Computers and Mathematics with Applications 86 (2021) 63–72

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Contraction operator transformation for the complex heterogeneous Helmholtz equation

Nikolay Yavich [a,b,*], Nikolay Khokhlov [b], Mikhail Malovichko [a,b], Michael S. Zhdanov [c,b]

[a] Center for Data-Intensive Science and Engineering, Skolkovo Institute of Science and Technology, Moscow, Russia
[b] Applied Computational Geophysics Lab, Moscow Institute of Physics and Technology, Dolgoprudny, Russia
[c] Department of Geology and Geophysics, University of Utah, Salt Lake City, UT, USA

ARTICLE INFO

Keywords:
Helmholtz equation
PDEs in connection with geophysics
Iterative methods
Preconditioning

ABSTRACT

An efficient solution of the three-dimensional Helmholtz equation is known to be crucial in many applications, especially geophysics. In this paper, we present and test two preconditioning approaches for the discrete problem resulting from the second order finite-difference discretization of this equation. The first approach combines shifted-Laplacian preconditioner with inversion of a separable matrix, corresponding to the horizontally-layered velocity model, using fast Fourier based transforms. The second approach is novel and involves a special transformation resulting in a preconditioner with a contraction operator (CO preconditioner). The two approaches have near the same arithmetical complexity; however, the second approach, developed in this paper, provides a faster convergence of an iterative solver as illustrated by numerical experiments and analysis of the spectral properties of the preconditioned matrices. Our numerical experiments involve parallel modeling of highly heterogeneous lossy and lossless media at different frequencies. We show that the CO-based solver can tackle problems with hundreds of millions of unknowns on a conventional cluster node. The CO preconditioned solver demonstrates a very moderate increase of iteration count with the frequency. We have conducted a comparison of the performance of the developed method versus open-source parallel sweeping preconditioner. The results indicate that, the CO solver is several times faster with respect to the wall-clock time and consumes substantially less memory than the code based on the sweeping preconditioner at least in the example we tested.

1. Introduction

Many applications require 3D numerical modeling of acoustic waves, especially geophysics. In this application, acoustic wave propagation is a truly multiscale phenomena since the wavelength is typically tens of *meters* while the domain of interest is typically tens of *kilometers* in diameter. Consequently, the discrete problem can easily involve 10^8 or 10^9 unknowns. The goal of this paper is to design, analyze, and test a novel preconditioning approach to numerical solution of the 3D complex heterogeneous Helmholtz equation, which governs time-harmonic visco-acoustic wave propagation. Even though we will neglect density variations, the stated problem is still challenging and computationally demanding.

There has been some progress with application of parallel sparse direct solvers to this problem, e.g. [1–3], despite massive memory allocations and quite time-consuming initialization (factorization). Since preconditioned iterative solvers are generally much more memory-economical and their initialization is much shorter (even momentary), we preferred investigating the opportunities of the latter.

The possibility to perform matrix–vector multiplication in $O(n)$ operations makes various types of Krylov iterative methods applicable to the discrete system. Earlier research suggested switching to normal equation [4] and using the conjugate gradient method. More recently, GMRes was applied in [5], in [6] IDR was used, but BiCGStab is a more common choice [7–9]. In geophysical scale modeling, the spectrum of the system matrix of the discretized equation contains eigenvalues with both positive and negative real parts. Consequently, the discrete solution is poorly approximated on Krylov subspaces [10], and thus iterative solvers tend to converge very slowly. In our work, we focused on the search of an efficient preconditioner rather than the best iterative solver. We thus picked the BiCGStab solver, which is most memory-economical among listed solvers and applicable to complex indefinite systems.

For a lossless medium, a wave-ray multigrid was suggested in [11], as well as sweeping preconditioner [12–14] and some other. However, the most commonly used preconditioning approach to this problem in the last 15 years was based on the shifted-Laplacian preconditioner (see

* Corresponding author at: Center for Data-Intensive Science and Engineering, Skolkovo Institute of Science and Technology, Moscow, Russia.
E-mail address: n.yavich@skoltech.ru (N. Yavich).

https://doi.org/10.1016/j.camwa.2021.01.018
Received 22 June 2020; Received in revised form 13 November 2020; Accepted 27 January 2021
Available online 19 February 2021

e.g. [15] and references therein). It consists of preconditioning the system matrix with a matrix involving an imaginary shift of the free term. From the physical standpoint, this matrix represents a discretization of a differential operator of visco-acoustic equation with high attenuation (damping). Various preconditioners were shown to be applicable to this damped problem, e.g. geometric [16,7,8,17,18] or algebraic [19] multigrid. The goal of this paper is to design and rigorously assess an alternative preconditioning approach.

The acoustic properties of the subsurface of the earth are known to vary predominantly in the vertical direction; therefore, a preconditioner can exploit this feature. Preconditioners that employ this feature can be designed using discrete separation of variables (discrete Fourier method) [9,6]. It can be noticed that, the performance of the latter preconditioners deteriorates when the velocity model gets stronger horizontal contrast. We present below a special transformation to the separation-of-variables preconditioner that results in a preconditioned system that is less sensitive to the velocity model contrast than the conventional formulation. Note that, contraction-type transformation was also considered for solving the visco-acoustic integral equations in [20].

This paper has the following outline. In Section 2, we present the governing equation and appropriate boundary conditions. We also discuss a transfer from the original system matrix to that of the damped problem, as well as the respective generalized eigenvalue problem. In Section 3, we design two preconditioners to the damped problem, while in Section 4 we analyze spectral properties of the arising preconditioned matrices. Finally, in Section 5, we compare performance of the designed solvers against each other, as well as against a publicly available code. The tests involve applications to geophysical modeling and both serial and parallel versions of the algorithms.

Within this paper, we use the right Cartesian coordinate system with x-axis directed east and z-axis directed downward. For vectors in $\mathbb{C}^n$, we denote the Euclidean scalar product as (u, v) and operator and vector norms as $\|\cdot\|$.

2. Differential and discrete problems

A time-harmonic acoustic wave of angular frequency ω propagating in a lossless continuous medium of primary velocity (speed of sound) $\widetilde{c} = \widetilde{c}(x, y, z) > 0$ and of constant density satisfies the heterogeneous Helmholtz equation,

$$-\Delta P - \frac{\omega^2}{\widetilde{c}^2} P = F, \tag{1}$$

where $P = P(x, y, z)$ is an acoustic pressure, and $F = F(x, y, z)$ is a pressure source. Whenever attenuation is present, it is usually modeled by complex-valued velocity,

$$\widetilde{c} = c - ic_2, \tag{2}$$

where c is primary velocity and c_2 accounts for energy absorption. In this case, $\widetilde{c}$ is expressed through physical characteristics of the medium. For example, using the widely adopted Kolsky–Futterman model [21,22] in its simplest version, $\widetilde{c}$ is determined by the following formula ([23, 5.92] up to the sign of the imaginary part),

$$\frac{1}{\widetilde{c}} = \frac{1}{c}\left(1 - \frac{i}{2Q}\right), \tag{3}$$

where the attenuation factor Q can be estimated from measured data, typically $Q \geq 10$. The Kolsky–Futterman model of seismic attenuation along with some models are rigorously analyzed in [24].

Combining (3) with (1), we receive,

$$-\Delta P - \frac{\omega^2}{c^2}\left(1 - \frac{1}{4Q^2} - \frac{i}{Q}\right) P = F. \tag{4}$$

Let us simplify notation and put $q = 1/Q$ and neglect the second order term $1/(4Q^2)$. Then we can rewrite the latter equation as the complex heterogeneous Helmholtz equation,

$$-\Delta P - \frac{\omega^2}{c^2}(1 - i\,q)\; P = F. \tag{5}$$

In seismic modeling, a solution of Eq. (5) is generally defined within a rectangular modeling domain D with its faces parallel to the coordinated axes, $D = [x_0, x_1] \times [y_0, y_1] \times [z_0,\ z_1]$. In this case, Eq. (5) is completed with zero Dirichlet boundary condition on the top face,

$$P = 0 \quad \text{on} \quad z = z_0, \tag{6}$$

while some absorbing boundary condition is commonly applied on the other five faces. We applied the PML boundary condition introduced in [25] in an extended domain $\widetilde{D}$, see the Appendix. The PML involves complex coordinate stretching factors $S_x(x)$, $S_y(y)$, and $S_z(z)$.

Let us cover domain $\widetilde{D}$ with a uniform orthogonal grid with step size h. Assuming that the discrete pressure values are attached to the cell centers, the discrete problem can be reduced to searching for discrete pressure $p \in \mathbb{C}^n$, where n is the number of cells, $n = n_x n_y n_z$. The second order cell-centered FD discretization of (A.4) completed with the introduced boundary conditions results in the following system of linear equations,

$$A\; p = f, \tag{7}$$

where A is a large sparse complex symmetric matrix, $f \in \mathbb{C}^n$ is the right-hand side. The structure of matrix A is similar to the structure of the differential operator in the left-hand side of Eq. (A.4),

$$A = L_{\text{PML}} - \omega^2 (1 - i\; q)\, \Sigma_{\text{PML}}, \tag{8}$$

where L_{PML} is the matrix of the discrete second derivatives with seven nonzero diagonals, Σ_{PML} is a diagonal matrix of averaged values of

$$S_x(x) S_y(y) S_z(z) / c\,(x, y, z)^2$$

over each cell.

We perform the first step in preconditioning the system matrix (8) by neglecting the PML coefficients, i.e. we introduce a matrix A_1 of the same size as A such that,

$$A_1 = L - \omega^2 (1 - i\; q)\, \Sigma, \tag{9}$$

where L is discrete negative Laplacian in $\widetilde{D}$ completed with Dirichlet boundary conditions, and Σ is a diagonal matrix of averaged squared slowness over each cell, $1/c\,(x, y, z)^2$.

The most popular approach to preconditioning Eq. (9) and related matrices is based on the shifted-Laplacian preconditioner, e.g., [15]. This approach consists of preconditioning A_1 with a matrix $\widetilde{A}$, involving strong artificial attenuation β, $0 < \beta < 1$,

$$\widetilde{A} = L - \omega^2 (1 - i\beta)\, \Sigma. \tag{10}$$

Basic understanding on the performance of such a preconditioner can be derived by studying the generalized eigenvalue problem,

$$A_1\; v = \lambda\; \widetilde{A}\; v. \tag{11}$$

In [26, Theorem 3.2], it is shown that its eigenvalues belong to a circle centered on the real axis and intersecting it at 1 and q/β, Fig. 1. However, we should admit that the spectrum gives only partial information on the convergence of an iterative solver like the GMRes. More detailed analysis was presented recently in [27].

The preconditioner (10) is one of the most perspective ways to design a preconditioned iterative solver, [28]. In the next section, we will focus on efficient solution of a linear system with matrix $\widetilde{A}$.

3. Design of the preconditioners to $\widetilde{\mathbf{A}}$

3.1. Preconditioner based on layered velocity model

Let us introduce a background velocity model, which varies with the vertical coordinate only, $c_b(z)$. Evidently, $c_b(z)$ can be defined in multiple ways, e.g. as a mean value of $c(x, y, z)$ within the z-plane. Further, we denote as Σ_b a diagonal matrix of the averaged squared

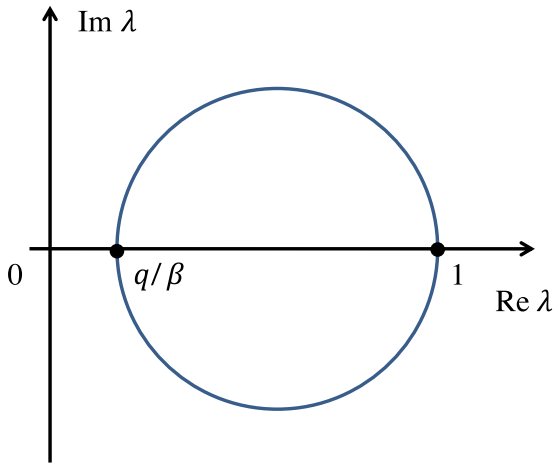


Fig. 1. The eigenvalues of (11) belong to the blue circle.

background slowness over each cell, $1/c_b(z)^2$. We define the following matrix,

$$\widetilde{A}_b = L - \omega^2 (1 - i\beta)\, \Sigma_b, \tag{12}$$

which corresponds to the visco-acoustic equation with the velocity of the background model. It is important to note that $\widetilde{A}_b$ is a separable matrix, thus the action of $\widetilde{A}_b^{-1}$ on a given vector can be computed in $O(n \log n)$ arithmetical operations via double fast sine transform completed by the tridiagonal matrix algorithm. This opens a way to using $\widetilde{A}_b$ as a preconditioner for (7).

For completeness, we present below the separation-of-variables algorithm to solve

$$\widetilde{A}_b u = g, \tag{13}$$

for arbitrary given vector g. The matrix has the following representation,

$$\widetilde{A}_b = I_z \otimes I_y \otimes L_x + I_z \otimes L_y \otimes I_x + \left(L_z - \omega^2(1 - i\beta)\Sigma_{bz}\right) \otimes I_y \otimes I_x, \tag{14}$$

where I_x, I_y and I_z are square identity matrices of dimension n_x, n_y, and n_z, respectively; L_x, L_y and L_z are square tridiagonal matrices of FD negative second derivative operators with respect to x, y, and z, respectively; Σ_{bz} is a diagonal matrix of dimension n_z such that $\Sigma_b = \Sigma_{bz} \otimes I_y \otimes I_x$.

Eigenvalues and eigenvectors of L_x and L_y are well-known. These matrices can be diagonalized with sine series (due to Dirichlet boundary condition on $\widetilde{D}$), and what is more important multiplication by the matrix of eigenvectors can be implemented with the fast discrete sine transform, reducing its complexity from $O(n_x^2)$ to $O(n_x \log(n_x))$. Denote their the eigenvector matrices as W_x and W_y and define

$$W = I_z \otimes W_y \otimes W_y. \tag{15}$$

The complexity of multiplication by W is $O(n_z n_x n_y \log(n_x n_y))$. Also note that W is orthogonal.

We can now use W to efficiently transform (13) to a system,

$$W \widetilde{A}_b W^T\, W u = W g. \tag{16}$$

The matrix $T = W \widetilde{A}_b W^T$ is block tridiagonal under an appropriate renumbering of the variables and thus (16) can be solved by $n_x n_y$ applications of the tridiagonal matrix algorithm, making the complexity of solving (16) $O(n_z n_x n_y)$.

To summarize, solution of (13) can be performed in $O(n_z n_x n_y \log(n_x n_y)) + O(n_z n_x n_y) = O(n \log(n))$ arithmetical operations. The algorithm is also straightforward to parallelize since both multiple sine transforms and tridiagonal solvers are applied independently.

With a few minor differences, this has been presented in [6,29]. We will refer to this approach as a Green's function (GF) preconditioner, since $\widetilde{A}_b^{-1}$ is a discrete Green's function of the visco-acoustic equation for background model. Since $\widetilde{A}_b$ incorporates vertical variation of the velocity model only, this approach will possibly degrade in a case of models with high horizontal contrast. Let us look at an alternative approach that will partially mitigate this problem.

3.2. Auxiliary operator G_b

In this subsection, we will introduce an auxiliary operator G_b such that its norm is unity. For that, consider two systems of equations with matrices $\widetilde{A}$ and $\widetilde{A}_b$ and some right-hand side g,

$$\widetilde{A}\, u = g, \tag{17}$$

and

$$\widetilde{A}_b\, u_b = g. \tag{18}$$

Within a preconditioned iterative solver, the current residual will play the role of g. We define also an anomalous response $u_a = u - u_b$ and a diagonal matrix $\Sigma_a = \Sigma - \Sigma_b$. Subtracting the two latter systems of equations, it is easy to obtain a system for u_a,

$$\widetilde{A}_b\, u_a = \omega^2 (1 - \beta i)\, \Sigma_a (u_a + u_b). \tag{19}$$

Performing a scalar multiplication of this system by u_a and taking the imaginary part, we arrive at the following equation:

$$\beta \left(u_a, \Sigma_b u_a\right) = \mathrm{Im}\ \left((1 - \beta i)\ (u_a, \Sigma_a (u_a + u_b))\right), \tag{20}$$

or equivalently,

$$\beta \left(u_a, \Sigma_b u_a\right) + \mathrm{Re}\ \left(i\,(1 - \beta i)\ (u_a, \Sigma_a (u_a + u_b))\right) = 0. \tag{21}$$

This identity will be used to transform Eq. (17) into a system with a contraction operator. For that, let us rewrite the identity as follows,

$$\left(\Sigma_b^{\frac{1}{2}} u_a, \Sigma_b^{\frac{1}{2}} u_a\right) + 2\ \mathrm{Re}\ \left(\Sigma_b^{\frac{1}{2}} u_a,\ \frac{i(1 - \beta i)}{2\beta} \Sigma_b^{-\frac{1}{2}} \Sigma_a (u_a + u_b)\right) = 0. \tag{22}$$

In fact, the last equality is of the form,

$$(s, s) + 2\ \mathrm{Re}\ (s,\ v) = 0, \tag{23}$$

with respective s and v. Applying the polarization identity, we can rearrange it as follows,

$$\|s + v\|^2 = \|v\|^2. \tag{24}$$

Let us make a closer look at s and apply Eq. (19),

$$s = \Sigma_b^{\frac{1}{2}} u_a = \omega^2 (1 - \beta i)\, \Sigma_b^{\frac{1}{2}} \widetilde{A}_b^{-1} \Sigma_a \left(u_a + u_b\right) = -2\beta i \omega^2 \Sigma_b^{\frac{1}{2}} \widetilde{A}_b^{-1} \Sigma_b^{\frac{1}{2}} v. \tag{25}$$

We thus receive,

$$\left\| (-2\beta i \omega^2 \Sigma_b^{\frac{1}{2}} \widetilde{A}_b^{-1} \Sigma_b^{\frac{1}{2}} + I) v \right\|^2 = \|v\|^2. \tag{26}$$

Since g was selected arbitrary, all variables depending on g including v are also arbitrary. We consequently can conclude that the norm of the operator in the left-hand side is equal to unity.

Denote this operator as G_b,

$$G_b = -2\beta i \omega^2 \Sigma_b^{\frac{1}{2}} \widetilde{A}_b^{-1} \Sigma_b^{\frac{1}{2}} + I. \tag{27}$$

3.3. Preconditioner based on a contraction transformation

Let us return to equation system (17). We will now show that it can be equivalently transformed into a system with the following structure,

$$(I - C)\, u' = g', \tag{28}$$

where $C = G_b K_2 K_1^{-1}$ and K_1 and K_2 are diagonal matrices,

$$K_1 = \frac{1}{2\sqrt{\beta}} \left(2\beta \Sigma_b + i\,(1 - i\beta)\, \Sigma_a\right) \Sigma_b^{-\frac{1}{2}},\ K_2 = \frac{i}{2\sqrt{\beta}} (1 - i\beta)\, \Sigma_a \Sigma_b^{-\frac{1}{2}}, \quad (29)$$

and $u' = K_1 u$, $g' = \sqrt{\beta} \Sigma_b^{\frac{1}{2}} \widetilde{A}_b^{-1} g$. In fact, when we switch form u' and g' to u and g, we obtain,

$$\left(K_1 - G_b K_2\right) u = \sqrt{\beta} \Sigma_b^{\frac{1}{2}} \widetilde{A}_b^{-1} g. \quad (30)$$

Multiplying both sides of the last equation on the left by $1/\sqrt{\beta} \Sigma_b^{-\frac{1}{2}}$ and substituting expressions for K_1 and K_2, we receive the following system,

$$\left(I - (1 - i\beta)\, \omega^2 \widetilde{A}_b^{-1} \Sigma_a\right) u = \widetilde{A}_b^{-1} g, \quad (31)$$

which differs from (17) only by the left multiplication by $\widetilde{A}_b^{-1}$.

3.4. Proof of the contraction property

Now, let us estimate the norm of C, $\|C\| = \left\|G_b K_2 K_1^{-1}\right\| \le \left\|K_2 K_1^{-1}\right\|$. The product $K_2 K_1^{-1}$ is equal to a diagonal matrix,

$$(1 - i\beta) \left(\Sigma\ \Sigma_b^{-1} - 1\right) \left((\beta + i)\, \Sigma\ \Sigma_b^{-1} + (\beta - i)\, I\right)^{-1}. \quad (32)$$

Its norm is the magnitude of the largest diagonal entry. Thus, calculation of $\left\|K_2 K_1^{-1}\right\|$ is equivalent to:

$$\left\|K_2 K_1^{-1}\right\| = \max_{\xi} \left| \frac{(1 - i\beta)\,(\xi - 1)}{(\beta + i)\,\xi + \beta - i} \right|, \quad (33)$$

where ξ runs through all of the diagonal values of $\Sigma\ \Sigma_b^{-1}$. The span of these values is straightforward to estimate. Noting the definition of Σ and Σ_b, we can conclude that $a \le \xi \le b$, where

$$a = \min_{\widetilde{D}} \frac{c_b\,(z)^2}{c\,(x, y, z)^2}, \quad b = \max_{\widetilde{D}} \frac{c_b\,(z)^2}{c\,(x, y, z)^2}. \quad (34)$$

Without loss of generality, we further assume $0 < a \le 1 \le b$. Now, the optimization problem (33) can be directly solved, and we obtain,

$$\|C\| \le \sqrt{1 + \beta^2} \max \left(\frac{1 - a}{\sqrt{(1 - a)^2 + \beta^2\,(1 + a)^2}}, \frac{b - 1}{\sqrt{(b - 1)^2 + \beta^2\,(b + 1)^2}} \right). \quad (35)$$

Finally, let us look at the square of the latter estimate assuming for a moment that maximum is attained in the second term,

$$\frac{\left(1 + \beta^2\right)(b - 1)^2}{(b - 1)^2 + \beta^2\,(b + 1)^2}. \quad (36)$$

Dividing the numerator and denominator by $(b - 1)^2$, we can easily see that this fraction is less than unity. Similarly, it can be shown that if the maximum is attained in the first term, then the estimate is also less than unity.

We ultimately conclude that C is a contraction operator for media of any contrast, though its norm tends to unity in the limiting cases, i.e. $0 < a \ll 1$ or $b \gg 1$, corresponding to high horizontal contrast. We will refer the transformation (28) as a contraction operator (CO) preconditioner.

To summarize the above discussion, we note that, the GF preconditioner results in a preconditioned matrix of the following form,

$$\widetilde{A}_b^{-1}\ A, \quad (37)$$

while the CO preconditioner results in a preconditioned matrix,

$$\sqrt{\beta} \Sigma_b^{\frac{1}{2}}\ \widetilde{A}_b^{-1}\ A\ K_1^{-1}, \quad (38)$$

Let us emphasize that all the transformations involved here can be efficiently implemented since they involve only diagonal scaling and multiplication by $\widetilde{A}_b^{-1}$ which has a complexity of $O(n \log n)$.

4. Analysis of the preconditioners to $\widetilde{\mathbf{A}}$

It is of practical interest to analyze and compare spectral properties of matrices (37) and (38). Note that, our development is essentially based on two steps: (1) preconditioning of A with $\widetilde{A}$ and (2) transition from $\widetilde{A}$ to either (12) or (28). The properties of application of the first step were already discussed, and this step is common to both GF and CO preconditioning approaches. We thus can concentrate on the search of the spectrum resulting from the second step. This is presented in the current section.

First, let us look at the spectral properties of the GF preconditioned matrix.

$$\widetilde{A} u = \lambda\ \widetilde{A}_b u. \quad (39)$$

A scalar multiplication by u results in the following expression:

$$(u, Lu) - \omega^2\,(1 - i\beta)\,(u, \Sigma u) = \lambda\,(u, Lu) - \omega^2 \lambda\,(1 - i\beta)\left(u, \Sigma_b u\right). \quad (40)$$

Next, we substitute $\lambda = \lambda_1 + i\lambda_2$, with real λ_1 and λ_2,

$$\left(1 - \lambda_1 - i\lambda_2\right)(u, Lu) - \omega^2\,(1 - i\beta)\,(u, \Sigma u) + \omega^2(\lambda_1 + i\lambda_2)\,(1 - i\beta)\left(u, \Sigma_b u\right) = 0. \quad (41)$$

The last equation can be written for the real and imaginary parts separately:

$$\begin{aligned} &\left(1 - \lambda_1\right)(u, Lu) - \omega^2\,(u, \Sigma u) + \omega^2\left(\lambda_1 + \beta\lambda_2\right)\left(u, \Sigma_b u\right) = 0, \\ &-\lambda_2\,(u, Lu) + \omega^2 \beta\,(u, \Sigma u) + \omega^2(\lambda_2 - \beta\lambda_1)\left(u, \Sigma_b u\right) = 0. \end{aligned} \quad (42)$$

Multiplying the first equation by λ_2, and the second by $\left(1 - \lambda_1\right)$, and summing them up, we receive,

$$\begin{aligned} &-\lambda_2\,(u, \Sigma u) + \lambda_2\left(\lambda_1 + \beta\lambda_2\right)\left(u, \Sigma_b u\right) + \\ &\quad \left(1 - \lambda_1\right)\beta\,(u, \Sigma u) + \left(1 - \lambda_1\right)(\lambda_2 - \beta\lambda_1)\left(u, \Sigma_b u\right) = 0. \end{aligned} \quad (43)$$

Employing the nonsingularity of Σ_b, we can divide the equation by $\left(u, \Sigma_b u\right)$. Let us introduce $\xi = (u, \Sigma u) / \left(u, \Sigma_b u\right)$; note that $a \le \xi \le b$. With this notation, we obtain,

$$-\lambda_2 \xi + \lambda_2\left(\lambda_1 + \beta\lambda_2\right) + \left(1 - \lambda_1\right)\beta\xi + \left(1 - \lambda_1\right)(\lambda_2 - \beta\lambda_1) = 0. \quad (44)$$

After completing the squares, we receive an equation for a family of circles on the complex plain,

$$\left(\lambda_1 - \frac{\xi + 1}{2}\right)^2 + \left(\lambda_2 - \frac{\xi - 1}{2\beta}\right)^2 = \frac{(\xi - 1)^2}{4}\left(1 + \frac{1}{\beta^2}\right). \quad (45)$$

All of the circles pass through the point $\lambda = 1$. The tangent line at that point to all of the circles in the family is given by $\lambda_2 = -\beta(\lambda_1 - 1)$. The circles that correspond to $1 < \xi \le b$ lie above the tangent line. Moreover, circles that correspond to larger ξ enclose those corresponding to smaller ξ. Thus the eigenvalues produced by low-velocity inclusions belong to the disk which corresponds to the largest $\xi = b$. The high-velocity inclusions correspond to $a \le \xi < 1$. The respective eigenvalues belong to the disk which corresponds to the smallest $\xi = a$. An example is given in Fig. 2, where we took $a = 0.1$, $b = 2.5$ and $\beta = 0.5$. The discussed spectrum belongs to the blue and red disks.

Let us examine the spectral properties of the CO preconditioned matrix,

$$(I - C)\, u = \lambda u. \quad (46)$$

The corresponding eigenvalues evidently belong to the disk centered at 1 and of radius $\|C\|$, notice (35),

$$|\lambda - 1| \le \|C\|. \quad (47)$$

The respective area is shaded in Fig. 2.

A comparison of the two spectra in this figure indicates the following. The GF preconditioner is sensitive to the presence of both low- and high-velocity inclusions. Also the spectrum spans to the left half-plane. On the other hand, the spectrum of the CO preconditioned matrix is

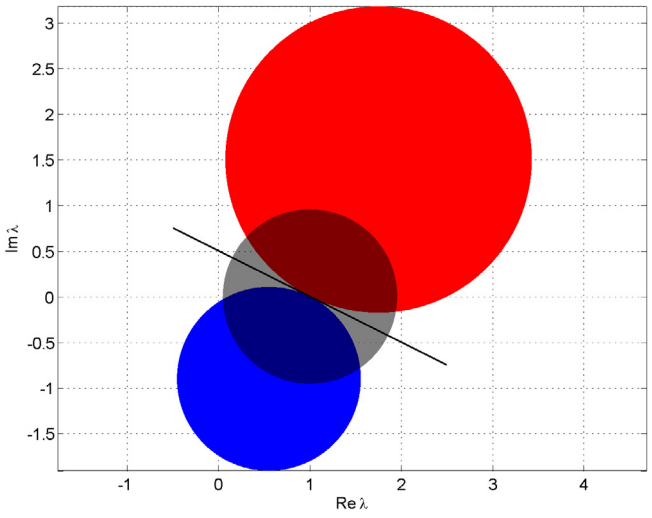


Fig. 2. The eigenvalues of (39) belong to either the blue or red disks. The black line indicates the tangent line to the two disks. The eigenvalues of (46) belong to the shaded disk. In this example, $a = 0.1$, $b = 2.5$ and $\beta = 0.5$.

always in the right half-plane and it is only sensitive to the maximum horizontal velocity contrast. We thus expect a faster convergence of an iterative solver applied to the CO preconditioned matrix than that of the GF preconditioned matrix when both types of inclusions are present.

5. Numerical experiments

In this section, we will compare the performance of the CO preconditioner against the GF preconditioner applied to the system (7), as well as against a publicly available code.

We have implemented an OpenMP multithread version of the algorithms in C++. We used FFTW3 library [30]. This section is devoted to assessment of accuracy and performance of the modeling code. Sections 5.1 and 5.2 present computations performed on a single thread, while multithreaded tests are presented in Section 5.3.

5.1. Performance assessment

A velocity model for this test is illustrated in Fig. 3. The model occupies the volume $[-1;1] \times [-1;1] \times [0;1]$ km^3 and is formed by three curved layers with velocities 1500, 3000, and 6000 m/s, respectively [31]. The medium was assumed to have attenuation $q = 0.05$. To simulate response at 10 Hz, the volume was covered by a FD grid $200 \times 200 \times 100$ with a step size of 10 m. The grid was further completed with 30 PML layers in the five directions. The overall size of the problem was near 9 million discrete unknowns. An acoustic source was located at (5, 5, 255) m.

Fig. 4 shows convergence of the BiCGStab solver preconditioned with the diagonal GF and CO preconditioners, assuming $\beta = 0.5$. The computations were performed on 2.8 GHz Intel (R) Xeon (R) X5660 CPU. To achieve a residual tolerance of $\varepsilon = 1e-6$, the CO preconditioned solver performed 84 iterations which required 13.5 min; the GF preconditioned solver performed 208 iterations and required 31.5 min; the diagonal preconditioned solver did not converge. We have concluded that the use of CO transformation resulted in a speedup of 2.5 times with respect to iterations and 2.3 times with respect to time versus the GF preconditioner. Since both high- and low-velocity inclusions were present in this model ($a = 0.25$, $b = 4$), the faster convergence of the CO preconditioned system was expected. The peak memory usage by both GF and CO preconditioned solvers was near 3.5 Gb, and that included system matrix storage, BiCGStab auxiliary vectors, as well as data needed by the preconditioners. It should be noted that, the memory consumption can be roughly halved if the system matrix is computed on-the-fly.

Table 1
Convergence of the BiCGStab preconditioned with GF and CO for the three-layered model for different values of attenuation q, $\beta = 0.5$ at 10 Hz.

q	GF solver iteration count, CPU time (min)	CO solver iteration count, CPU time (min)
0.10	169, 25.6	66, 10.6
0.05	208, 31.5	84, 13.5
0.01	369, 55.7	173, 27.6

Table 1 shows iteration count and execution time of the BiCGStab solver leveraged with the GF and CO preconditioners, required to achieve the residual tolerance of $1e-6$. Various values of attenuation q were tested. We observe that both preconditioned iterative solvers tend to improve convergence for higher attenuation. Also, the CO preconditioner provided 2 to 2.5 times faster convergence in all of cases. Finally, Fig. 5 shows the computed responses obtained by the two solvers. They are essentially identical. This illustrates the fact that the two preconditioners are based on equivalent transforms of the original equation system.

Finally, we investigated the performance of the preconditioners at different frequencies for this velocity model. A sequence of three grids corresponding to frequencies 10, 15, and 20 Hz was generated keeping approximately 15 grid steps per shortest wavelength. The tests were performed assuming $\beta = 0.5$, $q = 0.05$, and $\varepsilon = 1e-6$. We observe from the results (Table 2) that the GF preconditioner loses robustness at higher frequencies. This complies with the experiments published in [6,9]. At the same time, the CO preconditioned solver is robust and demonstrates a very moderate iteration count increase with the frequency (1.5 times with the frequency increase 2 times).

5.2. Overthrust model

In this section, we compare the two preconditioned iterative solvers using SEG/EAGE Overthrust model [32]. A part of this model has been extracted, as shown in Fig. 6. This model includes low- and high-velocity structures as well as strong horizontal contrast ($a = 0.32$, $b = 2.83$). We further assume attenuation to be $q = 0.05$ and $\beta = 0.5$.

We modeled acoustic responses at frequencies of 10, 15, 20, and 30 Hz from a source located 262 m below the surface of the model. The model was covered with different uniform grids with sizes shown in Table 3, depending on the frequency. We completed the grids further with PML layers as in the previous experiments.

The performance of the two preconditioned solvers is presented in Table 3 for the tolerance of $1e-6$. For the least frequency of 10 Hz, the performance of the solvers is close, though the CO solver is 30% faster. At 15 Hz, the CO solver is approximately 3 times faster both with respect to CPU time and iteration count, Fig. 7. At the higher frequencies, the GF solver degrades and does not seem to converge, while the CO solver still produce a robust result with some moderate iteration count increase. With the CO solver, we managed to solve a problem with 211 M unknowns on a single computer node.

5.3. Comparison of the CO solver versus the parallel sweeping preconditioner

In this section, we compare performance of the CO solver with an open-source solver based on the parallel sweeping preconditioner (PSP), introduced in [12] https://github.com/poulson/PSP.

Both our code and the PSP code use the standard 2nd-order cell-based discretization but there are differences in the PML setup. Both codes use the same PMLs formulation (compare [12, eq. (2.1)] and (A.2)–(A.4)). The different signs of the imaginary part arise from the different definitions of the Helmholtz differential operator [33]. Also,

Table 2
Performance of the preconditioned iterative solvers at different frequencies for the three curved layers model.

Frequency (Hz)	Step size (m)	Internal grid	Overall discrete problem size	Peak memory usage	GF solver Iteration count, CPU time (min)	CO solver Iteration count, CPU time (min)
10	10	200 × 200 × 100	8.8 M	3.5 Gb	208, 31.5	84, 13.5
15	7	285 × 285 × 140	20 M	7.9 Gb	> 200, > 4 h	97, 111
20	5	400 × 400 × 200	49 M	20 Gb	> 1200, > 50 h	125, 318

Table 3
Performance of the preconditioned iterative solvers at different frequencies for the overthrust model.

Frequency (Hz)	Step size (m)	Internal grid	Overall discrete problem size	Peak memory usage	GF solver Iteration count, CPU time (min)	CO solver Iteration count, CPU time (min)
10	15	210 × 210 × 130	12 M	4.5 Gb	81, 56 min	62, 43.6 min
15	10	320 × 320 × 200	33 M	13 Gb	192, 6.1 h	64, 2.1 h
20	8	400 × 400 × 250	59 M	24 Gb	n/a	71, 5.0 h
30	5	640 × 640 × 400	211 M	86 Gb	n/a	98, 27.6 h

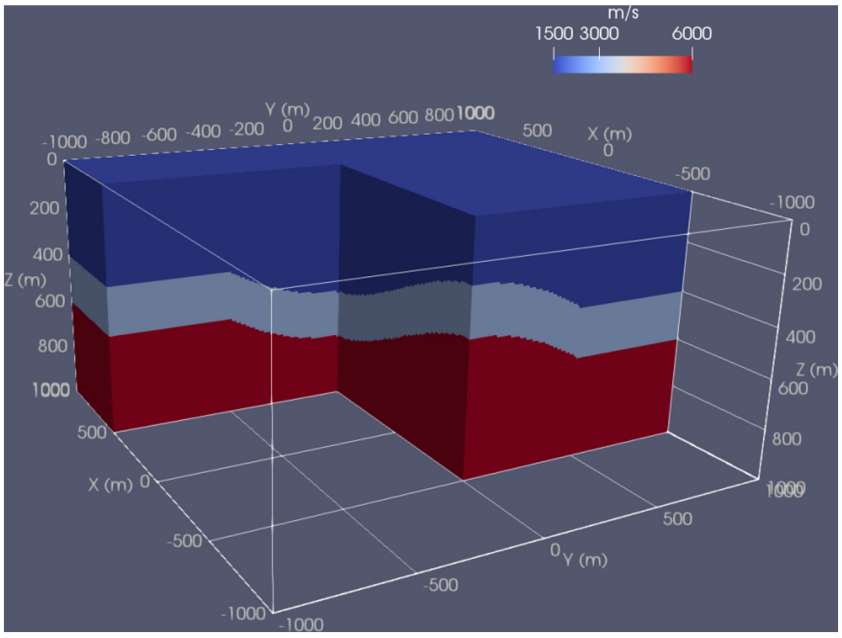


Fig. 3. Velocity model formed by three curved layers with velocities of 1500, 3000 and 6000 m/s.

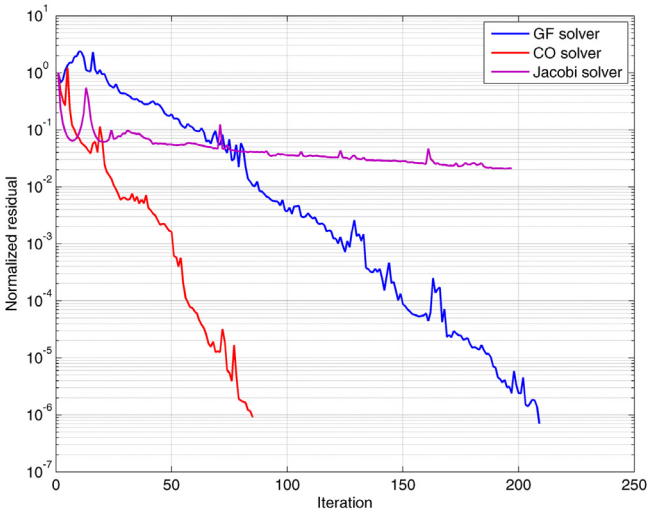


Fig. 4. Convergence of the BiCGStab solver preconditioned with the diagonal (Jacobi), Green-function (GF) and contraction-operator (CO) preconditioners with $\beta = 0.5$ at 10 Hz for the model with three curved layers.

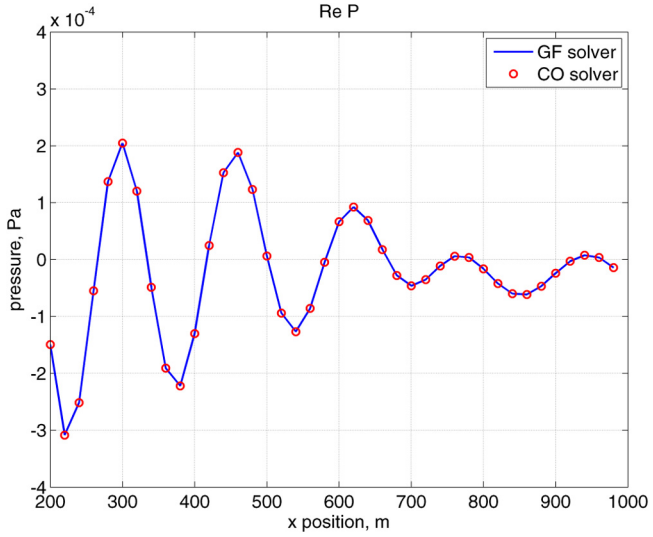


Fig. 5. Acoustic response for the three-layer model for GF and CO preconditioned solvers for attenuation $q = 0.05$ and frequency 10 Hz along the line $y = 0$ and $z = 255$.

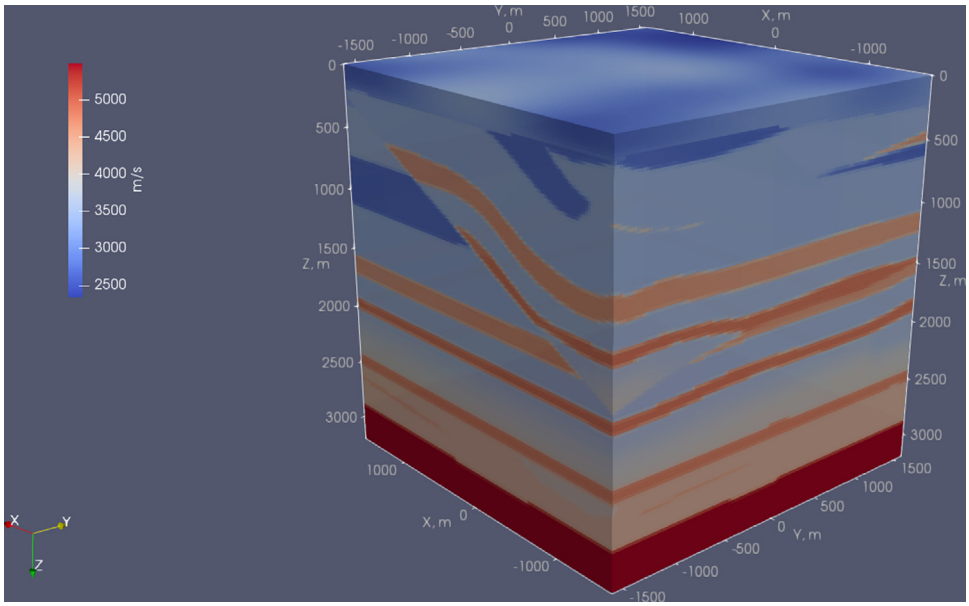


Fig. 6. A block of SEG/EAGE overthrust model.

PSP code treats several outermost cell layers as PMLs, whereas our code adds additional PMLs on top of existing grid. We have also noted that the PSP runtime significantly depends on the number of cells marked as PMLs, meaning to the user could not increase the thickness of PML. Thus, we expect some discrepancy in the two solutions due to remnant reflections from the outer boundary of PML. Both programs solve the system of linear equations iteratively. We used the BiCGStab solver, whereas PSP applied the restarted GMRes. In the examples given below the number of GMRes vectors was set to 20. The numerical complexity of the matrix–vector multiplication is the same for both codes: $O(n \log n)$. In addition, the PSP code performs an initialization step, which has complexity of $O(n^{4/3})$. Finally, the two programs use different parallelization strategies. Our code runs on several OpenMP threads, while the PSP code uses an MPI-based parallelization.

We used the full SEG/EAGE Overthrust model (Fig. 8). The physical dimension of the model was 20 × 20 × 4.65 km. The model was discretized by a 319 × 319 × 75 FD grid with the uniform step size, $h = 62.5$ m, in all directions. The source frequency was set to 3.175 Hz. The viscosity was set to zero in both programs while β was set to 0.5 in our program. The PML was applied to all six faces of the computational domain. The PSP was running with 5 PMLs and $\max \sigma(x) = 7.5$, whereas our software used 10 PMLs with $\alpha = 35$. A distributed source function $f(x)$ was set up as follows (see [13, p. C206]):

$$f(x) = -n_x e^{-10 n_x \|x - x_0\|^2}, \tag{48}$$

where x is the location of the cell center, x_0 is the location of the source center, n_x is the number of cells in the x-direction. The source was centered at $x_0 = (10000, 10000, 2325)$ m (Fig. 9).

Both the right-hand side of the system of FD equations and the velocity distribution were initially discretized by the PSP software and then imported in our code to ensure the same FD approximation. The tolerance of an iterative solver was set to 1e-5 in both codes.

In this test, we used a shared-memory computational system equipped with Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10 GHz, 12 cores, and 256 Gb RAM. The PSP code was running on 4 MPI processes, each attached to a single core. Our code was running on 4 OpenMP processes. Although the codes employ different parallelization schemes, they use the same set of physical CPU cores. We thus assume a moderate impact of this discrepancy on computational results. The PSP solver performed twice fewer iterations comparing to the CO solver. However, the overall wall-clock time for CO solver was 5.5 time smaller than

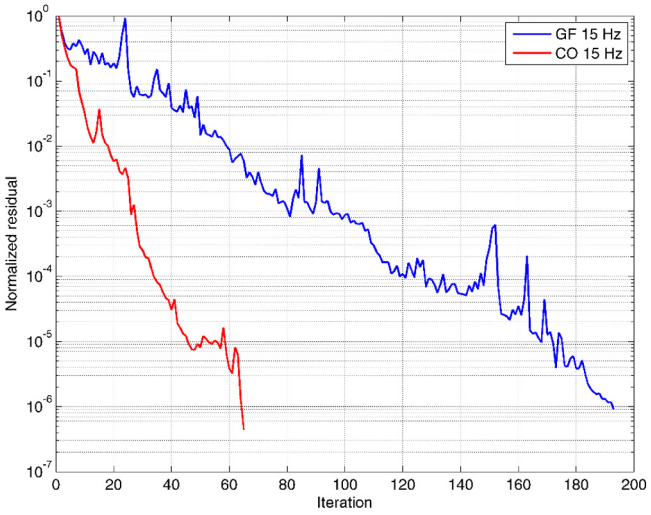


Fig. 7. Convergence of the BiCGStab iterative solver preconditioned with the GF and CO preconditioners for the overthrust model at 15 Hz.

that of the PSP, also the CO solver consumed 15 times less memory, as summarized in Table 4. On the other hand, we should note the tested example was possibly somewhat unfavorable for the PSP solver, since the approach is presumably beneficial at higher frequencies, multiple right-hand sides, and large number of parallel processes [13].

The responses are compared in Figs. 10 and 11. We observe a reasonable match between the two solutions, while the minor differences could be attributed to remnant reflections from the top and bottom faces of the computational domain, which are just 1–2 wavelength away from the source (note the difference in the wavefield depicted in Figs. 10(a) and 10(b) near the top and bottom boundaries). These faces generate remnant reflections which are not fully suppressed by the PMLs.

6. Conclusions and future applications

In this paper we have presented, analyzed and tested two preconditioning approaches to the algebraic problem resulting from the

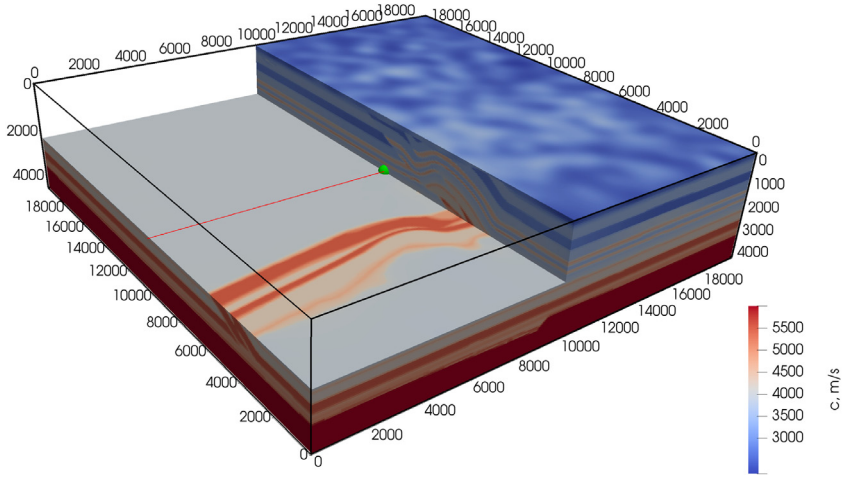


Fig. 8. The full Overthrust model with the source and receivers. The velocity in m/s is shown in color. The green ball marks the center of the source. The red line denotes the locations of the receivers.

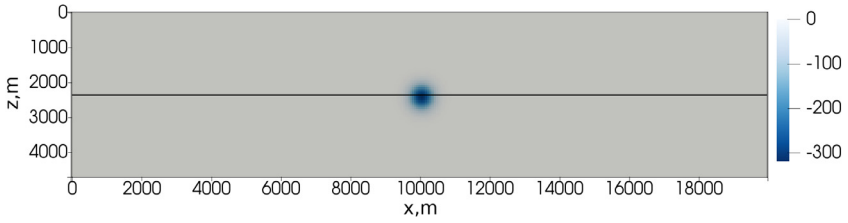
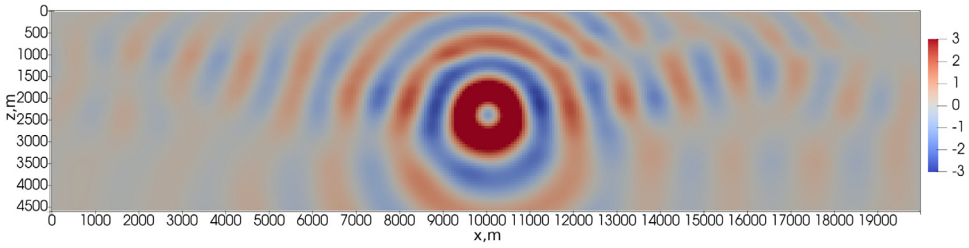
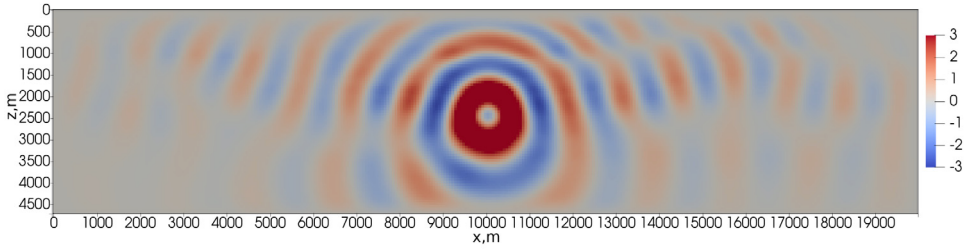


Fig. 9. The source function in the xz cross-section. The black line depicts the positions of the receivers.



(a) CO solver solution



(b) PSP solver solution

Fig. 10. Comparison of the CO (a) and PSP (b) solutions in the xz-plane passing through the center of the source. The real part of the pressure response is shown in color.

discretization of the complex heterogeneous Helmholtz equation. The first approach combines shifted-Laplacian preconditioner with the inversion of a separable matrix, corresponding to the horizontally layered velocity model, using fast Fourier transforms (GF preconditioner). The second approach is based on a special transformation resulting in a preconditioner with a contraction operator (CO preconditioner). The concept is a generalization to the case of the acoustic field of the contraction operator preconditioner developed for modeling low-frequency electromagnetic fields [34]. The two approaches have near the same arithmetical complexity; however, the second approach provides a faster convergence of an iterative solver as illustrated by numerical experiments. Our analysis of the preconditioned matrices spectra has indicated that a faster convergence of the iterative solvers applied

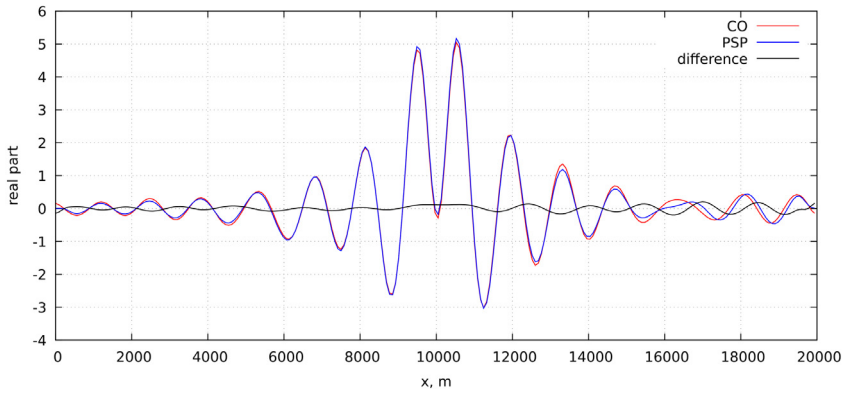


Fig. 11. Comparison of PSP and CO solutions at 3.175 Hz along the line depicted in Fig. 9. The real part of the pressure response is shown.

Table 4
Performance of the CO and PSP preconditioned iterative solvers.

	CO solver	PSP solver
Parallelization	4 OpenMP threads	4 MPI ranks, (1 core per MPI rank)
Peak memory usage, Gb	7.6 (total)	29 (per MPI rank)
Initialization	0	21.8 min
Iterative solver	BiCGStab	GMRes(20)
Iteration count	136	65
Iterative solver wall-clock time	7.9 min	21.4 min

to the CO preconditioned system is expected when both high- and low-velocity inclusions are present in the model.

Our numerical experiments involved modeling the time-harmonic visco-acoustic waves for complex high-contrast velocity models at different frequencies. While the GF preconditioner lost robustness at higher frequencies, the CO preconditioned solver performed well for all frequencies considered and it demonstrated a very moderate increase of the iteration count (1.5 times with the frequency increase two times). We managed to solve a problem with 211 M unknowns on a single compute node.

The developed approach is easily parallelizable for shared memory architecture. Our numerical experiments included tests with OpenMP multi-thread version of the solver. We compared performance of our code versus open-source parallel sweeping preconditioner. The comparison was performed on a complex seismic velocity model and involved near 10 M unknowns. In this test, CO solver converged 5.5 times faster with respect to the wall-clock time, and consumed 15 times less memory than the open-source program.

Note (37) differs from (38) only by two specifically chosen diagonal scaling factors. Evidently there are multiple options available. We assume that another choice of the factors is possible that would improve the approach.

The designed approach could be further leveraged by incorporating the PML directly into the preconditioner, [35,36]. We expect that this will somewhat reduce the iteration count. It also should be noted that grid size drastically impacts FFT performance and ultimately CPU time of the CO solver [37]. Optimal grid dimensions could be selected at every 8-10 points, reducing modeling time dramatically. We have considered the second order cell-centered FD discretization, though the developed strategy could be directly extended to the higher order discretizations as well. Future work will be directed at further generalization of the CO solver to problems with variable density and attenuation as well as for application it the full-waveform inversion.

Acknowledgments

The authors acknowledge the University of Utah′s Consortium for Electromagnetic Modeling and Inversion (CEMI) for support of this research and permission to publish. We would like to acknowledge the Complex for Simulation and Data Processing for Mega-Science Facilities at NRC Kurchatov Institute, http://ckp.nrcki.ru/ as well as Skoltech high performance computing cluster, Zhores [38], for providing the computing resources that have contributed to the results reported herein.

This research was supported by the Russian Science Foundation, Project No. 18-71-10071.

The authors appreciate comments and recommendations made by the anonymous reviewer.

Appendix. PML details

In this section, we describe the PML boundary conditions which are used to suppress reflections from the computational domain boundaries.

To define the PML boundary condition, let us consider two layers, $x_0 - \delta \le x \le x_0$ and $x_1 \le x \le x_1 + \delta$, to the west and east of domain D, and assume $\delta \ge 0$. In these layers, the second derivative with respect to x in Eq. (5)

$$-\frac{\partial^2 P}{\partial x^2}, \tag{A.1}$$

is substituted with the following expression,

$$-\frac{1}{S_x(x)}\frac{\partial}{\partial x}\left(\frac{1}{S_x(x)}\frac{\partial P}{\partial x}\right), \tag{A.2}$$

where the complex stretching factor $S_x(x)$ takes the form

$$S_x(x) = 1 - \frac{i\alpha}{\omega}\frac{(x-x_0)^2}{\delta^2},\ x \in [x_0-\delta, x_0],$$
$$S_x(x) = 1 - \frac{i\alpha}{\omega}\frac{(x-x_1)^2}{\delta^2},\ x \in [x_1, x_1+\delta]. \tag{A.3}$$

At $x = x_0 - \delta$ and $x = x_1 + \delta$ we assume $P = 0$. We further define $S_x(x) = 1$ for $x_0 \le x \le x_1$ so that $S_x(x)$ is a continuous function in $[x_0 - \delta, x_1 + \delta]$. Similarly, PML is introduced outside of other faces (except the top face) of the computational domain, and we define $S_y(y)$ and $S_z(z)$. Parameters α and δ are estimated heuristically. We will denote an extended modeling domain as $\widetilde{D}$, $\widetilde{D} = [x_0 - \delta, x_1 + \delta] \times [y_0 - \delta, y_1 + \delta] \times [z_0,\ z_1 + \delta]$.

We can multiply Eqs. (5), (A.2) by $S_x(x)\,S_y(y)\,S_z(z)$ so that we obtain an equation with a symmetric operator,

$$-S_y(y)\,S_z(z)\frac{\partial}{\partial x}\left(\frac{1}{S_x(x)}\frac{\partial P}{\partial x}\right) - S_x(x)\,S_z(z)\frac{\partial}{\partial y}\left(\frac{1}{S_y(y)}\frac{\partial P}{\partial y}\right)$$
$$-S_x(x)\,S_y(y)\frac{\partial}{\partial z}\left(\frac{1}{S_z(z)}\frac{\partial P}{\partial z}\right) - \omega^2\frac{S_x(x)\,S_y(y)\,S_z(z)}{c^2}(1 - i\ q)\,P$$
$$= S_x(x)\,S_y(y)\,S_z(z)\,F. \tag{A.4}$$

This equation is the starting point in our development.

References

[1] S. Operto, J. Virieux, P. Amestoy, J.-Y. L'Excellent, L. Giraud, H.B.H. Ali, 3d finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study, Geophysics 72 (5) (2007) SM195–SM211, http://dx.doi.org/10.1190/1.2759835.

[2] R. Brossier, V. Etienne, S. Operto, J. Virieux, Frequency-domain numerical modelling of visco-acoustic waves based on finite-difference and finite-element discontinuous Galerkin methods, in: D. Dissanayake (Ed.), Acoustic Waves, IntechOpen, Rijeka, 2010, http://dx.doi.org/10.5772/9714.

[3] V. Kostin, S. Solovyev, A. Bakulin, M. Dmitriev, Direct frequency-domain 3D acoustic solver with intermediate data compression benchmarked against time-domain modeling for full-waveform inversion applications, Geophysics 84 (4) (2019) T207–T219, http://dx.doi.org/10.1190/geo2018-0465.1.

[4] A. Bayliss, C. I.G.oldstein, E. Turkel, An iterative method for the helmholtz equation, J. Comput. Phys. (ISSN: 0021-9991) 49 (3) (1983) 443–457, http://dx.doi.org/10.1016/0021-9991(83)90139-0.

[5] E. Heikkola, Y. Kuznetsov, K. Lipnikov, Fictitious domain methods for the numerical solution of three-dimensional acoustic scattering problems, Journal of Computational Acoustics 07 (03) (1999) 161–183, http://dx.doi.org/10.1142/S0218396X99000126.

[6] M. Belonosov, M. Dmitriev, V. Kostin, D. Neklyudov, V. Tcheverda, An iterative solver for the 3D helmholtz equation, J. Comput. Phys. (ISSN: 0021-9991) 345 (2017) 330–344, http://dx.doi.org/10.1016/j.jcp.2017.05.026.

[7] G. Pan, A. Abubakar, Iterative solution of 3D acoustic wave equation with perfectly matched layer boundary condition and multigrid preconditioner, Geophysics 78 (5) (2013) T133–T140, http://dx.doi.org/10.1190/geo2012-0287.1.

[8] R.-E. Plessix, A helmholtz iterative solver for 3D seismic-imaging problems, Geophysics 72 (5) (2007) SM185–SM194, http://dx.doi.org/10.1190/1.2738849.

[9] R. Plessix, W. Mulder, Separation-of-variables as a preconditioner for an iterative helmholtz solver, Appl. Numer. Math. (ISSN: 0168-9274) 44 (3) (2003) 385–400, http://dx.doi.org/10.1016/S0168-9274(02)00165-4.

[10] O.G. Ernst, M.J. Gander, Why it is difficult to solve helmholtz problems with classical iterative methods, in: Numerical Analysis of Multiscale Problems, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-22061-6, 2012, pp. 325–363, http://dx.doi.org/10.1007/978-3-642-22061-6_10.

[11] I. Livshits, A. Brandt, Accuracy properties of the wave-ray multigrid algorithm for helmholtz equations, SIAM J. Sci. Comput. 28 (4) (2006) 1228–1251, http://dx.doi.org/10.1137/040620461.

[12] B. Engquist, L. Ying, Sweeping preconditioner for the helmholtz equation: Moving perfectly matched layers, Multiscale Model. Simul. 9 (2) (2011) 686–710, http://dx.doi.org/10.1137/100804644.

[13] J. Poulson, B. Engquist, S. Li, L. Ying, A parallel sweeping preconditioner for heterogeneous 3D helmholtz equations, SIAM J. Sci. Comput. 35 (3) (2013) C194–C212, http://dx.doi.org/10.1137/120871985.

[14] M. Gander, H. Zhang, A class of iterative solvers for the helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized schwarz methods, SIAM Rev. 61 (1) (2019) 3–76, http://dx.doi.org/10.1137/16M109781X.

[15] Y. Erlangga, C. Vuik, C. Oosterlee, On a class of preconditioners for solving the Helmholtz equation, Appl. Numer. Math. (ISSN: 0168-9274) 50 (3) (2004) 409–425, http://dx.doi.org/10.1016/j.apnum.2004.01.009.

[16] C.W. Oosterlee, C. Vuik, W.A. Mulder, R.-E. Plessix, Shifted-Laplacian preconditioners for heterogeneous helmholtz problems, in: B. Koren, K. Vuik (Eds.), Advanced Computational Methods in Science and Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-03344-5, 2010, pp. 21–46.

[17] C. Riyanti, A. Kononov, Y. Erlangga, C. Vuik, C. Oosterlee, R.-E. Plessix, W. Mulder, A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency helmholtz equation, J. Comput. Phys. (ISSN: 0021-9991) 224 (1) (2007) 431–448, http://dx.doi.org/10.1016/j.jcp.2007.03.033, Special Issue Dedicated to Professor Piet Wesseling on the occasion of his retirement from Delft University of Technology.

[18] C. Riyanti, Y.A. Erlangga, R.-E. Plessix, W.A. Mulder, C. Vuik, C. Oosterlee, A new iterative solver for the time-harmonic wave equation, Geophysics 71 (5) (2006) E57–E63, http://dx.doi.org/10.1190/1.2231109.

[19] M. Bollhofer, M. Grote, O. Schenk, Algebraic multilevel preconditioner for the helmholtz equation in heterogeneous media, SIAM J. Sci. Comput. 31 (5) (2009) 3781–3805, http://dx.doi.org/10.1137/080725702.

[20] A. Abubakar, T. Habashy, Three-dimensional visco-acoustic modeling using a renormalized integral equation iterative solver, J. Comput. Phys. (ISSN: 0021-9991) 249 (2013) 1–12, http://dx.doi.org/10.1016/j.jcp.2013.04.008.

[21] H. Kolsky, Lxxi. the propagation of stress pulses in viscoelastic solids, Phil. Mag. 1 (8) (1956) 693–710, http://dx.doi.org/10.1080/14786435608238144.

[22] W.I. Futterman, Dispersive body waves, Journal of Geophysical Research (1896-1977) 67 (13) (1962) 5279–5291, http://dx.doi.org/10.1029/JZ067i013p05279.

[23] K. Aki, P.G. Richards, Quantitative Seismology, 2nd Ed, University Science Books, 2002.

[24] B. Ursin, T. Toverud, Comparison of seismic dispersion and attenuation models, Studia Geophysica et Geodaetica (ISSN: 1573-1626) 46 (2) (2002) 293–320, http://dx.doi.org/10.1023/A:1019810305074.

[25] J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. (ISSN: 0021-9991) 114 (2) (1994) 185–200, http://dx.doi.org/10.1006/jcph.1994.1159.

[26] M. van Gijzen, Y. Erlangga, C. Vuik, Spectral analysis of the discrete helmholtz operator preconditioned with a shifted Laplacian, SIAM J. Sci. Comput. 29 (5) (2007) 1942–1958, http://dx.doi.org/10.1137/060661491.

[27] M.J. Gander, I.G. Graham, E.A. Spence, Applying gmres to the Helmholtz equation with shifted Laplacian preconditioning: What is the largest shift for which wavenumber-independent convergence is guaranteed? Numer. Math. 131 (3) (2015) 567–614, http://dx.doi.org/10.1007/s00211-015-0700-2, https://doi.org/10.1007/s00211-015-0700-2.

[28] S. Cools, W. Vanroose, On the optimality of shifted Laplacian in a class of polynomial preconditioners for the Helmholtz equation, in: Modern Solvers for Helmholtz Problems, Springer International Publishing, 2017, pp. 53–81, http://dx.doi.org/10.1007/978-3-319-28832-1_3.

[29] D. Neklyudov, M. Dmitriev, M. Belonosov, V. Tcheverda, Frequency-domain iterative solver for 3D acoustic wave equation with two-stage semi-analytical preconditioner, in: 76th EAGE Conference and Exhibition, 2014, http://dx.doi.org/10.3997/2214-4609.20140811.

[30] M. Frigo, S.G. Johnson, The design and implementation of fftw3, Proc. IEEE 93 (2) (2005) 216–231, http://dx.doi.org/10.1109/JPROC.2004.840301.

[31] N. Yavich, M. Malovichko, N. Khokhlov, M. Zhdanov, Preconditioning the time-harmonic acoustic wave equation based on a special contraction operator transformation, in: 2nd Conference on Geophysics for Mineral Exploration and Mining, 2018, http://dx.doi.org/10.3997/2214-4609.201802729.

[32] F. Aminzadeh, B. Jean, T. Kunz, 3-D Salt and Overthrust Models, SEG, 1997.

[33] M. Zhdanov, Geophysical Inverse Theory and Regularization Problems, first ed., Elsevier Science, 2002.

[34] N. Yavich, M.S. Zhdanov, Contraction pre-conditioner in finite-difference electromagnetic modelling, Geophys. J. Int. (ISSN: 0956-540X) 206 (3) (2016) 1718–1729, http://dx.doi.org/10.1093/gji/ggw237.

[35] E. Heikkola, T. Rossi, J. Toivanen, Fast direct solution of the helmholtz equation with a perfectly matched layer or an absorbing boundary condition, Internat. J. Numer. Methods Engrg. 57 (14) (2003) 2007–2025, http://dx.doi.org/10.1002/nme.752.

[36] V. Druskin, M. Zaslavsky, Fast finite-difference convolution for 3D problems in layered media, abs/1909.01299, 2019, ArXiv.

[37] E. Avdotin, N. Yavich, N. Khoohlov, M. Zhdanov, Increasing the effectiveness of 3D modeling visco-acoustic wave propagation with a solver based on contraction operator, (ISSN: 2214-4609) 2020 (1) (2020) 1–5, http://dx.doi.org/10.3997/2214-4609.202020050.

[38] I. Zacharov, R. Arslanov, M. Gunin, D. Stefonishin, A. Bykov, S. Pavlov, O. Panarin, A. Maliutin, S. Rykovanov, M. Fedorov, Zhores – petaflops supercomputer for data-driven modeling, machine learning and artificial intelligence installed in skolkovo institute of science and technology, Open Engineering 9 (1) (2019) 512–520, http://dx.doi.org/10.1515/eng-2019-0059.