propagation simulation in a linear viscoacoustic medium, *Geophys. J. Int.,* **93**(2), 393–401.

Carpenter, M.H., Nordström, J. & Gottlieb, D., 2010. Revisiting and extending interface penalties for multi-domain summation-by-parts operators, *J. Sci. Comput.,* **45**(1), 118–150.

Chen, G., Peng, Z. & Li, Y., 2022. A framework for automatically choosing the optimal parameters of finite-difference scheme in the acoustic wave modeling, *Comput. Geosci.,* **159,** doi:10.1016/j.cageo.2021.104948.

De Boor, C. & De Boor, C., 1978. *A Practical Guide to Splines,* Vol. **27,** Springer-Verlag New York.

de la Puente, J., Dumbser, M., Käser, M. & Igel, H., 2008. Discontinuous Galerkin methods for wave propagation in poroelastic media, *Geophysics,* **73**(5), T77–T97.

de la Puente, J., Ferrer, M., Hanzich, M., Castillo, J.E. & Cela, J.M., 2014. Mimetic seismic wave modeling including topography on deformed staggered grids, *Geophysics,* **79**(3), T125–T141.

Dormy, E. & Tarantola, A., 1995. Numerical simulation of elastic wave propagation using a finite volume method, *J. geophys. Res.,* **100**(B2), 2123–2133.

Dovgilovich, L. & Sofronov, I., 2015. High-accuracy finite-difference schemes for solving elastodynamic problems in curvilinear coordinates within multiblock approach, *Appl. Numer. Math.,* **93,** 176–194.

Dumbser, M., Käser, M. & De La Puente, J., 2007. Arbitrary high-order finite volume schemes for seismic wave propagation on unstructured meshes in 2D and 3D, *Geophys. J. Int.,* **171**(2), 665–694.

Emmerich, H. & Korn, M., 1987. Incorporation of attenuation into time-domain computations of seismic wave fields, *Geophysics,* **52**(9), 1252–1264.

Étienne, V., Chaljub, E., Virieux, J. & Glinsky, N., 2010. An hp-adaptive discontinuous galerkin finite-element method for 3-D elastic wave modelling, *Geophys. J. Int.,* **183**(2), 941–962.

Ezziani, J.D. *et al.*, 2010. Analytical solution for waves propagation in heterogeneous acoustic/porous media part I: the 2D case, *Commun. Comput. Phys.,* **7**(1), 171–194.

Fan, N., Zhao, L.-F., Gao, Y.-J. & Yao, Z.-X., 2015. A discontinuous collocated-grid implementation for high-order finite-difference modeling, *Geophysics,* **80**(4), T175–T181.

Fichtner, A., 2010. *Full Seismic Waveform Modelling and Inversion,* Springer Science & Business Media.

Fornberg, B., 1987. The pseudospectral method: comparisons with finite differences for the elastic wave equation, *Geophysics,* **52**(4), 483–501.

French, S. & Romanowicz, B.A., 2014. Whole-mantle radially anisotropic shear velocity structure from spectral-element waveform tomography, *Geophys. J. Int.,* **199**(3), 1303–1327.

Gao, L., Brossier, R., Pajot, B., Tago, J. & Virieux, J., 2015. An immersed free-surface boundary treatment for seismic wave simulation, *Geophysics,* **80**(5), T193–T209.

Geller, R.J. & Takeuchi, N., 1998. Optimally accurate second-order time-domain finite difference scheme for the elastic equation of motion: one-dimensional cases, *Geophys. J. Int.,* **135**(1), 48–62.

Graves, R.W., 1996. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences, *Bull. seism. Soc. Am.,* **86**(4), 1091–1106.

Hixon, R., 1998. Evaluation of a high-accuracy maccormack-type scheme using benchmark problems, *J. Comput. Acoust.,* **6**(03), 291–305.

Hixon, R. & Turkel, E., 2000. Compact implicit maccormack-type schemes with high accuracy, *J. Comput. Phys.,* **158**(1), 51–70.

Hughes, T.J., 2012. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis,* Courier Corporation.

Igel, H., Mora, P. & Riollet, B., 1995. Anisotropic wave propagation through finite-difference grids, *Geophysics,* **60**(4), 1203–1216.

Käser, M., Hermann, V. & Puente, J.D.L., 2008. Quantitative accuracy analysis of the discontinuous Galerkin method for seismic wave propagation, *Geophys. J. Int.,* **173**(3), 990–999.

Kelly, K.R., Ward, R.W., Treitel, S. & Alford, R.M., 1976. Synthetic seismograms: a finite-difference approach, *Geophysics,* **41**(1), 2–27.

Koene, E.F., Robertsson, J.O. & Andersson, F., 2020. A consistent implementation of point sources on finite-difference grids, *Geophys. J. Int.,* **223**(2), 1144–1161.

Komatitsch, D. & Tromp, J., 1999. Introduction to the spectral element method for three-dimensional seismic wave propagation, *Geophys. J. Int.,* **139**(3), 806–822.

Komatitsch, D. & Vilotte, J.-P., 1998. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, *Bull. seism. Soc. Am.,* **88**(2), 368–392.

Komatitsch, D., Barnes, C. & Tromp, J., 2000. Wave propagation near a fluid-solid interface: a spectral-element approach, *Geophysics,* **65**(2), 623–631.

Kostin, V., Lisitsa, V., Reshetova, G. & Tcheverda, V., 2015. Local time–space mesh refinement for simulation of elastic wave propagation in multi-scale media, *J. Comput. Phys.,* **281,** 669–689.

Kreiss, H.-O. & Scherer, G., 1974. Finite element and finite difference methods for hyperbolic partial differential equations, in *Mathematical Aspects of Finite Elements in Partial Differential Equations,* pp. 195–212, Elsevier.

Kristek, J., Moczo, P. & Galis, M., 2010. Stable discontinuous staggered grid in the finite-difference modelling of seismic motion, *Geophys. J. Int.,* **183**(3), 1401–1407.

Kristeková, M., Kristek, J., Moczo, P. & Day, S.M., 2006. Misfit criteria for quantitative comparison of seismograms, *Bull. seism. Soc. Am.,* **96**(5), 1836–1850.

Kristeková, M., Kristek, J. & Moczo, P., 2009. Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals, *Geophys. J. Int.,* **178**(2), 813–825.

Levander, A.R., 1988. Fourth-order finite-difference P-SV seismograms, *Geophysics,* **53**(11), 1425–1436.

Li, H., Zhang, W., Zhang, Z. & Chen, X., 2015. Elastic wave finite-difference simulation using discontinuous curvilinear grid with non-uniform time step: two-dimensional case, *Geophys. J. Int.,* **202**(1), 102–118.

Liu, Y., 2013. Globally optimal finite-difference schemes based on least squares, *Geophysics,* **78**(4), T113–T132.

Lo, D.S., 2014. *Finite Element Mesh Generation,* CRC Press.

Lombard, B., Piraux, J., Gélis, C. & Virieux, J., 2008. Free and smooth boundaries in 2-D finite-difference schemes for transient elastic waves, *Geophys. J. Int.,* **172**(1), 252–261.

Luo, Y. & Schuster, G., 1990. Parsimonious staggered grid finite-differencing of the wave equation, *Geophys. Res. Lett.,* **17**(2), 155–158.

Lysmer, J. & Drake, L.A., 1972. A finite element method for seismology, *Methods Comput. Phys.,* **11,** 181–216.

Marfurt, K.J., 1984. Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations, *Geophysics,* **49**(5), 533–549.

Mattsson, K. & Nordström, J., 2004. Summation by parts operators for finite difference approximations of second derivatives, *J. Comput. Phys.,* **199**(2), 503–540.

Mattsson, K. & Olsson, P., 2018. An improved projection method, *J. Comput. Phys.,* **372,** 349–372.

Mittet, R., 2017. On the internal interfaces in finite-difference schemes, *Geophysics,* **82**(4), T159–T182.

Moczo, P. & Kristek, J., 2005. On the rheological models used for time-domain methods of seismic wave propagation, *Geophys. Res. Lett.,* **32**(1), doi:10.1029/2004GL021598.

Moczo, P., Kristek, J., Vavrycuk, V., Archuleta, R.J. & Halada, L., 2002. 3D heterogeneous staggered-grid finite-difference modeling of seismic motion with volume harmonic and arithmetic averaging of elastic moduli and densities, *Bull. seism. Soc. Am.,* **92**(8), 3042–3066.

Moczo, P., Kristek, J. & Gális, M., 2014. *The Finite-Difference Modelling of Earthquake Motions: Waves and Ruptures,* Cambridge Univ. Press.

Moczo, P., Gregor, D., Kristek, J. & de la Puente, J., 2019. A discrete representation of material heterogeneity for the finite-difference modelling of seismic wave propagation in a poroelastic medium, *Geophys. J. Int.,* **216**(2), 1072–1099.

Nie, S., Wang, Y., Olsen, K.B. & Day, S.M., 2017. Fourth-order staggered-grid finite-difference seismic wavefield estimation using a discontinuous mesh interface (WEDMI), *Bull. seism. Soc. Am.,* **107**(5), 2183–2193.

Nissen-Meyer, T., van Driel, M., Stähler, S.C., Hosseini, K., Hempel, S., Auer, L., Colombi, A. & Fournier, A., 2014. Axisem: broadband 3-D seismic wavefields in axisymmetric media, *Solid Earth,* **5**(1), 425–445.

Olsson, P., 1995. Summation by parts, projections, and stability. I, *Math. Comput.,* **64**(211), 1035–1065.

Papageorgiou, A.S. & Pei, D., 1998. A discrete wavenumber boundary element method for study of the 3-D response 2-D scatterers, *Earthq. Eng. Struct. Dyn.,* **27**(6), 619–638.

Reddy, J.N., 2019. *Introduction to the Finite Element Method,* McGraw-Hill Education.

Salih, A., 2016. Weighted residual methods, Department of Aerospace Engineering, Indian Institute of Space Science and Technology, Thiruvananthapuram, December.

Samarskii, A.A., 2001. *The Theory of Difference Schemes,* Vol. **240,** CRC Press.

Seriani, G. & Priolo, E., 1994. Spectral element method for acoustic wave simulation in heterogeneous media, *Finite Elem. Anal. Des.,* **16**(3–4), 337–348.

Shragge, J. & Konuk, T., 2020. Tensorial elastodynamics for isotropic media, *Geophysics,* **85**(6), T359–T373.

Smith, W.D., 1975. The application of finite element analysis to body wave propagation problems, *Geophys. J. Int.,* **42**(2), 747–768.

Strand, B., 1994. Summation by parts for finite difference approximations for d/dx, *J. Comput. Phys.,* **110**(1), 47–67.

Sun, C., Yang, Z.-L., Jiang, G.-X.-X. & Yang, Y., 2020. Multiblock SBP-SAT methodology of symmetric matrix form of elastic wave equations on curvilinear grids, *Shock Vib.,* **2020**(1), doi:10.1155/2020/8401537.

Sun, Y.-C., Zhang, W. & Chen, X., 2016. Seismic-wave modeling in the presence of surface topography in 2D general anisotropic media by a curvilinear grid finite-difference method, *Bull. seism. Soc. Am.,* **106**(3), 1036–1054.

Takeuchi, N. & Geller, R.J., 2000. Optimally accurate second order time-domain finite difference scheme for computing synthetic seismograms in 2-D and 3-D media, *Phys. Earth planet. Inter.,* **119**(1–2), 99–131.

Tape, C., Liu, Q., Maggi, A. & Tromp, J., 2009. Adjoint tomography of the southern California Crust, *Science,* **325**(5943), 988–992.

Thompson, J.F., Warsi, Z. & Mastin, C.W., 1982. *Numerical Grid Generation,* Vol. **1982,** North-Holland.

Uphoff, C., Rettenberger, S., Bader, M., Madden, E.H., Ulrich, T., Wollherr, S. & Gabriel, A.-A., 2017. Extreme scale multi-physics simulations of the Tsunamigenic 2004 Sumatra megathrust earthquake, in *Proceedings of the SC '17: International Conference for High Performance Computing, Networking, Storage and Analysis,* Denver, CO, USA, 12–17 November 2017, pp. 1–16, Association for Computing Machinery.

Virieux, J., 1986. P-sv wave propagation in heterogeneous media: velocity-stress finite-difference method, *Geophysics,* **51**(4), 889–901.

Virieux, J. & Operto, S., 2009. An overview of full-waveform inversion in exploration geophysics, *Geophysics,* **74**(6), WCC1–WCC26.

Virieux, J., Calandra, H. & Plessix, R.-É., 2011. A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging, *Geophys. Prospect.,* **59**(5), Special issue: Modelling Methods for Geophysical Imaging: Trends and Perspectives, 794–813.

Warburton, T., 2013. A low-storage curvilinear discontinuous Galerkin method for wave problems, *SIAM J. Sci. Comput.,* **35**(4), A1987–A2012.

Weng, H. & Ampuero, J.-P., 2019. The dynamics of elongated earthquake ruptures, *J. geophys. Res.,* **124**(8), 8584–8610.

Weng, H. & Ampuero, J.-P., 2020. Continuum of earthquake rupture speeds enabled by oblique slip, *Nat. Geosci.,* **13**(12), 817–821.

Wilcox, L.C., Stadler, G., Burstedde, C. & Ghattas, O., 2010. A high-order discontinuous Galerkin method for wave propagation through coupled elastic–acoustic media, *J. Comput. Phys.,* **229**(24), 9373–9396.

Yee, K., 1966. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antenn. Propagat.,* **14**(3), 302–307.

Zahradník, J., 1995. Simple elastic finite-difference scheme, *Bull. seism. Soc. Am.,* **85**(6), 1879–1887.

Zang, N., Zhang, W. & Chen, X., 2021. An overset-grid finite-difference algorithm for simulating elastic wave propagation in media with complex free-surface topography, *Geophysics,* **86**(4), T277–T292.

Zhang, W. & Chen, X., 2006. Traction image method for irregular free surface boundaries in finite difference seismic wave simulation, *Geophys. J. Int.,* **167**(1), 337–353.

Zhang, W., Shen, Y. & Zhao, L., 2012a. Three-dimensional anisotropic seismic wave modelling in spherical coordinates by a collocated-grid finite-difference method, *Geophys. J. Int.,* **188**(3), 1359–1381.

Zhang, W., Zhang, Z. & Chen, X., 2012b. Three-dimensional elastic wave numerical modelling in the presence of surface topography by a collocated-grid finite-difference method on curvilinear grids, *Geophys. J. Int.,* **190**(1), 358–378.

Zhang, W., Zhang, Z., Li, M. & Chen, X., 2020. Gpu implementation of curved-grid finite-difference modelling for non-planar rupture dynamics, *Geophys. J. Int.,* **222**(3), 2121–2135.

Zhang, Z., Zhang, W. & Chen, X., 2014. Three-dimensional curved grid finite-difference modelling for non-planar rupture dynamics, *Geophys. J. Int.,* **199**(2), 860–879.

Zhang, Z., Huang, H., Zhang, W. & Chen, X., 2016. On the free-surface problem in dynamic-rupture simulation of a nonplanar fault, *Bull. seism. Soc. Am.,* **106**(3), 1162–1175.

Zhou, H., Liu, Y. & Wang, J., 2021. Elastic wave modeling with high-order temporal and spatial accuracies by a selectively modified and linearly optimized staggered-grid finite-difference scheme, *IEEE Trans. Geosci. Remote Sens.,* **60,**.

APPENDIX: CONSTRUCTION OF THE BASIS FUNCTIONS

In this section, I detail the construction of the three different kinds of bases used to represent the wavefield and construct the DFD operators. The three bases kinds are piecewise polynomials bases where the basis functions have compact support. Using basis function with compact support is crucial to building up efficient DFD algorithms because we obtain band diagonal matrices from eq. (10). Also, when representing functions with the three bases considered, the end values to the left and the right only depend on the first and the last basis functions, respectively. In other words, the basis functions are zero at the limits of the definition interval except for the first one and the last one. This makes it easier to impose the boundary condition when modelling wave propagation. The first basis kind is a simple polynomial basis; it is based on polynomial interpolation, its basis functions are continuous with discontinuous first derivative (C^0 continuity). The second basis kind is a cubic Hermite basis; it is constructed using Hermite interpolation, its basis functions, and their first derivative are continuous (C^1 continuity). The third basis kind is a B-spline basis; its basis functions and their first few derivatives are continuous (C^{p-1} continuity). Both the polynomial and the Hermite bases are interpolating. The expansion coefficients representing a function in those bases are the values of the function at some interpolation points. The B-spline basis is not interpolating. All the bases considered have the partition of unity property.

A1 Polynomial basis

The polynomial basis is constructed from the piecewise interpolating function

$$g(x) = g(\mathbf{x}, \mathbf{y}, \mathbf{k}, x) \tag{A1}$$

with polynomial order p defined on the interval $[a, b]$. The function $g(x)$ takes a set of N points $\mathbf{x} = [x_1, \ldots, x_N]$ (with $a \leq x_i < x_{i+1} \leq b$) together with their attached data $\mathbf{y} = [y_1, \ldots, y_N]$ and a knot vector $\mathbf{k} = [k_1, k_2, \ldots, k_{N-p+1}]$ (with $a \leq k_i < k_{i+1} \leq b$) as an input and returns the interpolated value $g(x) = y(x)$ of the data $\mathbf{y}$ at point x. Within each knot interval $[k_i, k_{i+1}]$ the interpolating function takes the value of the interpolating polynomial $P_i(x)$ passing by the subset of data points with coordinates $[(x_i, y_i), \ldots, (x_{i+p}, y_{i+p})]$. We have:

$$g(x) = P_i(x) \quad \begin{cases} k_i \leq x < k_{i+1} \\ 1 \leq i \leq N - p \end{cases}. \tag{A2}$$

The polynomial $P_i(x)$ may be evaluated, for example, using Lagrange interpolation. Provided that the interpolating function $g(x)$ can be computed, a family of basis functions B_i $(i = 1, N)$ can be constructed by interpolating the discrete vector basis

$$\mathbf{e}_1 = \begin{vmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{vmatrix}, \mathbf{e}_2 = \begin{vmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{vmatrix}, \ldots, \mathbf{e}_N = \begin{vmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{vmatrix}, \tag{A3}$$

we have

$$B_i(x) = g(\mathbf{x}, \mathbf{e}_i, \mathbf{k}, x) \; \forall \; i = 1, N. \tag{A4}$$

In this study, all the examples are computed using evenly spaced points, that is we have $x_i = a + \frac{(i-1)(b-a)}{(N-1)}$ for $i = 1, N$. Further, to ensure continuity of the basis functions B_i, the knot vector $\mathbf{k}$ is taken as a subset of the point vector $\mathbf{x}$. We have:

$$\mathbf{k} = \left[k_1 = x_1, k_2 = x_l, k_3 = x_{l+1}, \quad \ldots \quad , k_{N-p} = x_{N+l-p-2}, k_{N-p+1} = x_N\right], \tag{A5}$$

where the shift parameter l allows to control the degree of symmetry or the skewness of the basis functions B_i. The second order bases obtained with parameters $p = 2, l = 2, N = 17$ and $p = 2, l = 3, N = 17$ are pictured in Fig. A1(a) in black and red, respectively, and, the fourth order bases obtained with parameters $p = 4, l = 3, N = 17$ and $p = 4, l = 4, N = 17$ are pictured in Fig. 1(a) in black and red, respectively.

A2 Cubic Hermite basis

The cubic Hermite basis is constructed from the Hermite piecewise interpolating function

$$h(x) = h(\mathbf{x}, \mathbf{y}, \mathbf{y}', x) \tag{A6}$$

with polynomial order $p = 3$ defined on the interval $[a, b]$. The function $h(x)$ takes a set of N points $\mathbf{x} = [x_1, \ldots, x_N]$ (with $a \leq x_i < x_{i+1} \leq b$) and their attached data $\mathbf{y} = [y_1, \ldots, y_N]$ together with their derivative $\mathbf{y}' = [y'_1, y'_2, \ldots, y'_N]$ as an input and returns the interpolated value $h(x) = y(x)$ of the data $\mathbf{y}$ at point x. Within each interval $[x_i, x_{i+1}]$ we have

$$\begin{aligned} h(x) = {} & h_{00}(t)\mathbf{y}_i \quad + h_{10}(t)(x_{i+1} - x_i)\mathbf{y}'_i \\ & + h_{01}(t)\mathbf{y}_{i+1} + h_{11}(t)(x_{i+1} - x_i)\mathbf{y}'_{i+1}, \end{aligned} \tag{A7}$$

where

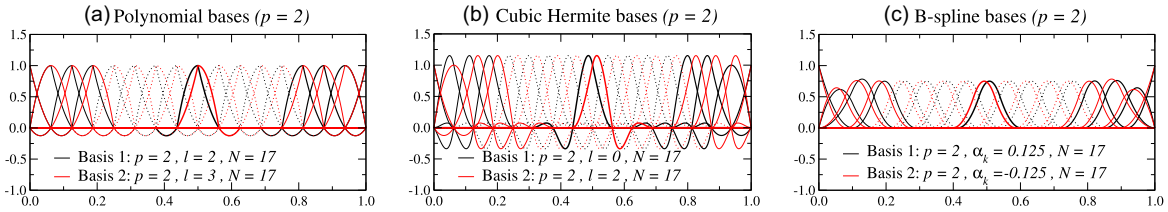

Figure A1. The three primary kinds of bases used to construct the DFD operators. (a) The black and red lines represent the basis functions of two distinct polynomial bases (1 and 2, respectively) constructed following the approach in Section A1 with polynomial order $p = 2$ and $N = 17$; they are continuous with discontinuous first derivative. The basis functions in black are left-skewed and have been constructed using the shift parameter $l = 2$. The basis functions in red are right-skewed and have been constructed using the shift parameter $l = 3$. In this particular example, the basis functions in red overlap the black ones when flipping their left- and right-hand sides. When modelling wave propagation, the displacement is represented using the left-skewed basis while the stress is represented using the right-skewed basis, for example, to prevent spurious modes from appearing. (b) Same as in (a) but using the Hermite bases described in Section A2, the basis functions in black and red have been constructed using the shift parameters $l = 0$ and $l = 2$, respectively, they are continuous with continuous first derivative. (d) Same as in (a) but using the B-spline bases described in Section A3, the basis functions in black and red have been constructed using the knot shift parameters $\alpha_k = 1/8$ and $\alpha_k = -1/8$, respectively, they are continuous as well as their first derivatives.

$$h_{00} = (1 + 2t)(1 - t)^2 \tag{A8}$$

$$h_{10} = t(1 - t)^2 \tag{A9}$$

$$h_{01} = t^2(3 - 2t) \tag{A10}$$

$$h_{11} = t^2(t - 1) \tag{A11}$$

and

$$t = \frac{x - x_i}{x_{i+1} - x_i}. \tag{A12}$$

In this study, the derivatives in vector $\mathbf{y}'$ are computed from the vectors $\mathbf{y}$ and $\mathbf{x}$ using

$$\mathbf{y'}_i = \begin{cases} P'_1(x_i) & \text{if} \quad i - l < 1 \\ P'_{i-l}(x_i) & \text{if} \quad 1 \leq i - l \leq N - p \\ P'_{N-p}(x_i) & \text{if} \quad N - p < i - l \end{cases}, \tag{A13}$$

where $P'_i(x)$ is the the derivative of the polynomial $P_i(x)$ passing by the subset of data points with coordinates $[(x_i, y_i), \ldots, (x_{i+p}, y_{i+p})]$. As for the polynomial basis, the shift parameter l allows to control the skewness of the basis functions H_i. Note that the order p of the polynomials $P_i(x)$ can be greater than that of the Hermite polynomials h_{ij} ($p = 3$). When increasing the order of $P_i(x)$ the basis functions away from the interval's boundaries will converge toward the ideal interpolant $\mathrm{sinc}(x)$, this allows to reduce the computational error even though the polynomial order of the basis function is constant ($p = 3$). As for the polynomial basis, a set of basis functions H_i ($i = 1, N$) are constructed by interpolating the discrete vector basis in eq. (A3). We have:

$$H_i(x) = h(\mathbf{x}, \mathbf{e}_i, \mathbf{e'}_i, x) \; \forall \; i = 1, N, \tag{A14}$$

where $\mathbf{e'}_i$ is computed from $\mathbf{x}_i$ and $\mathbf{e}_i$ using eq. (A13). The second order bases obtained with parameters $p = 2$, $l = 0$, $N = 17$ and $p = 2$, $l = 2$, $N = 17$ are pictured in Fig. A1(b) in black and red, respectively, and, the fourth order bases obtained with parameters $p = 4$, $l = 1$, $N = 17$ and $p = 4$, $l = 3$, $N = 17$ are pictured in Fig. 1(b) in black and red, respectively.

A3 B-spline basis

Given the knot vector with $p + 1$ repeated end knots

$$\begin{aligned} \mathbf{k} = &[k_1 = k_2 = \cdots = k_{p+1}, k_2, \ldots, \\ &k_N, k_{N+1} = k_{N+2} = \cdots = k_{N+p+1}]' \end{aligned} \tag{A15}$$

where $a \leq k_i < k_{i+1} \leq b$, the B-spline basis functions $B_i^p(x)$ with polynomial order p can be computed on the interval $[a, b]$ using the Cox de Boor recursion formula (De Boor & De Boor 1978)

$$B_i^0(x) = \begin{cases} 1 \text{ if } k_i < x < k_{i+1} \\ 0 \text{ otherwise} \end{cases} \tag{A16}$$

$$B_i^p(x) = \frac{x - k_i}{k_{i+p} - k_i} B_i^{p-1}(x) + \frac{k_{i+p+1} - x}{k_{i+p+1} - k_{i+1}} B_{i+1}^{p-1}(x). \tag{A17}$$

The B-spline basis functions $B_i^p(x)$ have C^{p-1} continuity when the inner knots are all different (i.e. there is no repeated knots). In this study, we construct the B-spline basis using the shifted knot vector $\mathbf{k}^\alpha$ with components

$$k_i^\alpha = k_i^0 \qquad \textbf{if} \qquad 1 < i < p+1 \tag{A18}$$

$$k_i^\alpha = \left\{ \begin{array}{l} (1-\alpha_k)k_i^0 + \alpha_k k_{i+1}^0 \text{ if } \alpha_k > 0 \\ (1-\alpha_k)k_i^0 + \alpha_k k_{i-1}^0 \text{ if } \alpha_k < 0 \end{array} \right\} \qquad \textbf{if} \qquad p+2 < i < N \tag{A19}$$

$$k_i^0 = k_i^0 \qquad \textbf{if} \qquad N+1 < i < N+p+1, \tag{A20}$$

where the knot shift parameter α_k produces basis function with a right shift when $\alpha_k > 0$ and a left shift when $\alpha_k < 0$. The components of the zero shift knot vector $\mathbf{k}^0$ are

$$k_i^\alpha = a \qquad \textbf{if} \qquad 1 < i < p+1 \tag{A21}$$

$$k_i^0 = a + \frac{(b-a)}{2}\left[1 - \frac{N+p-2(i-1)}{N-1}\right] \qquad \textbf{if} \qquad p+2 < i < N \tag{A22}$$

$$k_i^0 = b \qquad \textbf{if} \qquad N+1 < i < N+p+1. \tag{A23}$$

The second order B-spline bases obtained with parameters $p = 2$, $\alpha_k = 1/8$, $N = 17$ and $p = 2$, $\alpha_k = -1/8$, $N = 17$ are pictured in Fig. A1(c) in black and red, respectively, and, the fourth order B-spline bases obtained with parameters $p = 4$, $\alpha_k = 1/8$, $N = 17$ and $p = 4$, $\alpha_k = -1/8$, $N = 17$ are pictured in Fig. 1(c) in black and red, respectively.