

## 3D numerical simulation of elastic waves with a frequency-domain iterative solver

Mikhail Belonosov<sup>1</sup>, Victor Kostin<sup>2</sup>, Dmitry Neklyudov<sup>2</sup>, and Vladimir Tcheverda<sup>2</sup>

### ABSTRACT

The efficiency of any inversion method for estimating the medium parameters from seismic data strongly depends on simulation of the wave propagation, i.e., forward modeling. The requirements are that it should be accurate, fast, and computationally efficient. When the inversion is carried out in the frequency domain (FD), e.g., FD full-waveform inversion, only a few monochromatic components are involved in the computations. In this situation, FD forward modeling is an appealing potential alternative to conventional time-domain solvers. Iterative FD solvers, based on a Krylov subspace iterative method, are of interest due to their moderate memory requirements compared with direct solvers. A huge issue preventing their

successful use is a very slow convergence. We have developed an iterative solver for the elastic wave propagation in 3D isotropic heterogeneous land models. Its main ingredient is a novel preconditioner, which provides the convergence of the iteration. We have developed and justified a method to invert our preconditioner effectively on the base of the 2D fast Fourier transform and solving a system of linear algebraic equations with a banded matrix. In addition, we determine how to parallelize our solver using the conventional hybrid parallelization (MPI in conjunction with OpenMP) and demonstrate the good scalability for the widespread 3D SEG/EAGE overthrust model. We find that our method has a high potential for low-frequency simulations in land models with moderate lateral variations and arbitrary vertical variations.

### INTRODUCTION

Numerical simulation of seismic wave propagation is the backbone of many technologies being developed in exploration geophysics to recover the reflectivity and the velocity model of the subsurface. For instance, applied several times at each iteration of full-waveform inversion (FWI) (Pratt, 1999; Symes, 2008; Virieux et al., 2009), it is greatly responsible for the total computational time and the accuracy of an output velocity.

Most of current successful solutions for 3D large-scale numerical simulation applied in the industry are time-domain methods (Sirgue et al., 2007; Etienne et al., 2014; Kostin et al., 2015; Lisitsa et al., 2016). Advances in computational technologies, known as the “supercomputing era,” gave an impulse for development of an alternative approach — frequency-domain (FD) methods. For FWI applications, when only several frequencies are needed, they may

be a good option for forward modeling. Discretizing the wave equation by finite-difference or finite-element approximations creates a system of linear algebraic equations (SLAE). One approach is to solve these equations by a direct method (Operto et al., 2007). For 3D problems, the main bottleneck is the hundreds of gigabytes of memory required, even for acoustic media, making it unrealistic to be applied in production. Attempts to resolve this issue include applying data compression techniques based on the block low-rank approach, hierarchically semiseparable formats of storing data, and low-rank approximation of matrices (Wang et al., 2012; Weisbecker et al., 2013; Kostin et al., 2017).

An alternative to the direct approach is an iterative approach based on a Krylov-type iterative method (Saad, 2003). Its memory requirements are much more modest. However, the indefiniteness of the coefficient matrix in seismic applications leads to slow convergence. As a remedy, an appropriate preconditioner might be

Manuscript received by the Editor 27 October 2017; revised manuscript received 19 June 2018; published ahead of production 30 August 2018; published online 23 October 2018.

<sup>1</sup>Aramco Overseas Company B.V., Aramco Research Center — Delft, Informaticaalaan 6, Delft 2628 ZD, the Netherlands. E-mail: mikhail.belonosov@aramcooverseas.com.

<sup>2</sup>Trofimuk Institute of Petroleum Geology and Geophysics SB RAS, 3, Koptyug Ave, Novosibirsk 630090, Russia. E-mail: kostinv@ipgg.sbras.ru; dmitn@mail.ru; cheverdava@ipgg.sbras.ru.

© 2018 Society of Exploration Geophysicists. All rights reserved.

applied. For instance, for the acoustic wave equation, the shifted Laplace operator (Oosterlee et al., 2010) is an option.

Onshore seismic data contain all types of seismic waves, with P-waves often being the weakest part of the recorded signal. Inversion of P-waves requires careful suppression of other events in the data because they are considered as noise. Removing noise is a challenging task that can result in removal of the signal as well (Bakulin et al., 2015). This is one of the motivations to develop elastic FWI to use some of the other modes recorded, such as shear/surface waves to provide an accurate velocity model. This type of inversion requires elastic forward modeling.

A few 3D elastic iterative solvers have been developed so far (Databas and Louer, 2013; Li et al., 2015; Rizzuti and Mulder, 2015). The method proposed in this paper extends the approach from Belonosov et al. (2017) developed for the 3D acoustic wave equation, in which the preconditioner is a complex damped Helmholtz operator with some 1D vertically heterogeneous background. This was developed for FWI applications for macrovelocity reconstruction and showed fast convergence at low frequencies.

In this paper, we build the preconditioner following a similar methodology with special modifications for the elastic operator. We show the effectiveness of our method for cases of moderate lateral velocity variations in the overburden in Saudi Arabian land seismic data. Vertical variations of any kind are acceptable.

To use the capabilities of modern high-performance computing (HPC) systems, we parallelize our method in a hybrid manner involving message passing interface (MPI) and open multiprocessing (OpenMP). The speed of any parallel algorithm is strongly dependent on how effectively computation can be allocated among the available hardware. In that regard, strong scaling (Colella et al., 2007) is an important characteristic of a parallel method reflecting the ability to decrease the runtime with the increasing number of MPI processes/OpenMP threads involved in the computations. We provide scalability and performance data for our method.

A brief structure of the paper is as follows: We begin with a mathematical statement of the problem and describe the algorithm with details provided in Appendices A and B. Then, we validate the method via examples of simulation using different models. Finally, we summarize and provide with possible directions for further development of the method.

## BASICS OF THE METHOD

### Statement of the problem

Consider the first-order system of 3D isotropic elasticity equations in the temporal-FD written in terms of compliance

$$\left[ i\omega \mathbf{M}(x, y, z) - \mathbf{P} \frac{\partial}{\partial x} - \mathbf{Q} \frac{\partial}{\partial y} - \gamma(z) \mathbf{R} \frac{\partial}{\partial z} \right] \mathbf{u} = \mathbf{f}(x, y, z), \quad (1)$$

where  $\mathbf{u} = (u_x, u_y, u_z, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy})^T$  is a vector function of the particle velocity components ( $u_x, u_y, u_z$ ) and components of the stress tensor ( $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy}$ ),  $\omega$  is the real angular frequency

$$\mathbf{M} = \begin{pmatrix} \rho(x, y, z) \mathbf{I}_{3 \times 3} & 0 \\ 0 & \mathbf{S}(x, y, z) \end{pmatrix}, \quad (2)$$

where  $\rho$  is the density,  $\mathbf{I}_{3 \times 3}$  is the 3-by-3 identity matrix, and  $\mathbf{S}(x, y, z)$  is the compliance 6-by-6 matrix

$$\mathbf{S}(x, y, z) = \begin{pmatrix} a & -b & -b & & & & \\ -b & a & -b & & & & 0 \\ -b & -b & a & & & & \\ & & & c & 0 & 0 & \\ & & & 0 & 0 & c & 0 \\ & & & 0 & 0 & 0 & c \end{pmatrix},$$

$$a(x, y, z) = \frac{\lambda + \mu}{\mu(2\mu + 3\lambda)}, \quad b(x, y, z) = \frac{\lambda}{2\mu(2\mu + 3\lambda)},$$

$$c(x, y, z) = \frac{1}{\mu}, \quad (3)$$

where  $\lambda$  and  $\mu$  are the Lamé parameters. The  $\mathbf{P}$ ,  $\mathbf{Q}$ , and  $\mathbf{R}$  are 9-by-9 constant matrices of the following structures:

$$\mathbf{P} = \begin{pmatrix} 0 & \bar{\mathbf{P}} \\ \bar{\mathbf{P}}^T & 0 \end{pmatrix}, \quad \bar{\mathbf{P}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\mathbf{Q} = \begin{pmatrix} 0 & \bar{\mathbf{Q}} \\ \bar{\mathbf{Q}}^T & 0 \end{pmatrix}, \quad \bar{\mathbf{Q}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

$$\mathbf{R} = \begin{pmatrix} 0 & \bar{\mathbf{R}} \\ \bar{\mathbf{R}}^T & 0 \end{pmatrix}, \quad \bar{\mathbf{R}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (4)$$

where  $\gamma(z)$  is a damping function responsible for the perfectly matched layers (PMLs) (Berenger, 1996) along the vertical direction. The term  $\mathbf{f}(x, y, z)$  is the right side representing a seismic source.

Elastic equation 1 is solved in a cuboid domain  $D = [0, X] \times [0, Y] \times [0, Z]$  filled with a heterogeneous medium. The top boundary of the domain is the free surface unless stated to be absorbing. That means  $\sigma_{zz} = \sigma_{yz} = \sigma_{xz} = 0$  for  $z = 0$  or, equivalently,  $\mathbf{B}_0 \mathbf{u}(x, y, 0) = 0$ , where

$$\mathbf{B}_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5)$$

The computational domain is assumed to include the PML on the bottom; i.e., interval  $[0, Z]$  consists of two subintervals, where  $[0, Z_1]$  is the actual computational domain and  $[Z_1, Z]$  is the PML with the boundary condition on boundary  $z = Z$  (Collino and Tsogka, 1998):  $u_z = \sigma_{yz} = \sigma_{xz} = 0$ . The latter condition could be rewritten in an equivalent form  $\mathbf{B}_Z \mathbf{u}(x, y, Z) = 0$ , where

$$\mathbf{B}_Z = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (6)$$

On the lateral boundaries, domain  $D$  includes specially constructed sponge zones with periodic boundary conditions (for details, refer to Belonosov et al., 2017)

$$\begin{aligned}\bar{\mathbf{u}}(0, y, z) &= \bar{\mathbf{u}}(X, y, z), & \boldsymbol{\sigma}_x(0, y, z) &= \boldsymbol{\sigma}_x(X, y, z), \\ \bar{\mathbf{u}}(x, 0, z) &= \bar{\mathbf{u}}(x, Y, z), & \boldsymbol{\sigma}_y(x, 0, z) &= \boldsymbol{\sigma}_y(x, Y, z),\end{aligned}\quad (7)$$

with  $\bar{\mathbf{u}} = (u_x, u_y, u_z)^T$ ,  $\boldsymbol{\sigma}_x = (\sigma_{xx}, \sigma_{xy}, \sigma_{xz})^T$ , and  $\boldsymbol{\sigma}_y = (\sigma_{xy}, \sigma_{yy}, \sigma_{yz})^T$ . These conditions arise from the 2D fast Fourier transform (FFT) by the lateral directions being applied later. Also, it is worth explaining that its application does not allow using the PML in the  $x$ - and  $y$ -directions. Schematically, the structure of computational domain  $D$  is illustrated in Figure 1.

To numerically solve the boundary value problem for equation 1, we use the Krylov-type iterative method. In most cases, its straightforward application does not guarantee convergence. To handle this issue, a strategy based on a specially designed preconditioner is applied.

## Preconditioner

Denote the left side of elastic equation 1 by  $\mathcal{L}$

$$\mathcal{L} = i\omega\mathbf{M}(x, y, z) - \mathbf{P}\frac{\partial}{\partial x} - \mathbf{Q}\frac{\partial}{\partial y} - \gamma(z)\mathbf{R}\frac{\partial}{\partial z}. \quad (8)$$

Note that we also use notation  $\mathcal{L}$  for the operator defined by differential expression 8 subject to the boundary conditions described above.

Following the form of a preconditioner developed for the 2D elastic wave equation in (Neklyudov et al., 2011), consider an auxiliary, depth-dependent operator  $\mathcal{L}_0$  subject to the same boundary conditions as operator  $\mathcal{L}$

$$\mathcal{L}_0 = i\omega\mathbf{M}_0(z) - \mathbf{P}\frac{\partial}{\partial x} - \mathbf{Q}\frac{\partial}{\partial y} - \gamma(z)\mathbf{R}\frac{\partial}{\partial z}, \quad (9)$$

with complex-valued matrix  $\mathbf{M}_0$

$$\mathbf{M}_0(z) = \begin{pmatrix} \rho_0(z)\mathbf{I}_{3 \times 3} & 0 \\ 0 & (1 + i\beta)\mathbf{S}_0(z) \end{pmatrix}, \quad (10)$$

where

$$\mathbf{S}_0(z) = \begin{pmatrix} a_0(z) & -b_0(z) & -b_0(z) & 0 \\ -b_0(z) & a_0(z) & -b_0(z) & 0 \\ -b_0(z) & -b_0(z) & a_0(z) & c_0(z) \\ 0 & 0 & 0 & c_0(z) \end{pmatrix}, \quad (11)$$

where  $\beta$  is a positive real number smaller than one and is responsible for  $\mathbf{M}_0(z)$  complex shifting by analogy with the shifted Laplacian (Erlangga and Nabben, 2008a). At each depth level  $z$ , the values of functions  $\rho_0(z)$ ,  $a_0(z)$ ,  $b_0(z)$ , and  $c_0(z)$  are defined as some averaging of functions  $\rho(x, y, z)$ ,  $a(x, y, z)$ ,  $b(x, y, z)$ , and  $c(x, y, z)$  along the horizontal directions. In general,  $\rho_0(z)$ ,  $a_0(z)$ ,  $b_0(z)$ , and  $c_0(z)$  might be functions of any kind, but satisfying two conditions, where  $\rho_0(z) > 0$  and  $\mathbf{S}_0(z)$  is positive definite. We define them in a certain way to provide the proximity of operator  $\mathcal{L}_0$  to the original operator  $\mathcal{L}$  to ensure convergence of the

Krylov-type iterative method. In fact, operator  $\mathcal{L}_0(z)$  is of exactly the same structure as operator  $\mathcal{L}(x, y, z)$ , but with complex-valued coefficients that act in a 1D varying medium. In the next section, we illustrate a computationally efficient numerical algorithm to solve the boundary value problems for equation 1 with operator  $\mathcal{L}_0$  used instead of  $\mathcal{L}$ , or, in other words — how to invert operator  $\mathcal{L}_0$ .

We use this operator as a preconditioner to solve the boundary value problem for equation 1 that leads to the equivalent equation

$$\mathcal{L}\mathcal{L}_0^{-1}\mathbf{v} = \mathbf{f}, \quad (12)$$

over unknown vector  $\mathbf{v}$ , such that

$$\mathbf{u} = \mathcal{L}_0^{-1}\mathbf{v}. \quad (13)$$

We expect operator  $\mathcal{L}\mathcal{L}_0^{-1}$  to have better spectral properties than operator  $\mathcal{L}$  and by this to improve convergence of the iterative method. An illustration on how a successful preconditioner works can be found in Belonosov et al. (2017, Figure 1).

Even though the Krylov-type iterative method converged fast, the second requirement for successful application would be to perform multiplication of operator  $\mathcal{L}\mathcal{L}_0^{-1}$  by a vector at each iteration for a reasonable time. We represent operator  $\mathcal{L}$  as a perturbation of operator  $\mathcal{L}_0$  by operator  $\delta\mathcal{L}$

$$\begin{aligned}\mathcal{L}\mathbf{u} = \mathcal{L}_0\mathbf{u} - \delta\mathcal{L}\mathbf{u} &= \left[ i\omega\mathbf{M}_0(z) - \mathbf{P}\frac{\partial}{\partial x} - \mathbf{Q}\frac{\partial}{\partial y} - \gamma(z)\mathbf{R}\frac{\partial}{\partial z} \right] \mathbf{u} \\ &\quad - i\omega[\mathbf{M}_0(z) - \mathbf{M}(x, y, z)]\mathbf{u}.\end{aligned}\quad (14)$$

Substitution of expansion 14 into preconditioned equation 12 transforms it into

$$(I - \delta\mathcal{L}\mathcal{L}_0^{-1})\mathbf{v} = \mathbf{f}, \quad (15)$$

where  $I$  is the unity operator.

To solve equation 15, we apply the biconjugate gradient stabilized method (BiCGSTAB) (Van der Vorst, 1992; Saad, 2003) because it is the one with the lowest memory requirements of the available

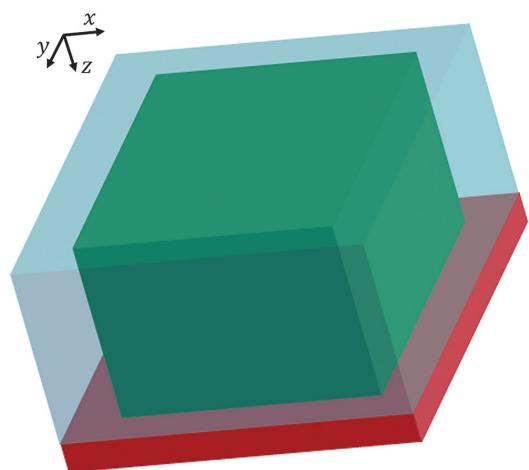


Figure 1. Computational domain  $D$  showing the physical domain (green area), sponge boundaries (blue), and PML (red).

methods. At each iteration, BiCGSTAB requires two matrix-by-vector products.

### Computation of $\mathcal{L}_0^{-1}\mathbf{w}$

Computation of  $\mathbf{u} = \mathcal{L}_0^{-1}\mathbf{w}$ , where  $\mathbf{w}$  is an arbitrary vector is equivalent to solving the boundary value problem for the equation

$$\left[ i\omega \mathbf{M}_0(z) - \mathbf{P} \frac{\partial}{\partial x} - \mathbf{Q} \frac{\partial}{\partial y} - \gamma(z) \mathbf{R} \frac{\partial}{\partial z} \right] \mathbf{u}(x, y, z) = \mathbf{w}(x, y, z) \quad (16)$$

in domain  $D$  with the same boundary conditions as are set on the initial elastic equation 1.

Assume that function  $\mathbf{w}(x, y, z)$  is expanded into the Fourier series with respect to the  $x$ - and  $y$ -coordinates and denote its coefficients by  $\hat{\mathbf{w}}(k_x, k_y; z)$ , where  $k_x$  and  $k_y$  are the respective spatial frequencies. Then, solving the boundary value problem for equation 16 could be broken down into two parts. First, we search for  $\hat{\mathbf{u}}(k_x, k_y; z)$ , which are coefficients of the Fourier series for function  $\mathbf{u}(x, y, z)$ . They are solutions to the ordinary differential equation (ODE) boundary value problems

$$\begin{aligned} \left[ i\omega \mathbf{M}_0(z) - ik_x \mathbf{P} - ik_y \mathbf{Q} - \gamma(z) \mathbf{R} \frac{d}{dz} \right] \hat{\mathbf{u}}(k_x, k_y; z) &= \hat{\mathbf{w}}(k_x, k_y; z), \\ \mathbf{B}_0 \hat{\mathbf{u}}(k_x, k_y; 0) &= 0, \\ \mathbf{B}_Z \hat{\mathbf{u}}(k_x, k_y; Z) &= 0, \end{aligned} \quad (17)$$

on the interval  $[0, Z]$ . The existence problem is addressed in Appendix A. Second, function  $\mathbf{u}(x, y, z)$  is reconstructed by computation of the Fourier series.

Numerically, for  $\mathbf{w}$  being a grid vector-function defined on a uniform grid along the  $x$ - and  $y$ -axes, computation of  $\hat{\mathbf{w}}(k_x, k_y; z)$  is the 2D discrete Fourier transform (DFT) of function  $\mathbf{w}(x, y, z)$  over the lateral coordinates. The Fourier series for  $\mathbf{u}(x, y, z)$  reduces to a finite sum that is the 2D inverse DFT of  $\hat{\mathbf{u}}(k_x, k_y; z)$ . To compute the DFTs, we apply a computationally efficient FFT from the highly

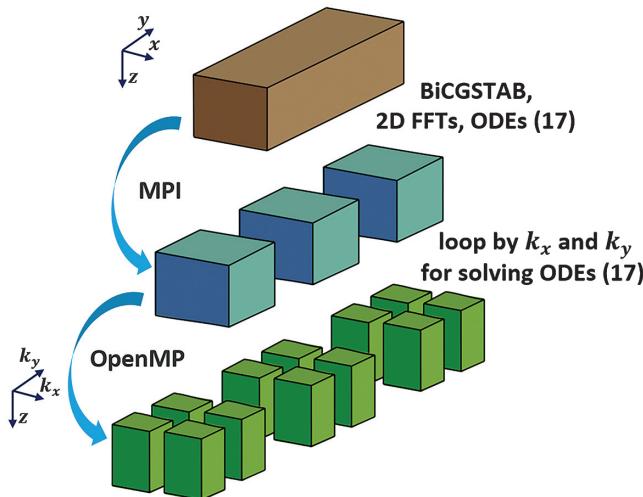


Figure 2. Hybrid parallelization scheme for the elastic iterative solver.

optimized software library Intel MKL (Intel, 2017). The boundary value problems 17 are solved numerically by finite-difference approximation that results in SLAEs with small, banded matrices (for details, see Appendix B) solved using a banded matrix solver from the Intel MKL library.

To sum up, the algorithm for matrix-by-vector multiplication is as follows:

- forward 2D FFT over lateral coordinates to vector-function  $\mathbf{w}$
- numerical solution of the ODE boundary value problems 17
- inverse 2D FFT to obtain the matrix-by-vector product.

### Hybrid parallelization

Numerical simulations of seismic waves in a 3D model were carried out using high-performance computing clusters. To optimize this process, different parallelization strategies, depending on architecture available might be applied. An important advantage of the proposed method is that it allows hybrid parallelization: MPI in conjunction with OpenMP. This allows more effective allocating parallel processes among available computational resources.

A parallelization strategy for our method is illustrated in Figure 2 and comprises two levels. The first level is MPI parallelization via spatial decomposition (the initial brown domain is decomposed into several smaller blue subdomains) of the BiCGSTAB, the 2D FFT and solving the boundary value problems 17. The 2D FFT version that we use supports parallelization along one of coordinates only. The second level is OpenMP parallelization within each blue subdomain: each OpenMP thread is responsible for its own set of  $k_x$  and  $k_y$  in boundary value problem 17.

According to our scheme, we expect that the main speed-up will be from MPI parallelization because it is a “global” one. OpenMP is used at each iteration, but “locally,” only for a minor part of the code. It might be beneficial in cases in which involvement of new MPI processes no longer accelerates the computational process, but some CPU cores are still available.

## NUMERICAL EXPERIMENTS

All results presented in the paper have been computed on a HPC cluster comprising nodes each having two Intel Xeon E5-2680v4 at 2400 MHz CPUs. The double precision floating point format has been used in the computations. As a stopping criterion for the BiCGSTAB, we used a  $10^{-3}$  threshold for the relative residual of the  $L_2$ -norm providing enough accuracy for FWI applications. We use a vertical point force as the seismic source.

A proper nondimensionalization is necessary to be applied to equation 1. This is a common practice performed in applications when different variables are at very different scales. In our case, the difference between computed stresses and velocities is of order  $10^6$ . To minimize this issue, we scale velocities by a factor of  $10^3$  and the stress components by a factor of  $10^{-3}$ .

### Accuracy analysis

At first, the wavefield computed via the solver in a homogeneous model with the PML at the top boundary was compared with the analytical solution (Aki and Richards, 1980). A domain of  $12 \times 12 \times 4.5$  km is filled with a homogeneous elastic medium of 2600 m/s P-wave velocity, 1500 m/s S-wave velocity, and  $2210 \text{ kg/m}^3$  den-

sity. For numerical computations, the domain was discretized using a uniform grid with lateral cell sizes of 60 m and vertical cells of 15 m. The source was excited in the middle of the model at 15 m depth. A 1D profile comparison of the 10 Hz monochromatic part of the vertical velocity component is presented in Figure 3. All traces are normalized to their maximum value. There is a very good agreement between the analytic and numerical modeling results (the root-mean-square [rms] deviation of the computed solution from the exact solution along the 1D profile in percent is 0.88%). Note that in this case, the lateral steps correspond to 2.5 points per minimum wavelength and the vertical cell size is 10 points.

For further verification, we have benchmarked the solver against a known time-domain solver based on explicit fourth-order finite differences for the 3D SEG/EAGE overthrust model (Aminzadeh et al., 1997) (Figure 4) with the free-surface top boundary. This model is discretized using a  $660 \times 660 \times 155$  uniform grid with cell sizes of 30 m in all directions resulting in eight points per minimum wavelength at 5 Hz frequency. Synthetic wavefields excited by a source fired in the middle of the model at 15 m depth were simulated with both methods at a frequency of 5 Hz. In Figure 5, we present the vertical particle velocities along the  $x$ -axis as well as along the  $y$ -axis. The results are normalized to their maximum value

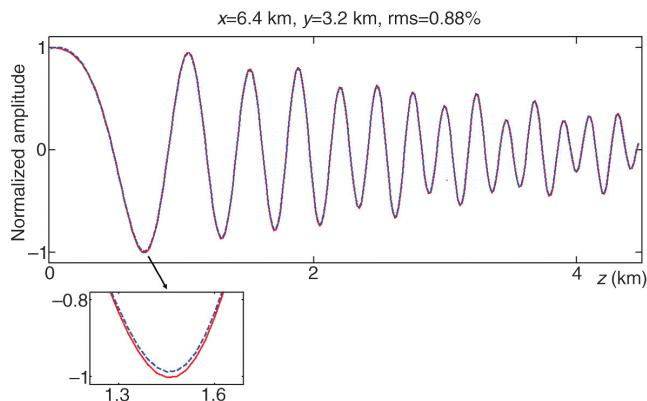


Figure 3. Vertical profile (real part) of the FD  $z$ -component of the displacement velocity at 10 Hz computed in the homogeneous model at location  $x = 6400$  m,  $y = 3200$  m. The exact solution is the dashed blue line and the iterative solver solution is the red line. The arrow points to a magnified part of the picture.

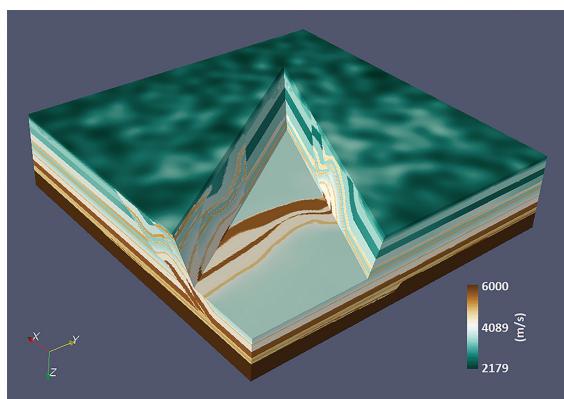


Figure 4. A 3D view of the  $V_p$ -velocity distribution in the 3D SEG/EAGE overthrust model ( $19.8 \times 19.8 \times 4.65$  km).

and show close agreement (rms difference is 3.9%–4.7%). The total 3D view of the vertical velocity computed by the iterative solver is shown in Figure 6.

### Strong scaling analysis

We estimate the strong scaling of our algorithm on an example of numerical simulation at a frequency of 5 Hz in the SEG/EAGE overthrust model (Figure 4). We measure  $t_1/t_m$  (see Figure 7), where  $t_m$  is the total computational time with  $m$  MPI processes. Results show the very good scalability of the algorithm up to 64 CPUs after which scalability worsens. This might be partly explained by the fact that the FFT shows strong scalability decay when the number of points in a subdomain becomes too small (Dmitruk et al., 2001).

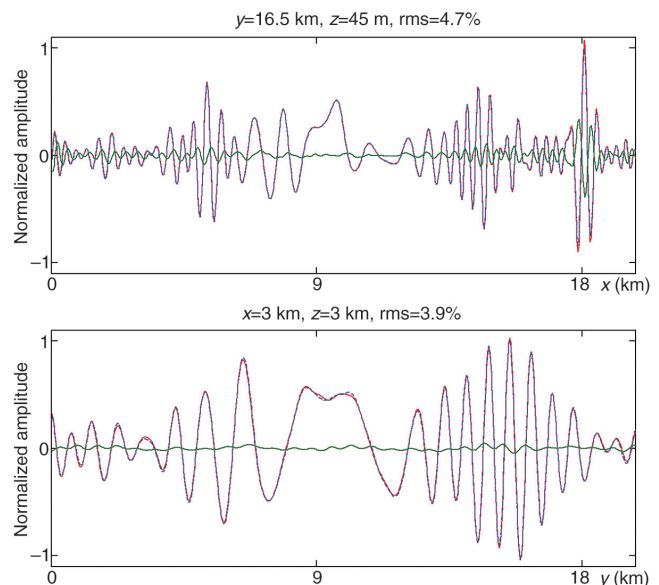


Figure 5. The horizontal profiles of the real part of the vertical component of the 5 Hz FD displacement velocities computed in the SEG/EAGE overthrust model at locations  $y = 16,500$  m,  $z = 45$  m (top) and  $x = 3000$  m,  $z = 3000$  m (bottom). The TD solver solution is the dashed blue line, the iterative solver solution is in red, and the difference multiplied by five is in green.

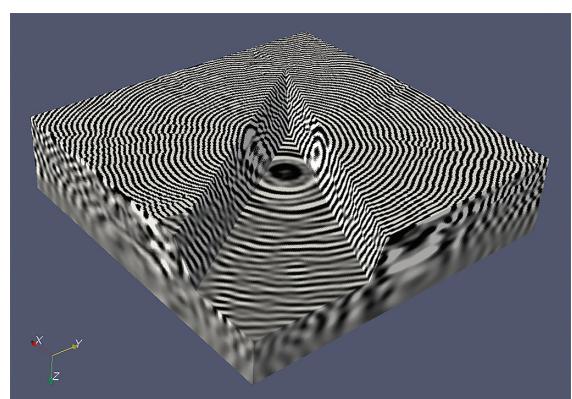


Figure 6. A 3D view of the real part of an FD wavefield of the vertical component of the particle velocity computed for the SEG/EAGE overthrust model.

For the case of OpenMP parallelization, the simulations were performed on a single CPU with 14 cores with hyperthreading switched off and without using MPI. Remember that multithreading is used at each iteration, but only to parallelize the loop over spatial frequencies solving the boundary value problems [17]. The strong scaling data measured for that part separately as well as for solving the whole problem are presented in Figure 8 showing that the OpenMP part scales well. The OpenMP contribution to acceleration of the total runtime (the green line), has an upper limit close to two. This is in a good agreement with Amdahl's (1967) law because only a part taking approximately 50% of the execution time is parallelized via OpenMP. According to the results presented in Figures 7 and 8, the speed-up factors in case of pure MPI parallelization with 32 processes is  $t_{32}^{\text{MPI}} = 30.2$  and with pure OpenMP with six threads is  $t_6^{\text{OMP}} = 1.63$ . Hybridization results in a speed-up of 46.8 that is very close to  $t_{32}^{\text{MPI}} \cdot t_6^{\text{OMP}}$ .

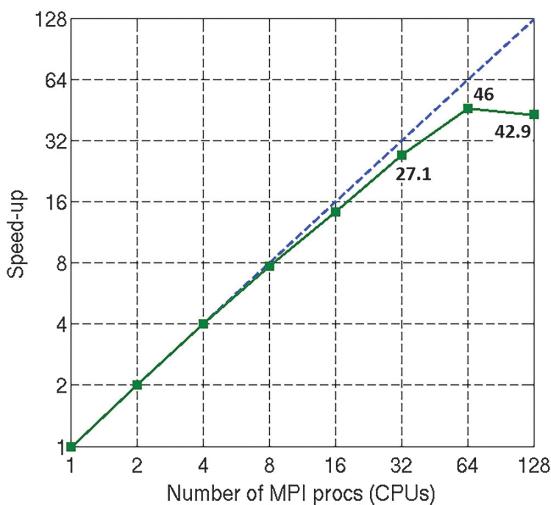


Figure 7. The MPI strong scalability analysis. The ideal speed-up is the dashed blue line, and the iterative solver scalability is in green.

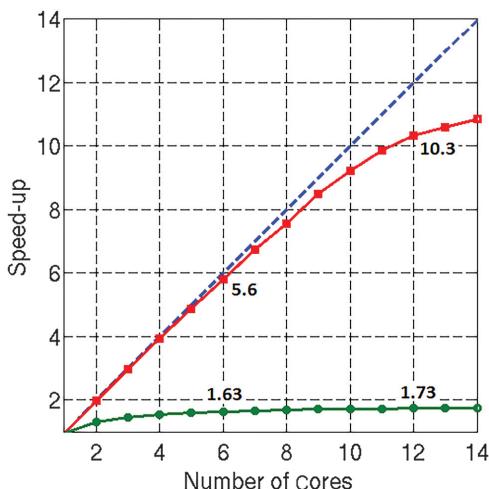


Figure 8. OpenMP strong scalability analysis. The ideal speed-up is the dashed blue line, the iterative solver multithread part speed-up is in red, and the total speed-up for the iterative solver is in green.

The further acceleration and increase of the OpenMP contribution of the total runtime might be achieved by hybrid parallelization of the BiCGSTAB process. Currently used function supports only MPI parallelization.

### Comparison with a known direct solver

As previously mentioned, the main advantage of an iterative approach over its FD competitor — direct approach, is that it requires less RAM. For 3D simulations, when hundreds of gigabytes are required that factor may become crucial when choosing a solver. For an example illustrating this fact, we considered a conventional direct method based on the lower–upper (LU) factorization. To decrease the memory needed, we applied it to the second-order elastic equation, approximated by the second-order finite differences. The major RAM required by this method is being spent to store LU factors that we computed via the Intel MKL function — PARDISO (Intel, 2017). In the third column of Table 1, we present the necessary memory being consumed by the described direct solver; in the second column — by our solver. The results show the difference being of around an order of magnitude. Note that for this comparison, the absorbing layers are not included for either method.

### Benchmarking against a known iterative solver

For performance analysis, we considered a known 3D elastic iterative solver — CARP-CG (Li et al., 2015). Following the paper, the same simulations were performed in a  $14.4 \times 14.4 \times 14.4$  km homogeneous model of 5000 m/s P-wave velocity, 2886.75 m/s S-velocity, and the density was estimated via Gardner's relation. We used the same lateral steps of four points per minimal wavelength. In the vertical direction, we used a smaller step of 10 points per minimal wavelength due to fourth-order finite-difference approximation used in the current version of our solver. It is worth explaining that we considered only the homogeneous model because other 3D simulations in this paper were performed in the inhomogeneous model not available to us.

In Table 2, we demonstrate the number of BiCGSTAB iterations for our solver (second column), as well as the number of iterations for CARP-CG (third column) (these data are taken from Table 9 of Li et al., 2015). Note that the ratio between the elements of the third column and the second column is almost the same (approximately 38) for all frequencies. We do not provide computational times because in our opinion, this comparison may be a bit unfair due to the difference in hardware and software, the code optimizations

**Table 1. RAM requirements for the iterative solver and the direct solver ( $N_x \times N_y \times N_z$  — the size of the computational domain in points; RAM(It) — RAM consumption of the iterative solver; and RAM(D) — the memory necessary to store LU factors).**

$N_x \times N_y \times N_z$	RAM (It)	RAM (D)
$50 \times 50 \times 50$	0.083 GB	13 GB
$75 \times 75 \times 75$	0.28 GB	72 GB
$100 \times 100 \times 100$	0.67 GB	285 GB
$115 \times 115 \times 115$	1 GB	467 GB

applied, and, possibly, some other factors. Having no access to the actual CARP-CG code, it is problematic to estimate its computational time on our hardware and with our settings.

### Convergence analysis in a model with strong lateral variations

The convergence of our solver strongly depends on the lateral variations of the medium parameters. To illustrate that, we performed simulations at 5 Hz in the model with a vertical interface comprising two layers of equal size (see Figure 9) with constant parameters  $V_P = V_{P_1}$ ,  $V_S = V_{S_1}$ ,  $\rho = \rho_1$  for  $x \geq L$  and  $V_P = V_{P_2}$ ,  $V_S = V_{S_2}$ ,  $\rho = \rho_2$  for  $x < L$  (here, we assume that  $V_S = V_P/\sqrt{3}$ ,  $\rho$  is estimated via Gardner's relation and  $V_{P_2} > V_{P_1}$ ). It is discretized with a uniform grid of  $400 \times 400 \times 200$  points. The steps correspond to 2.5 points per minimal wavelength in the horizontal directions and 10 points in the vertical direction. The source location is in the middle of the blue subdomain. In Table 3, we illustrate how the convergence depends on the ratio of  $V_{P_2}$  (for increasing  $V_{P_2}$ ) and

**Table 2. Our solver versus CARP-CG:**  $N_{\text{solver}}$  — number of BiCGSTAB iterations for our solver and  $N_{\text{CARP-CG}}$  — number of CARP-CG iterations.

Frequency	$N_{\text{solver}}$	$N_{\text{CARP-CG}}$
2.5 Hz	14	546
7.5 Hz	39	1402
12.5 Hz	58	2162
17.5 Hz	73	2817
22.5 Hz	87	3389

**Table 3. Convergence of the iterative solver for different lateral contrasts in the model presented in Figure 11.**

$V_{P_2}/V_{P_1}$	Number of iterations
1.25	17
1.5	42
1.75	1151
2	>10,000 (divergence)

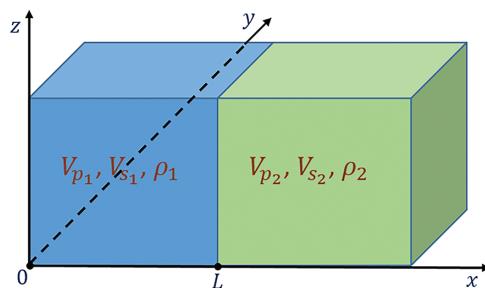


Figure 9. A model with a vertical interface.

$V_{P_1} = 2000$  m/s. The ratio of two is an example when the solver diverges.

For further improvement of the convergence, an extra preconditioning might be an option. For example, a similar ideology of a second-level preconditioner developed for the acoustic wave equation (Belonosov et al., 2017) or a preconditioner based on deflation (Erlangga and Nabben, 2008b) might be considered.

### Simulation in a realistic synthetic model

As an example of a numerical simulation using a realistic scenario, we considered a land model depicted in Figure 10 with the deep structure common to the eastern province of Saudi Arabia. Its dimensions correspond to one swath size typically used in the land

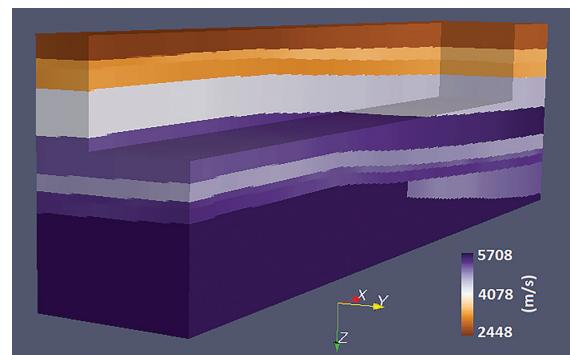


Figure 10. Compressional velocity distribution for a typical Saudi Arabian land model ( $24 \times 4.2 \times 6.5$  km deep).

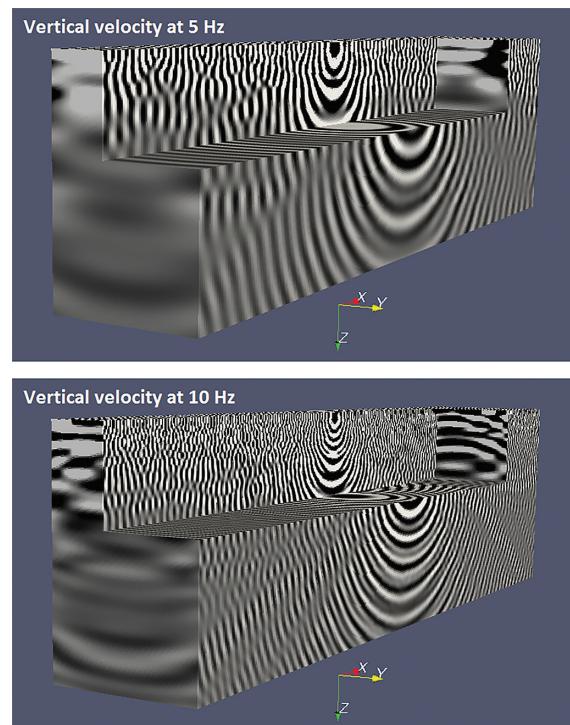


Figure 11. A 3D view of the real part of a FD wavefield of the vertical component of the particle velocity computed for the model depicted in Figure 10.

**Table 4. Number of iterations versus frequency for numerical simulation with the elastic iterative solver for the model depicted in Figure 9.**

Frequency	2 Hz	4 Hz	6 Hz	8 Hz	10 Hz	12 Hz	14 Hz	16 Hz	20 Hz
Iterations	10	27	58	92	158	229	311	450	1123

seismic acquisition. The model was discretized with a uniform grid of  $957 \times 169 \times 651$  points with a lateral cell size of 25 m and a vertical cell size of 10 m. The source was placed in the middle of the area at 10 m depth. In Figure 11, we present a 3D view of the vertical velocity at 5 and 10 Hz computed with the iterative solver. To obtain these results, we used 18 computational nodes with four MPI processes per node and seven OpenMP threads per MPI process. The total computational times for the 5 and 10 Hz solutions are 32 and 108 min, respectively.

To understand how the number of iterations depends on the frequency, we carried out the previous experiment but over a range of frequencies from 2 to 20 Hz (Table 4). Here, we used the spatial lateral steps corresponding to 2.5 points per minimal wavelength and vertical — 10 points, so that the total number of grid points for each experiment is different. Our results suggest that when the frequency is low ( $\leq 10$  Hz), doubling of the frequency leads to the increase of the number of iterations by a factor of three. For higher frequencies, this coefficient increases, but up to 20 Hz, it is less than eight. It is worth mentioning that for models with more complicated lateral structures, the convergence rate deteriorates. For instance, for the SEG/EAGE overthrust model (Figure 4) at 5 Hz, it converges for 94 iterations, whereas it converges at 10 Hz for 904 iterations.

## CONCLUSION

We presented an iterative method for parallel FD elastic wavefield simulation in a 3D land scenario. Its key component, a unique preconditioner, ensures fast convergence of the iterative process in models with moderate lateral variations at low frequencies needed for macrovelocity reconstruction with FWI. Having a limitation of planar free-surface topography, it could be incorporated as a forward modeling engine in an inversion process with data after static corrections or redatuming has been applied. For further improvement of the parallel implementation and, as the result, of the total computational time, one might consider incorporation of a parallel FFT based on 2D domain decomposition, which could increase the upper limit of MPI scalability.

In the future, our method might be extended to handle an arbitrary curved air-rock free surface using a hybrid approach, in which the simulation in the upper part of the model including that surface could be performed by a finite-element method.

Simulation for the marine case that is not covered by our paper is another challenge. Due to the S-wave velocity being zero in water, our method in its current state could not be applied to a combined acoustic/elastic model. That issue might be overcome by an appropriate perturbation of an imaginary part of the  $\mu$  Lamé parameter in water. Preliminary numerical experiments show convergence of the method in the case of a planar seabed model. The case of a curved sea bottom requires further investigation.

## AUTHOR CONTRIBUTIONS

The method presented in the paper has been proposed and theoretically justified by V. Kostin and V. Tcheverda. The numerical algorithm has been developed by M. Belonosov and D. Neklyudov. The software development and verification of the method have been carried out by M. Belonosov.

## ACKNOWLEDGMENTS

We are grateful to M. Jervis for reviewing our manuscript. Special thanks go to A. Bakulin, V. Lisitsa, and M. Dmitriev for fruitful discussions and advice on this topic. Two of the authors (V. Kostin and V. Tcheverda) have been sponsored by the Russian Science Foundation grant 17-17-01128.

## DATA AND MATERIALS AVAILABILITY

Data associated with this research are confidential and cannot be released.

## APPENDIX A

### EXISTENCE OF THE SOLUTION TO THE BOUNDARY VALUE PROBLEM 17

In the general case, the boundary value problem for a system of ODEs subject to some boundary conditions may not have a solution. That happens when there is a nonzero solution to the problem with a zero right side. Let us demonstrate that for the system of equations 17 this is not the case.

We restrict ourselves to the case in which  $\gamma(z) = 1$ , this is, equivalent to excluding the effect of PML on the bottom boundary. Let us introduce three relevant functional spaces:

- $L_2([0, Z])$ : a Hilbert space of square-integrable 9C vector-functions  $\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9)^T$  with the inner product defined as

$$(\mathbf{u}, \mathbf{v})_{L_2} = \int_0^Z (\mathbf{u}(z), \mathbf{v}(z)) dz, \\ \text{for } \mathbf{u}(z) \text{ and } \mathbf{v}(z) \text{ from } L_2([0, Z]), \quad (\text{A-1})$$

with the integrand being the inner product in vector space  $C^9$  of vectors of nine complex valued components

$$(\mathbf{u}(z), \mathbf{v}(z)) = \sum_{j=1}^9 u_j(z) \overline{v_j(z)}; \quad (\text{A-2})$$

- $H^1([0, Z])$ : a subspace in  $L_2([0, Z])$  of functions whose first derivatives belong to  $L_2([0, Z])$ ;
- $\mathbb{H}$ : a linear subspace in  $H^1([0, Z])$  defined by boundary conditions

$$\begin{cases} \mathbf{B}_0 \mathbf{u}(0) = 0 \\ \mathbf{B}_Z \mathbf{u}(Z) = 0 \end{cases} \quad (\text{A-3})$$

that is the domain of differential operator  $\mathcal{L}_1$  defined by expression

$$\mathcal{L}_1 \mathbf{u} = \left[ ik_x \mathbf{P} + ik_y \mathbf{Q} + \mathbf{R} \frac{d}{dz} \right] \mathbf{u}, \quad (\text{A-4})$$

where matrices  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$ ,  $\mathbf{B}_0$ , and  $\mathbf{B}_Z$  are defined in formulas 4–6.

Operator  $\mathcal{L}_1$  maps its domain  $\mathbb{H}$  into space  $L_2([0, Z])$ . Given the right side vector-function  $\hat{\mathbf{w}}(k_x, k_y; z) \in L_2([0, Z])$ , boundary value problem 17 can be represented as an operator equation:

$$\left[ i\omega \begin{pmatrix} \rho_0(z) \mathbf{I}_{3 \times 3} & 0 \\ 0 & (1 + i\beta) \mathbf{S}_0(z) \end{pmatrix} - \mathcal{L}_1 \right] \hat{\mathbf{u}}(k_x, k_y; z) = \hat{\mathbf{w}}(k_x, k_y; z) \quad (\text{A-5})$$

with the respective homogeneous problem:

$$\left[ i\omega \begin{pmatrix} \rho_0(z) \mathbf{I}_{3 \times 3} & 0 \\ 0 & (1 + i\beta) \mathbf{S}_0(z) \end{pmatrix} - \mathcal{L}_1 \right] \mathbf{u} = 0 \quad (\text{A-6})$$

and, consequently, our task is to show that problem A-6 has only trivial solutions. That is equivalent to prove that the eigenvalue problem

$$\mathcal{L}_1 \mathbf{u} = \lambda \begin{pmatrix} \rho_0 \mathbf{I}_{3 \times 3} & 0 \\ 0 & (1 + i\beta) \mathbf{S}_0 \end{pmatrix} \mathbf{u} \quad (\text{A-7})$$

cannot have purely imaginary eigenvalues  $\lambda$ .

One can easily verify that operator  $\mathcal{L}_1$  is skew-Hermitian; i.e., for any pair of vector-functions  $\mathbf{u}$  and  $\mathbf{v}$  from space  $\mathbb{H}$  the following equality holds:

$$(\mathcal{L}_1 \mathbf{u}, \mathbf{v})_{L_2} = -(\mathbf{u}, \mathcal{L}_1 \mathbf{v})_{L_2}. \quad (\text{A-8})$$

Hence, the quadratic form  $(\mathcal{L}_1 \mathbf{u}, \mathbf{u})_{L_2}$  may have only purely imaginary values. For solution  $\mathbf{u}$  to problem A-7, the respective  $\lambda$  can be expressed as the ratio

$$\lambda = (\mathcal{L}_1 \mathbf{u}, \mathbf{u})_{L_2} / \left( \begin{pmatrix} \rho_0 \mathbf{I}_{3 \times 3} & 0 \\ 0 & (1 + i\beta) \mathbf{S}_0 \end{pmatrix} \mathbf{u}, \mathbf{u} \right)_{L_2}. \quad (\text{A-9})$$

The numerator of the ratio in equality A-9 is purely imaginary; hence, there are only two possibilities for  $\lambda$  to be purely imaginary, i.e.,  $\lambda = i\omega$ :

- $\lambda = 0$ , which means  $\omega = 0$ . We do not consider this case because the angular frequency  $\omega \neq 0$ .
- $\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{S}_0 \end{pmatrix} \mathbf{u}, \mathbf{u} \Big|_{L_2} = 0$ . This implies that  $u_4 = u_5 = u_6 = u_7 = u_8 = u_9 = 0$  due to  $\mathbf{S}_0$  is a symmetric positive definite matrix. Using this, from equation A-3 one gets that  $\lambda \rho_0(u_1, u_2, u_3)^T = 0$ . Having  $\lambda \neq 0$ , we conclude that  $(u_1, u_2, u_3)^T = 0$ . Hence,  $\mathbf{u}$  is zero.

Thus, we have just proved that in the case of  $\gamma(z) = 1$  the boundary value problem 17 is resolvable. In the case of the PML in place, when  $\gamma(z)$  is not a constant, we do not have a theoretical proof, but in our modeling tests, we did not encounter any issues.

## APPENDIX B

### NUMERICAL SOLUTION TO THE BOUNDARY VALUE PROBLEM 17

For unknown vector-function  $\hat{\mathbf{u}}(k_x, k_y; z) = (\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4, \hat{u}_5, \hat{u}_6, \hat{u}_7, \hat{u}_8, \hat{u}_9)^T$  and right side vector-function  $\hat{\mathbf{w}}(k_x, k_y; z) = (\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4, \hat{w}_5, \hat{w}_6, \hat{w}_7, \hat{w}_8, \hat{w}_9)^T$ , the system in formula 17 can be reduced to the following system of six ODEs:

$$\begin{aligned} \frac{\gamma(z)}{i\omega} \frac{d}{dz} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_6 \\ \hat{u}_3 \\ \hat{u}_7 \\ \hat{u}_8 \end{bmatrix} + & \begin{bmatrix} 0 & 0 & 0 & \frac{k_x}{\omega} & 0 & -c_0 \\ 0 & 0 & 0 & \frac{k_y}{\omega} & -c_0 & 0 \\ 0 & 0 & 0 & -\rho_0 & \frac{k_y}{\omega} & \frac{k_x}{\omega} \\ d_{11} & d_{12} & d_{13} & 0 & 0 & 0 \\ d_{21} & d_{22} & d_{23} & 0 & 0 & 0 \\ d_{31} & d_{32} & d_{33} & 0 & 0 & 0 \end{bmatrix} \\ \times \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_6 \\ \hat{u}_3 \\ \hat{u}_7 \\ \hat{u}_8 \end{bmatrix} & = -\frac{1}{i\omega} \begin{bmatrix} \hat{w}_8 \\ \hat{w}_7 \\ \hat{w}_3 \\ \bar{\hat{w}}_1 \\ \bar{\hat{w}}_2 \\ \bar{\hat{w}}_3 \end{bmatrix}, \end{aligned} \quad (\text{B-1})$$

for six components  $\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_6, \hat{u}_7, \hat{u}_8$  subject to boundary conditions

$$\begin{aligned} \hat{u}_6(0) &= \hat{u}_7(0) = \hat{u}_8(0) = 0, \\ \hat{u}_3(Z) &= \hat{u}_7(Z) = \hat{u}_8(Z) = 0. \end{aligned} \quad (\text{B-2})$$

Three remaining components are evaluated by explicit formulas

$$\begin{aligned} \hat{u}_4 &= \frac{k_x a_0 \hat{u}_1 + k_y b_0 \hat{u}_2}{\omega(1 + i\beta)(a_0^2 - b_0^2)} + \frac{b_0 \hat{u}_6}{a_0 - b_0} + \frac{a_0 \hat{w}_4 + b_0 \hat{w}_5}{i\omega(1 + i\beta)(a_0^2 - b_0^2)}, \\ \hat{u}_5 &= \frac{k_y b_0 \hat{u}_1 + k_x a_0 \hat{u}_2}{\omega(1 + i\beta)(a_0^2 - b_0^2)} + \frac{b_0 \hat{u}_6}{a_0 - b_0} + \frac{b_0 \hat{w}_4 + a_0 \hat{w}_5}{i\omega(1 + i\beta)(a_0^2 - b_0^2)}, \\ \hat{u}_9 &= \frac{1}{\omega(1 + i\beta)c_0} [k_y \hat{u}_1 + k_x \hat{u}_2 - \omega \hat{w}_9]. \end{aligned} \quad (\text{B-3})$$

In formulas B-2 and B-3, we use the following notations:

$$\begin{aligned}
d_{11} &= \frac{k_x b_0}{\omega(a_0 - b_0)}, \quad d_{12} = \frac{k_y b_0}{\omega(a_0 - b_0)}, \\
d_{13} &= \frac{2(1+i\beta)b_0^2}{a_0 - b_0} - (1+i\beta)a_0, \quad d_{21} = \frac{k_x k_y}{\omega^2(1+i\beta)} \left[ \frac{b_0}{a_0^2 - b_0^2} + \frac{1}{c_0} \right], \\
d_{22} &= \frac{k_x^2}{\omega^2(1+i\beta)c_0} + \frac{k_y^2 a_0}{\omega^2(1+i\beta)(a_0^2 - b_0^2)} - \rho_0, \quad d_{23} = \frac{k_y b_0}{\omega(a_0 - b_0)}, \\
d_{31} &= \frac{k_x^2 a_0}{\omega^2(1+i\beta)(a_0^2 - b_0^2)} + \frac{k_y^2}{\omega^2(1+i\beta)c_0} - \rho_0, \\
d_{32} &= \frac{k_x k_y}{\omega^2(1+i\beta)} \left[ \frac{b_0}{a_0^2 - b_0^2} + \frac{1}{c_0} \right], \\
d_{33} &= \frac{k_x b_0}{\omega(a_0 - b_0)}, \quad \bar{w}_1 = \hat{w}_6 + \frac{b_0(\hat{w}_4 + \hat{w}_5)}{(a_0 - b_0)}, \\
\bar{w}_2 &= \hat{w}_2 + \frac{k_x}{\omega(1+i\beta)c_0} \hat{w}_9 + \frac{k_y(b_0 \hat{w}_4 + a_0 \hat{w}_5)}{\omega(1+i\beta)(a_0^2 - b_0^2)}, \\
\bar{w}_3 &= \hat{w}_1 + \frac{k_y}{\omega(1+i\beta)c_0} \hat{w}_9 + \frac{k_x(a_0 \hat{w}_4 + b_0 \hat{w}_5)}{\omega(1+i\beta)(a_0^2 - b_0^2)}. \tag{B-4}
\end{aligned}$$

Let us emphasize that these manipulations do not affect the conclusion made in Appendix A regarding solvability of the boundary value problem 17; i.e., the boundary value problem B-1 subject to B-2 is solvable.

To numerically solve boundary value problem B-1, given integer  $N$  — the number of grid points in the  $z$ -direction, interval  $[0, Z]$  is covered by a uniform grid of step  $h_z = Z/(N-1)$  composed of  $N$  nodes with integer indices  $z_i = i \cdot h_z$  ( $i = 0, \dots, N-1$ ), so that  $z_0 = 0$  and  $z_{N-1} = Z$ , and  $N-1$  nodes with half-integer indices  $z_{i+\frac{1}{2}} = (i+1/2) \cdot h_z$  ( $i = 0, 1, \dots, N-2$ ). For the sake of brevity, the grid points with integer indices we call integer grid points and the points with half-integer indices we call half-integer grid points.

To approximate a certain function  $v(z)$  at integer grid point  $z_i$  or half-integer grid point  $z_{i+\frac{1}{2}}$  we take its value at this point and denote by  $(v)_{z_i}$  or  $(v)_{z_{i+\frac{1}{2}}}$ , respectively. Its first derivative is approximated by standard staggered-grid (Richtmyer and Morton, 1957) finite-difference approximation with a stencil depicted in the top image of Figure B-1:

$$\begin{aligned}
\frac{dv}{dz} \Big|_{z_i} &= \frac{9(v)_{z_{i+\frac{1}{2}}} - (v)_{z_{i-\frac{1}{2}}} - (v)_{z_{i+\frac{3}{2}}} - (v)_{z_{i-\frac{3}{2}}}}{8h_z} + O(h_z^4), \\
\frac{dv}{dz} \Big|_{z_{i+\frac{1}{2}}} &= \frac{9(v)_{z_i} - (v)_{z_{i+1}} - (v)_{z_{i+2}} - (v)_{z_{i-1}}}{8h_z} + O(h_z^4). \tag{B-5}
\end{aligned}$$

Applying these rules to approximate the first three ODEs of system B-1 at integer grid points  $z_i$  for  $2 \leq i \leq N-3$  and to its other three equations at half-integer grid point  $z_{i+\frac{1}{2}}$  for  $1 \leq i \leq N-3$ , we arrive at the following linear algebraic equations:

$$\left\{
\begin{aligned}
&\frac{\gamma(z_i)}{i\omega} \left[ \frac{9(\hat{u}_1)_{z_{i+\frac{1}{2}}} - (\hat{u}_1)_{z_{i-\frac{1}{2}}} - (\hat{u}_1)_{z_{i+\frac{3}{2}}} - (\hat{u}_1)_{z_{i-\frac{3}{2}}}}{8h_z} \right] + \frac{k_x}{\omega} (\hat{u}_3)_{z_i} \\
&- c_0(z_i) (\hat{u}_8)_{z_i} = -\frac{\hat{w}_8(z_i)}{i\omega}, \\
&\frac{\gamma(z_i)}{i\omega} \left[ \frac{9(\hat{u}_2)_{z_{i+\frac{1}{2}}} - (\hat{u}_2)_{z_{i-\frac{1}{2}}} - (\hat{u}_2)_{z_{i+\frac{3}{2}}} - (\hat{u}_2)_{z_{i-\frac{3}{2}}}}{8h_z} \right] + \frac{k_y}{\omega} (\hat{u}_3)_{z_i} \\
&- c_0(z_i) (\hat{u}_7)_{z_i} = -\frac{\hat{w}_7(z_i)}{i\omega}, \\
&\frac{\gamma(z_i)}{i\omega} \left[ \frac{9(\hat{u}_6)_{z_{i+\frac{1}{2}}} - (\hat{u}_6)_{z_{i-\frac{1}{2}}} - (\hat{u}_6)_{z_{i+\frac{3}{2}}} - (\hat{u}_6)_{z_{i-\frac{3}{2}}}}{8h_z} \right] - \rho_0(z_i) (\hat{u}_3)_{z_i} + \frac{k_y}{\omega} (\hat{u}_7)_{z_i} \\
&+ \frac{k_x}{\omega} (\hat{u}_8)_{z_i} = -\frac{\hat{w}_8(z_i)}{i\omega}, \\
&\frac{\gamma(z_{i+\frac{1}{2}})}{i\omega} \left[ \frac{9(\hat{u}_3)_{z_{i+\frac{1}{2}}} - (\hat{u}_3)_{z_i} - (\hat{u}_3)_{z_{i+2}} - (\hat{u}_3)_{z_{i-1}}}{8h_z} \right] + d_{11}(z_{i+\frac{1}{2}}) (\hat{u}_1)_{z_{i+\frac{1}{2}}} \\
&+ d_{12}(z_{i+\frac{1}{2}}) (\hat{u}_2)_{z_{i+\frac{1}{2}}} + d_{13}(z_{i+\frac{1}{2}}) (\hat{u}_6)_{z_{i+\frac{1}{2}}} = -\frac{\bar{w}_1(z_{i+\frac{1}{2}})}{i\omega}, \\
&\frac{\gamma(z_{i+\frac{1}{2}})}{i\omega} \left[ \frac{9(\hat{u}_7)_{z_{i+\frac{1}{2}}} - (\hat{u}_7)_{z_i} - (\hat{u}_7)_{z_{i+2}} - (\hat{u}_7)_{z_{i-1}}}{8h_z} \right] + d_{21}(z_{i+\frac{1}{2}}) (\hat{u}_1)_{z_{i+\frac{1}{2}}} \\
&+ d_{22}(z_{i+\frac{1}{2}}) (\hat{u}_2)_{z_{i+\frac{1}{2}}} + d_{23}(z_{i+\frac{1}{2}}) (\hat{u}_6)_{z_{i+\frac{1}{2}}} = -\frac{\bar{w}_2(z_{i+\frac{1}{2}})}{i\omega}, \\
&\frac{\gamma(z_{i+\frac{1}{2}})}{i\omega} \left[ \frac{9(\hat{u}_8)_{z_{i+\frac{1}{2}}} - (\hat{u}_8)_{z_i} - (\hat{u}_8)_{z_{i+2}} - (\hat{u}_8)_{z_{i-1}}}{8h_z} \right] + d_{31}(z_{i+\frac{1}{2}}) (\hat{u}_1)_{z_{i+\frac{1}{2}}} \\
&+ d_{32}(z_{i+\frac{1}{2}}) (\hat{u}_2)_{z_{i+\frac{1}{2}}} + d_{33}(z_{i+\frac{1}{2}}) (\hat{u}_6)_{z_{i+\frac{1}{2}}} = -\frac{\bar{w}_3(z_{i+\frac{1}{2}})}{i\omega}.
\end{aligned} \tag{B-6}
\right.$$

To approximate the first derivative of a certain function  $v(z)$  at integer points  $z_i$  ( $i = 1, N-2$ ) or at half-integer grid points  $z_{i+\frac{1}{2}}$  ( $i = 0, N-2$ ), we use another staggered-grid scheme with a stencil presented in the middle image of Figure B-1:

$$\begin{aligned}
\frac{dv}{dz} \Big|_{z_i} &= \frac{(v)_{z_{i+\frac{1}{2}}} - (v)_{z_{i-\frac{1}{2}}}}{h_z} + O(h_z^2), \\
\frac{dv}{dz} \Big|_{z_{i+\frac{1}{2}}} &= \frac{(v)_{z_{i+1}} - (v)_{z_i}}{h_z} + O(h_z^2).
\end{aligned} \tag{B-7}$$

Using approximations B-6, the first three ODEs of system B-1 at grid points  $z_1, z_{N-2}$ , and its other three equations at points  $z_{\frac{1}{2}}, z_{N-\frac{3}{2}}$  are approximated by equations

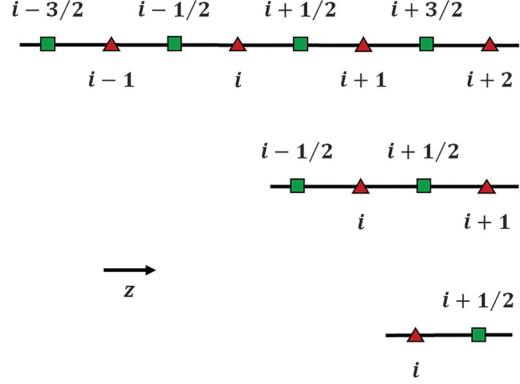


Figure B-1. Stencils used for a finite-difference approximation: the top — in formula B-4, the middle — in formula B-6, and the bottom — in formula B-8.

$$\left\{
\begin{aligned}
& \frac{\gamma(z_i)}{i\omega} \frac{(\hat{u}_1)_{z_{i+\frac{1}{2}}} - (\hat{u}_1)_{z_{i-\frac{1}{2}}}}{h_z} + \frac{k_x}{\omega} (\hat{u}_3)_{z_i} - c_0(z_i) (\hat{u}_8)_{z_i} \\
&= -\frac{\hat{w}_3(z_i)}{i\omega}, \\
& \frac{\gamma(z_i)}{i\omega} \frac{(\hat{u}_2)_{z_{i+\frac{1}{2}}} - (\hat{u}_2)_{z_{i-\frac{1}{2}}}}{h_z} + \frac{k_y}{\omega} (\hat{u}_3)_{z_i} - c_0(z_i) (\hat{u}_7)_{z_i} \\
&= -\frac{\hat{w}_7(z_i)}{i\omega}, \\
& \frac{\gamma(z_i)}{i\omega} \frac{(\hat{u}_6)_{z_{i+\frac{1}{2}}} - (\hat{u}_6)_{z_{i-\frac{1}{2}}}}{h_z} - \rho_0(z_i) (\hat{u}_3)_{z_i} + \frac{k_y}{\omega} (\hat{u}_7)_{z_i} \\
&+ \frac{k_x}{\omega} (\hat{u}_8)_{z_i} = -\frac{\bar{w}_3(z_i)}{i\omega}, \\
& \frac{\gamma(z_{i+\frac{1}{2}})}{i\omega} \frac{(\hat{u}_3)_{z_{i+1}} - (\hat{u}_3)_{z_i}}{h_z} + d_{11}(z_{i+\frac{1}{2}}) (\hat{u}_1)_{z_{i+\frac{1}{2}}} + d_{12}(z_{i+\frac{1}{2}}) (\hat{u}_2)_{z_{i+\frac{1}{2}}} \\
&+ d_{13}(z_{i+\frac{1}{2}}) (\hat{u}_6)_{z_{i+\frac{1}{2}}} = -\frac{\bar{w}_1(z_{i+\frac{1}{2}})}{i\omega}, \\
& \frac{\gamma(z_{i+\frac{1}{2}})}{i\omega} \frac{(\hat{u}_7)_{z_{i+1}} - (\hat{u}_7)_{z_i}}{h_z} + d_{21}(z_{i+\frac{1}{2}}) (\hat{u}_1)_{z_{i+\frac{1}{2}}} + d_{22}(z_{i+\frac{1}{2}}) (\hat{u}_2)_{z_{i+\frac{1}{2}}} \\
&+ d_{23}(z_{i+\frac{1}{2}}) (\hat{u}_6)_{z_{i+\frac{1}{2}}} = -\frac{\bar{w}_2(z_{i+\frac{1}{2}})}{i\omega}, \\
& \frac{\gamma(z_{i+\frac{1}{2}})}{i\omega} \frac{(\hat{u}_8)_{z_{i+1}} - (\hat{u}_8)_{z_i}}{h_z} + d_{31}(z_{i+\frac{1}{2}}) (\hat{u}_1)_{z_{i+\frac{1}{2}}} + d_{32}(z_{i+\frac{1}{2}}) (\hat{u}_2)_{z_{i+\frac{1}{2}}} \\
&+ d_{33}(z_{i+\frac{1}{2}}) (\hat{u}_6)_{z_{i+\frac{1}{2}}} = -\frac{\bar{w}_3(z_{i+\frac{1}{2}})}{i\omega}.
\end{aligned} \tag{B-8}
\right.$$

Let us approximate the first derivative of a certain function  $v(z)$  at point  $z_i$  by the following formula with a stencil presented in the bottom image of Figure B-1:

$$\frac{dv}{dz}\Big|_{z_i} = \frac{(v)_{z_{i+\frac{1}{2}}} - (v)_{z_i}}{h_z/2} + O(h_z). \tag{B-9}$$

Approximation B-8 is used to replace  $d\hat{u}_6/dz$  at point  $z_0$  in the third equation of system B-1 to get

$$(\hat{u}_3)_{z_0} = \frac{1}{i\omega\rho_0(z_0)} \left[ 2\gamma(z_0) \frac{(\hat{u}_6)_{z_{\frac{1}{2}}}}{h_z} + \hat{w}_3(z_0) \right]. \tag{B-10}$$

Together, formulas B-5, B-7, B-9, and boundary conditions B-2 build the finite-difference approximation of boundary value problem B-1. This results in a SLAE with a square, banded matrix of size  $(6N - 9) \times (6N - 9)$ . Due to approximation of the boundary conditions with the first and second order, this scheme is not of fourth order. It can be improved to the fourth order by an appropriate approximation of the boundary conditions, but we skip this step in this paper for the sake of brevity.

It is worth mentioning that if for initial equation 1 we assumed the top boundary of the computational domain to be absorbing (for example by including a PML), then formula B-9 would be replaced by  $(\hat{u}_3)_{z_0} = 0$ . The rest of the scheme would be the same.

## REFERENCES

- Aki, K., and P. G. Richards, 1980, Quantitative seismology, theory and methods: W. H. Freeman and Co 1.
- Amdahl, G. M., 1967, Validity of the single processor approach to achieving large-scale computing capabilities: AFIPS Conference Proceedings, 483–485.
- Aminzadeh, F., J. Brac, and T. Kuntz, 1997, 3-D salt and overthrust models: SEG.
- Bakulin, A., M. Dmitriev, P. Golikov, and D. Neklyudov, 2015, Enhancing 3D broadband land seismic data with smart super groups for processing and FWI: 77th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: [10.3997/2214-4609.201413435](https://doi.org/10.3997/2214-4609.201413435).
- Belonosov, M., M. Dmitriev, V. Kostin, D. Neklyudov, and V. Tcheverda, 2017, An iterative solver for the 3D Helmholtz equation: Journal of Computational Physics, **345**, 330–344, doi: [10.1016/j.jcp.2017.05.026](https://doi.org/10.1016/j.jcp.2017.05.026).
- Berenger, J. P., 1996, Three-dimensional perfectly matched layer for the absorption of electromagnetic waves: Journal of Computational Physics, **127**, 363–379, doi: [10.1006/jcph.1996.0181](https://doi.org/10.1006/jcph.1996.0181).
- Colella, P., J. Bell, N. Keen, T. Ligocki, M. Lijewski, and B. van Straalen, 2007, Performance and scaling of locally-structured grid methods for partial differential equations: Journal of Physics: Conference Series, **78**, doi: [10.1088/1742-6596/78/1/012013/meta](https://doi.org/10.1088/1742-6596/78/1/012013/meta).
- Collino, F., and C. Tsogka, 1998, Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media: Technical report RR-3471, INRIA.
- Darbas, M., and F. Louer, 2013, Analytic preconditioners for the iterative solution of elastic scattering problems: HAL, hal-00839653, 1–32.
- Dmitruk, P., L. P. Wang, W. H. Matthaeus, R. Zhang, and D. Seckel, 2001, Scalable parallel FFT for spectral simulations on a Beowulf cluster: Parallel Computing, **27**, 1921–1936, doi: [10.1016/S0167-8191\(01\)00120-X](https://doi.org/10.1016/S0167-8191(01)00120-X).
- Erlangga, Y. A., and R. Nabben, 2008a, On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian: Electronic Transactions on Numerical Analysis, **31**, 403–424.
- Erlangga, Y. A., and R. Nabben, 2008b, Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices: SIAM Journal on Matrix Analysis and Applications, **30**, 684–699, doi: [10.1137/060678257](https://doi.org/10.1137/060678257).
- Etienne, V., T. Tonello, P. Thierry, V. Berthoumieux, and C. Andreoli, 2014, Optimization of the seismic modeling with the time-domain finite-difference method: 84th Annual International Meeting, SEG, Expanded Abstracts, 3536–3540, doi: [10.1190/segam2014-0176.1](https://doi.org/10.1190/segam2014-0176.1).
- Intel, 2017, Intel® Math Kernel Library (Intel® MKL), <https://software.intel.com/en-us/intel-mkl>, accessed 01 May 2017.
- Kostin, V., V. Lisitsa, G. Reshetova, and V. Tcheverda, 2015, Local time-space mesh refinement for simulation of elastic wave propagation in multi-scale media: Journal of Computational Physics, **281**, 669–689, doi: [10.1016/j.jcp.2014.10.047](https://doi.org/10.1016/j.jcp.2014.10.047).
- Kostin, V., S. Solovev, H. Liu, and A. Bakulin, 2017, HSS cluster-based direct solver for acoustic wave equation: 87th Annual International Meeting, SEG, Expanded Abstracts, 4017–4021, doi: [10.1190/segam2017-17443086.1](https://doi.org/10.1190/segam2017-17443086.1).
- Li, Y., L. Métivier, R. Brossier, B. Han, and J. Virieux, 2015, 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver: Geophysics, **80**, no. 3, T101–T118, doi: [10.1190/geo2014-0480.1](https://doi.org/10.1190/geo2014-0480.1).
- Lisitsa, V., V. Tcheverda, and C. Botter, 2016, Combination of discontinuous Galerkin method with finite differences for simulation of elastic wave: Journal of Computational Physics, **311**, 142–157, doi: [10.1016/j.jcp.2016.02.005](https://doi.org/10.1016/j.jcp.2016.02.005).
- Neklyudov, D., V. Tcheverda, and I. Silvestrov, 2011, Frequency domain iterative solver for elasticity with semi-analytical preconditioner: 81st Annual International Meeting, SEG, Expanded Abstracts, 2931–2935.
- Oosterlee, C. W., C. Vuik, W. A. Mulder, and R.-E. Plessix, 2010, Shifted-Laplacian preconditioners for heterogeneous Helmholtz problems, in B. Koren and K. Vuik, eds., Advanced Computational Methods in Science and Engineering: Springer, 21–46.
- Operto, S., J. Virieux, P. Amestoy, J. L'Excellent, L. Giraud, and H. Hadj, 2007, 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study: Geophysics, **72**, no. 5, SM195–SM211, doi: [10.1190/1.2759835](https://doi.org/10.1190/1.2759835).
- Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain. Part I: Theory and verification in a physical scale model: Geophysics, **64**, 888–901, doi: [10.1190/1.1444597](https://doi.org/10.1190/1.1444597).
- Richtmyer, R., and K. Morton, 1957, Difference methods for initial value problems, 2nd ed.: Interscience Publishers.
- Rizzuti, G., and W. A. Mulder, 2015, A multigrid-based iterative solver for the frequency-domain elastic wave equation: 77th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: [10.3997/2214-4609.201413295](https://doi.org/10.3997/2214-4609.201413295).
- Saad, Y., 2003, Iterative methods for sparse linear systems, 2nd ed.: SIAM.
- Sirgue, L., J. Etgen, U. Albertin, and S. Brandsberg-Dahl, 2007, System and method for 3D frequency domain waveform inversion based on 3D time-domain forward modeling: U.S. Patent, 11/756, 384.
- Symes, W. W., 2008, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790, doi: [10.1111/j.1365-2478.2008.00698.x](https://doi.org/10.1111/j.1365-2478.2008.00698.x).
- Van der Vorst, H. A., 1992, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems: SIAM Journal on Scientific Computing, **13**, 631–644.

Virieux, J., S. Operto, H. Ben-Hadj-Ali, R. Brossier, V. Etienne, F. Sourber, L. Giraud, and A. Haidar, 2009, Seismic wave modeling for seismic imaging: *The Leading Edge*, **28**, 538–544, doi: [10.1190/1.3124928](https://doi.org/10.1190/1.3124928).  
Wang, S., M. V. de Hoop, J. Xia, and X. S. Li, 2012, Massively parallel structured multifrontal solver for time-harmonic elastic waves in 3-D anisotropic media: *Geophysical Journal International*, **191**, 346–366, doi: [10.1111/j.1365-246X.2012.05634.x](https://doi.org/10.1111/j.1365-246X.2012.05634.x).

Weisbecker, C., P. Amestoy, O. Boiteau, R. Brossier, A. Buttari, J.-Y. L'Excellent, S. Operto, and J. Virieux, 2013, 3D frequency-domain seismic modeling with a block low-rank algebraic multifrontal direct solver: 83rd Annual International Meeting, SEG, Expanded Abstracts, 3411–3416, doi: [10.1190/segam2013-0603.1](https://doi.org/10.1190/segam2013-0603.1).