

Available online at www.sciencedirect.com

Journal of Computational Physics 224 (2007) 431–448

JOURNAL OF
COMPUTATIONAL
PHYSICSwww.elsevier.com/locate/jcp

A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation

C.D. Riyanti ^{a,*}, A. Kononov ^b, Y.A. Erlangga ^c, C. Vuik ^a, C.W. Oosterlee ^a,
R.-E. Plessix ^d, W.A. Mulder ^d

^a Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

^b Computational Physics Group, Delft University of Technology, The Netherlands

^c Scientific Computing, Technical University, Berlin, Germany

^d Shell International E&P, P.O. Box 60, 2280 AB Rijswijk, The Netherlands

Received 30 September 2006; received in revised form 27 March 2007; accepted 29 March 2007

Available online 18 April 2007

Abstract

We investigate the parallel performance of an iterative solver for 3D heterogeneous Helmholtz problems related to applications in seismic wave propagation. For large 3D problems, the computation is no longer feasible on a single processor, and the memory requirements increase rapidly. Therefore, parallelization of the solver is needed. We employ a complex shifted-Laplace preconditioner combined with the Bi-CGSTAB iterative method and use a multigrid method to approximate the inverse of the resulting preconditioning operator. A 3D multigrid method with 2D semi-coarsening is employed. We show numerical results for large problems arising in geophysical applications.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Helmholtz equation; Krylov subspace method; Preconditioner; Multigrid method

1. Introduction

Important applications of the acoustic wave equation can be found in many fields, for instance, in geophysics, marine and aero acoustics. In geophysics, numerical methods can be employed in acoustic imaging techniques to gain insight in geological structures and physical properties deep within the Earth's subsurface [25,26]. The numerical solution of the wave equation has greatly helped both forward modeling and migration of seismic wave fields in complex Earth structures. It also serves as a starting point for solving the full inverse problem [29,30]. The wave equation can be solved either in the time- or in the frequency-domain. The numerical scheme can be based on finite difference or finite elements discretizations. In this paper, we focus on developing an efficient method for solving the wave equation in the frequency domain based on second-order finite-difference scheme.

* Corresponding author.

E-mail addresses: C.D.Riyanti@tudelft.nl, dwiriyanti@gmail.com (C.D. Riyanti).

Applying the Fourier transform [5] with respect to time to the acoustic wave equation, the frequency-domain wave equation (also called Helmholtz equation) is obtained. The Helmholtz equation can also be obtained if one considers the frequency-domain solution of the Telegraph equation, the Schrödinger equation in non-relativistic quantum mechanics [24], and the Klein–Gordon equation in relativistic quantum mechanics [14]. Once the solutions in the frequency domain are obtained, we can calculate the solution of the wave equation in the time domain with the help of the inverse Fourier transform [5]. To represent a solution of the wave equation for a given time interval accurately, however, a wide range of frequencies has to be resolved and the Helmholtz equation has to be solved for each frequency separately. For the purpose of the top 5 to 6 km of the Earth imaging in exploration seismology, the frequencies usually range from 10 to 60 Hz, with a small interval between the frequencies.

For 2D problems the computation can be performed by using, for example, direct methods combined with nested dissection reordering [13], and a LU decomposition. Only one LU decomposition is needed to calculate the solutions at multiple source locations. This leads to algorithms in the frequency domain that are more efficient than time domain methods. However, in 3D problems, the matrix size rapidly becomes too large and instead of LU decomposition, one has to resort to an iterative method. With the iterative method, we no longer have the main advantage in the frequency domain which is associated with the LU decomposition. Still, efficiency may be gained from using only a subset of the available frequencies to generate the time-domain solution [25].

For the 3D Helmholtz equation, many iterative methods suffer from slow convergence, especially if high frequencies have to be resolved. For this reason, the development of fast iterative methods for high-frequency Helmholtz problems remains a subject of active research. For recent work on this subject, see [21,9,17]. Recently, a novel 2D iterative solver has been developed in [11,12]. This method is robust and efficient for solving heterogeneous high-frequency Helmholtz problems. Numerical results for large 2D problems representative for the geophysical applications have been presented in [31]. In the present paper, we solve the 3D heterogeneous high-frequency Helmholtz equation efficiently, based on the 2D solver proposed in [12].

From the exploration-seismology point of view, the Earth is a heterogeneous semi-infinite medium. Since we choose to simulate seismic recording of finite duration of waves propagation at a finite velocity through the Earth, we can truncate our Earth model to the region of explorational interest, and apply absorbing conditions to the truncated edges of the Earth model [18]. A popular choice of boundary conditions, often used in geophysics, is a so-called absorption layer (or “sponge” layer) as the boundary conditions; (see, e.g., [7,16] and [22]). This layer is not a physical layer. It is used to gradually damp the outgoing waves by adding more dissipation to the wave equation. An iterative solver should be able to deal with these discretization features efficiently.

The outline of this paper is as follows. In Section 2, we present the finite difference discretization of the 3D Helmholtz equation, with the description of the second-order absorbing boundary conditions. We verify the accuracy of the discretization by comparing to the results obtained from an analytical solution for a constant wavenumber. In Section 3, a 3D multigrid method with semi-coarsening for the preconditioner solve is discussed. In Section 4, we introduce a parallel approach for 3D problems. A performance comparison of a single processor to that of multiple processors for a number of problems is presented. Finally, numerical results for a large 3D model arising in a geophysical application are reported.

2. Finite difference discretization

We start with the description of the 3D Helmholtz equation in a domain $\Omega \subset \mathbb{R}^3$ of the following form:

$$-\nabla^2 u(\mathbf{x}, \omega) - k(\mathbf{x})^2 u(\mathbf{x}, \omega) = g(\mathbf{x}, \omega), \quad \mathbf{x} \in \Omega, \quad (1)$$

where u represents the pressure field, $k(\mathbf{x}) = \omega/c(\mathbf{x})$ is the wavenumber, $c(\mathbf{x})$ is the acoustic-wave velocity which varies with position and $\omega = 2\pi f$ denotes angular frequency. The source term is denoted by g and $\nabla^2 = (\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2})$ is the Laplace operator.

In order to mimic an infinite space and to avoid reflections from the boundary, we add absorption layers (Ω^e) to the physical domain, Ω , (see Fig. 1). In Ω^e , we consider the damped Helmholtz equation in the following form [33]:

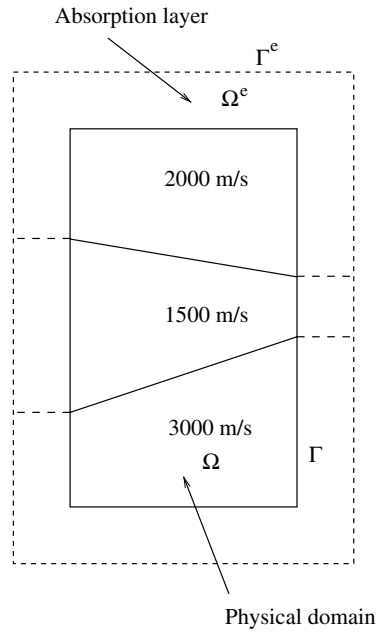


Fig. 1. A cross-section of the 3D domain with absorption layer in the case of a regular heterogeneous medium.

$$-\nabla^2 u(\mathbf{x}, \omega) - (1 - \alpha)k(\mathbf{x})^2 u(\mathbf{x}, \omega) = g(\mathbf{x}, \omega), \quad \mathbf{x} \in \Omega^e, \tag{2}$$

where

$$\alpha = 0.25 \frac{\|\mathbf{x} - \mathbf{x}_d\|^2}{\|\mathbf{x}_e - \mathbf{x}_d\|^2}, \quad \mathbf{x} \in \Omega^e, \tag{3}$$

with \mathbf{x}_d is a point of the boundary Γ and \mathbf{x}_e is a point in the absorption layer Γ^e . The imaginary unit is denoted by i . On the boundary Γ_e , this equation is supplemented by the following second-order absorbing boundary conditions [10,3].

On each face the boundary condition is of the form:

$$\frac{\partial u}{\partial x_k} - i\omega u - \frac{i}{2\omega} \sum_{1 \leq j \neq k \leq 3} \frac{\partial^2 u}{\partial x_j^2} = 0 \quad \text{on } \Gamma^e. \tag{4}$$

On each edge, we have the condition

$$-\frac{3}{2}\omega^2 u - i\omega \left(\frac{\partial u}{\partial x_m} \frac{\partial u}{\partial x_j} \right) - \frac{1}{2} \frac{\partial^2 u}{\partial x_k^2} = 0 \quad \text{on } \Gamma^e. \tag{5}$$

Furthermore, on each corner, the conditions are given by:

$$-2i\omega u + \sum_{j=1}^3 \frac{\partial u}{\partial x_j} = 0 \quad \text{on } \Gamma^e. \tag{6}$$

Eqs. (1)–(6) are discretized by second-order scheme. The discretization results in the linear system:

$$A\phi = b, \tag{7}$$

where ϕ and b are the pressure field and source, respectively.

For second order finite-differences, the well-known 7-point discretization stencil A_h corresponding to matrix A is:

$$A_h \triangleq -\frac{1}{h^2} \left(\begin{bmatrix} 0 & & \\ 0 & -1 & 0 \\ & 0 & \end{bmatrix}_{i_3-1} \begin{bmatrix} & -1 & \\ -1 & 6 - k^2 h^2 & -1 \\ & -1 & \end{bmatrix}_{i_3} \begin{bmatrix} 0 & & \\ 0 & -1 & 0 \\ & 0 & \end{bmatrix}_{i_3+1} \right). \quad (8)$$

Now, A is a large but sparse matrix with complex values, because of the absorbing boundary conditions. The vectors ϕ and b are discrete versions of u and g in (1), respectively. We denote by n_1 , n_2 , and n_3 the number of discretization points in the x_1 , x_2 and x_3 -coordinates, respectively, and assume that $n_1 = n_2 = n_3 = n$, so A is of size $n^3 \times n^3$. With the natural ordering of the grid points, A has a band width of size n^2 , but with only $O(n^3)$ non-zero elements. In order to obtain an accurate numerical solution, it is required to choose 10–12 points per wavelength λ_f (for the second-order discretization), where $\lambda_f = \frac{c}{f}$ with f is the frequency [28]. In a heterogeneous medium, the smallest velocity is usually selected as the representative wavelength. The domain of the absorption layer is normally defined between $2\lambda_f$ and $3\lambda_f$. The wavenumber k can be large, which implies that the wave Eq. (1) has both positive and negative eigenvalues, and therefore the matrix A is indefinite. In order to avoid a reduction of accuracy for a second order scheme due to so-called *pollution effect* [1–3] $k^2 h^3$ should be set constant. As for an iterative solution method, keeping kh constant is more severe; we stay with $kh = 0.625$ as discussed in [12] for 2D case.

The system of linear Eq. (7) becomes very large for high frequencies. In three-dimensional problems, one can have millions of unknowns and the complexity of a direct solver is $O(n^7)$, while the storage required for L and U in an LU-decomposition is of $O(n^5)$. Therefore, we focus on an iterative solution technique.

Common iterative methods for indefinite systems are based on Krylov subspace methods [32,6]. These methods are generalizations of the conjugate-gradient (CG) methods. The Bi-conjugated gradient stabilized (Bi-CGSTAB) algorithm [36] is one of Krylov subspace algorithms. In this algorithm, ϕ_i is iteratively updated from an initial guess ϕ_0 until the residual error $\frac{\|b - A\phi_i\|}{\|\phi_i\|}$ is small enough. Without a preconditioner, this method converges slowly or not at all for Helmholtz problems [28]. By preconditioning, we solve an equivalent linear system as:

$$AM^{-1}\tilde{\phi} = b, \quad \tilde{\phi} = M\phi. \quad (9)$$

In [12], an operator of the preconditioner of the following form has been proposed for 2D applications:

$$\mathcal{M}_p = -(\beta_1 - i\beta_2)k^2 - \nabla^2, \quad (10)$$

where the parameters β_1 and β_2 should be chosen such that \mathcal{M}_p resembles A , but is easy to solve at the same time. k is again the wave number and ∇^2 the Laplacian. The operator \mathcal{M}_p is called a complex-shifted Laplacian and corresponds to a damped wave equation. Boundary conditions are set identical to those for the original wave equation. The preconditioning matrix M is obtained from finite-difference discretization of \mathcal{M}_p . In this paper, the preconditioning matrix M is discretized by the 7-point finite difference stencil.

2.1. Validation of the numerical results

In order to validate the accuracy of the discretization, we consider a constant wavenumber model with $k = \omega/c$, $\omega = 2\pi f$, $f = 40$ Hz and $c = 2000$ m/s, respectively. The size of the model is $D = [0, 304] \times [0, 128] \times [0, 304]$ m³. We excite a source located at (152, 64, 4) m. The model is discretized on a $76 \times 32 \times 76$ grid with a grid spacing of 4 m. The wavelength is 50 m; the number of grid points per wavelength is about 12. For the constant wavenumber model, an analytical solution can be found in the form of a Green's function [8]. This analytical solution can be used to evaluate the accuracy of the numerical results using the second order discretization and the absorbing boundary conditions obtained by the iterative method.

In the numerical experiment, we compare two different approaches for the boundary treatment, and compare their accuracy. First, we consider the wave equation with the second-order boundary condition employed directly at the boundaries [12]. We apply the second-order finite-difference scheme to obtain the matrices. Fig. 2a presents the real part of the numerical solution along the $x_2 = 64$ m line.

In the second test experiment, extra damping layers are added along the physical domain. An absorbing layer of 25 points was added to each side. This means that the width of the extra damping layer is approximately two times the wavelength. The total number of grid points is $126 \times 82 \times 126$ in this case. Fig. 2b presents

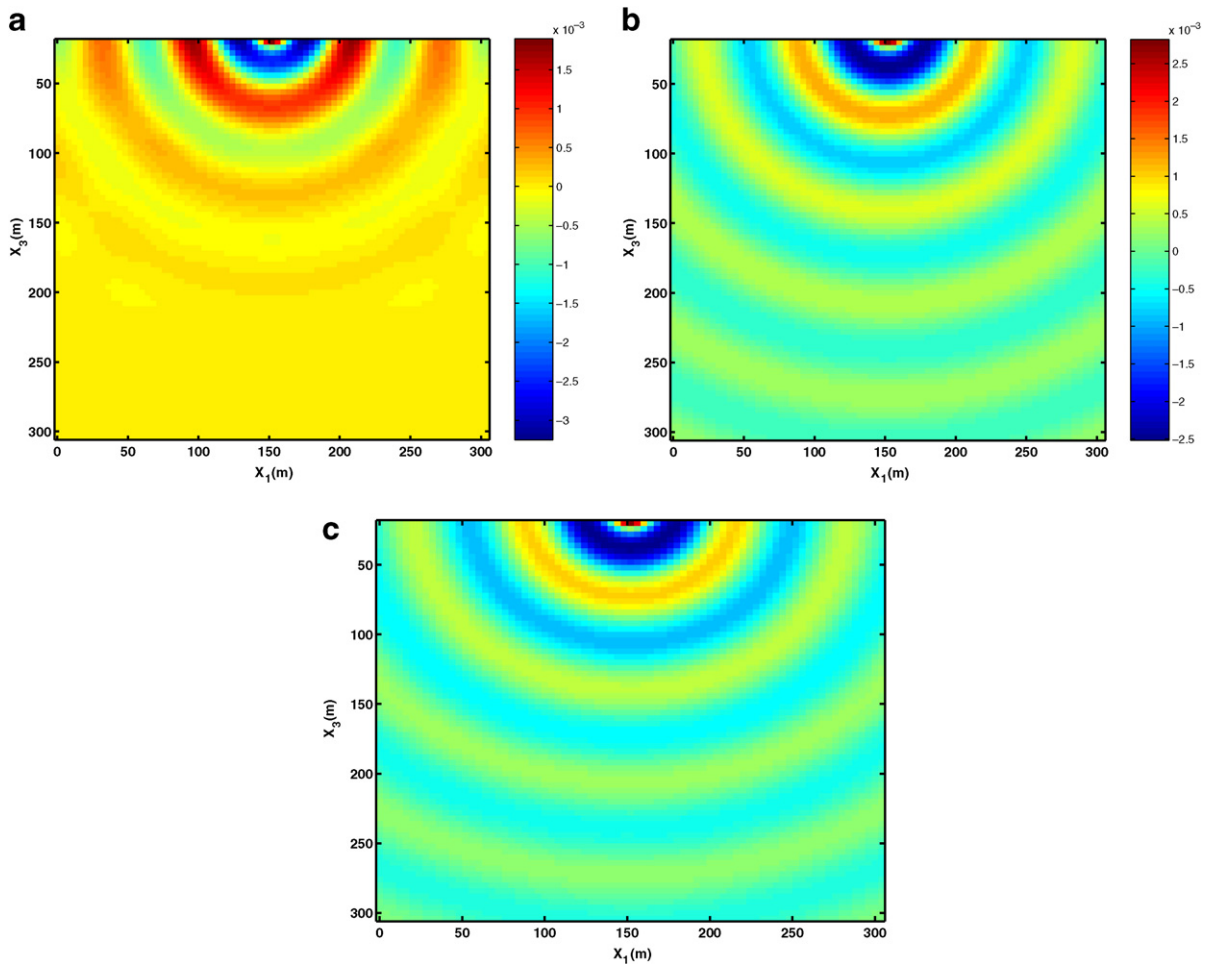


Fig. 2. The real part of the numerical results obtained by employing second-order discretization: (a) without absorbing boundary condition; (b) with absorbing boundary condition; (c) analytical solution along the $x_2 = 64$ m line for a velocity $c = 2000$ m/s.

the real part of the numerical solution with the absorbing boundary conditions; the analytical solution along the $x_2 = 64$ m line is depicted in Fig. 2c. Notice that in this paper, the real part of the numerical results is plotted in 2D, although we compute the numerical solution in 3D. These 2D depictions of the results are representative of the results for the entire 3D domain. From Figs. 2a–c, we find that the solution without the damping layer is less accurate than the solution with the damping layer. Without a damping layer, unphysical reflections at the boundaries are present, while with the damping layer, reflections from the boundaries are suppressed.

In Fig. 3, we compare the numerical solution with absorbing layer to the analytical solution at a receiver position of $(x_1, x_2) = (120, 64)$ m. From this figure, it can be observed that the numerical solution with the absorbing boundary condition is in very good agreement with the analytical solution. In Table 1, we present the comparison between the numerical solution and the analytical solution for different positions. From this table, it is shown that the difference between the analytical and numerical results for different positions is less than $2.0e^{-2}$.

3. Multigrid method for the preconditioner

A preconditioned Krylov subspace method has been employed for solving the 3D Helmholtz Eq. (9). An inversion of the preconditioner in the algorithm by a direct solution method is too expensive. An alternative

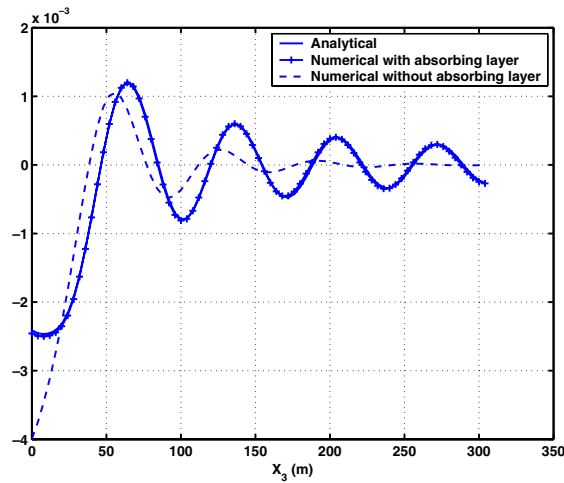


Fig. 3. Comparison between numerical solution and analytical solution for $c = 2000$ m/s at $x_1 = 120$ m.

Table 1
Comparison between the numerical solution and the analytical solution for different positions

x_1 -coordinate	x_3 -coordinate	$\frac{ \text{exact-numeric} }{ \text{exact} }$
0	0	7.9×10^{-3}
16	32	1.7×10^{-3}
60	75	1.5×10^{-2}
128	60	3.0×10^{-3}
150	150	2.6×10^{-3}
184	200	5.2×10^{-3}
250	304	2.5×10^{-3}
256	200	5.9×10^{-3}
304	120	6.7×10^{-3}
304	304	1.4×10^{-3}

approach is to approximately solve the inverse of the preconditioner by using, for example, a multigrid method [4,35]. The multigrid method performs well for carefully chosen values of β_1 and β_2 in (10), which is interesting, as multigrid does not perform well for the original Helmholtz problem [35]. Local Fourier analysis tools helped in the determination of appropriate values of β_1 and β_2 in 2D. The multigrid method and the parameters as proposed in [12] are employed in a preconditioner solver that is now generalized to 3D.

The iterative solver proposed in this paper for the solution of the original (undamped) Helmholtz equation is a combination of one inner iteration with the multigrid method for solving the (damped) preconditioner system (10) and outer iterations with BiCGSTAB.

For 2D problems, multigrid with standard two-dimensional coarsening was used. The multigrid algorithm was based on the F-cycle with one pre- and one post-smoothing iteration. The fully-parallel point-wise ω -Jacobi smoother with relaxation factor $\omega = 0.5$ was chosen. Coarse-to-fine grid interpolation was based on the matrix-dependent interpolation operator in [38]. This type of interpolation is especially suited for matrices with varying coefficients. Restriction from fine to coarse grids was based on full weighting. The coarse grid matrix was built by the Galerkin coarse grid discretization. In summary, the multigrid components chosen were:

- (i) Restriction I_h^H was 2D full weighting,
- (ii) Prolongation, I_H^h was 2D matrix-dependent interpolation,
- (iii) Coarse grid matrix was the Galerkin coarse grid discretization: $M_H = I_h^H M_h I_H^h$, etc.

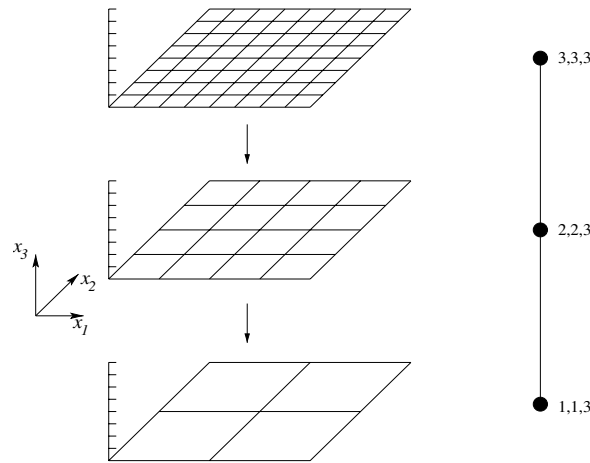


Fig. 4. Semi-coarsening of three grid levels: standard coarsening in two directions (x_1 and x_2), the third (x_3) direction is kept uncoarsened.

With these multigrid components the inclusion of the absorbing boundary conditions as explained in (3)–(6), does not lead to any multigrid convergence difficulties. With the matrix-dependent prolongation and with the Galerkin coarse grid discretization, we can handle this with the typical multigrid efficiency. A rigorous Fourier analysis (RFA) for these multigrid components in 2D was discussed in [12].

For 3D problems, we employ in this paper a multigrid method with a 2D semi-coarsening strategy combined with line-wise ω -Jacobi smoothing in the third direction. This means that the coarsening is performed in two directions simultaneously, whereas we keep the fine grid spacing in the third direction throughout the algorithm on all multigrid levels. This is schematically illustrated in Fig. 4. A particular robust variant of multigrid with 2D semi-coarsening has been introduced in [37] with corresponding analysis. This approach is based on 2D problems [12] where the 2D standard multigrid is employed. We adopt a variant that is easily set up on the basis of a 2D multigrid method with standard coarsening, as all the multigrid components from 2D apply. The 2D point-wise Jacobi smoother is generalized here to a line-wise Jacobi smoother, updating all points in the uncoarsened, x_3 -direction simultaneously. The robustness and efficiency of the present 3D multigrid method should be at least similar to a 3D multigrid method based on standard 3D grid coarsening and point-wise smoothing [37].

The x_3 -line Jacobi relaxation with underrelaxation factor ω_{jac} is defined by the following iteration:

$$(M_{x_3} + D)\tilde{\phi}^{j+1} + (M_{x_1} + M_{x_2})\tilde{\phi}^j = b, \tag{11}$$

$$\tilde{\phi}^{j+1} = \omega_{jac}\tilde{\phi}^{j+1} + (1 - \omega_{jac})\tilde{\phi}^j, \tag{12}$$

or

$$\begin{aligned} \frac{1}{\omega_{jac}}(M_{x_3} + D)\tilde{\phi}^{j+1} &= b - (M_{x_1} + M_{x_2})\tilde{\phi}^j + \frac{1 - \omega_{jac}}{\omega_{jac}}(M_{x_3} + D)\tilde{\phi}^j, \\ &= b - M\tilde{\phi}^j + \frac{1}{\omega_{jac}}(M_{x_3} + D)\tilde{\phi}^j, \end{aligned} \tag{13}$$

with in the case of $O(h^2)$ -discretization.

$$(M_{x_1}\tilde{\phi})_{i_1,i_2,i_3} := -\frac{1}{h^2}(\tilde{\phi}_{i_1+1,i_2,i_3} + \tilde{\phi}_{i_1-1,i_2,i_3}), \tag{14}$$

$$(M_{x_2}\tilde{\phi})_{i_1,i_2,i_3} := -\frac{1}{h^2}(\tilde{\phi}_{i_1,i_2+1,i_3} + \tilde{\phi}_{i_1,i_2-1,i_3}), \tag{15}$$

$$(M_{x_3}\tilde{\phi})_{i_1,i_2,i_3} := -\frac{1}{h^2}(\tilde{\phi}_{i_1,i_2,i_3+1} + \tilde{\phi}_{i_1,i_2,i_3-1}), \tag{16}$$

$$(D\tilde{\phi})_{i_1,i_2,i_3} := \frac{1}{h^2}(6 - (\beta_1 - \beta_2\hat{j})k^2h^2)\tilde{\phi}_{i_1,i_2,i_3}. \tag{17}$$

3.1. Transfer operators

In the 3D multigrid algorithm based on 2D semi-coarsening, we need 2D transfer operators. The interpolation operator can be based on bilinear interpolation or a matrix-dependent interpolation. We choose for the matrix-dependent interpolation as in [37,38], which is especially useful for problems with strong heterogeneities. For the complex Helmholtz preconditioner, we modify the weights in the prolongation operator. The weights are then based on the *modulus* of the complex-valued functions, as these have been shown to be the natural extension to the complex case in [12]. The restriction operator can be the same as in 2D because coarsening is not employed in the third direction. In the definition of the Galerkin coarse grid matrices, the restriction operator is usually chosen to be the transpose conjugate of the interpolation operator. In numerical tests in [12], however, it was shown that the full-weighting restriction operator in combination with the matrix-dependent prolongation operator, resulted in a robust overall solution method for several Helmholtz problems with highly irregular heterogeneities and strong contrasts. So, $I_h^H \neq (I_H^h)^*$. The 2D full weighting restriction operator in stencil notation reads:

$$I_h^H \triangleq \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^H \tag{18}$$

The 3D interpolation weights are now explained. Assuming we have the following 27-point stencil matrix (see Fig. 5):

$$\begin{aligned} (M\tilde{\phi})_{i_1,i_2,i_3} = & \sum_{iz=-1,0,1} [m(iz)_{i_1,i_2,i_3}^1 \tilde{\phi}_{i_1-1,i_2-1,i_3+iz} + m(iz)_{i_1,i_2,i_3}^2 \tilde{\phi}_{i_1,i_2-1,i_3+iz} + m(iz)_{i_1,i_2,i_3}^3 \tilde{\phi}_{i_1+1,i_2-1,i_3+iz} \\ & + m(iz)_{i_1,i_2,i_3}^4 \tilde{\phi}_{i_1-1,i_2,i_3+iz} + m(iz)_{i_1,i_2,i_3}^5 \tilde{\phi}_{i_1,i_2,i_3+iz} + m(iz)_{i_1,i_2,i_3}^6 \tilde{\phi}_{i_1+1,i_2,i_3+iz} \\ & + m(iz)_{i_1,i_2,i_3}^7 \tilde{\phi}_{i_1-1,i_2+1,i_3+iz} + m(iz)_{i_1,i_2,i_3}^8 \tilde{\phi}_{i_1,i_2+1,i_3+iz} + m(iz)_{i_1,i_2,i_3}^9 \tilde{\phi}_{i_1+1,i_2+1,i_3+iz}]. \end{aligned} \tag{19}$$

Assuming that coarsening is only done in the (x_1, x_2) -directions, a lumped 9-point stencil matrix \tilde{M} in an (x_1, x_2) -plane is defined as:

$$\begin{aligned} (\tilde{M}\tilde{\phi})_{i_1,i_2,i_3} = & \tilde{m}_{i_1,i_2,i_3}^1 \tilde{\phi}_{i_1-1,i_2-1,i_3} + \tilde{m}_{i_1,i_2,i_3}^2 \tilde{\phi}_{i_1,i_2-1,i_3} + \tilde{m}_{i_1,i_2,i_3}^3 \tilde{\phi}_{i_1+1,i_2-1,i_3} + \tilde{m}_{i_1,i_2,i_3}^4 \tilde{\phi}_{i_1-1,i_2,i_3} + \tilde{m}_{i_1,i_2,i_3}^5 \tilde{\phi}_{i_1,i_2,i_3} \\ & + \tilde{m}_{i_1,i_2,i_3}^6 \tilde{\phi}_{i_1+1,i_2,i_3} + \tilde{m}_{i_1,i_2,i_3}^7 \tilde{\phi}_{i_1-1,i_2+1,i_3} + \tilde{m}_{i_1,i_2,i_3}^8 \tilde{\phi}_{i_1,i_2+1,i_3} + \tilde{m}_{i_1,i_2,i_3}^9 \tilde{\phi}_{i_1+1,i_2+1,i_3}, \end{aligned} \tag{20}$$

with

$$\tilde{m}_{i_1,i_2,i_3}^p = m(-1)_{i_1,i_2,i_3}^p + m(0)_{i_1,i_2,i_3}^p + m(1)_{i_1,i_2,i_3}^p, \quad p = 1, 2, \dots, 9. \tag{21}$$

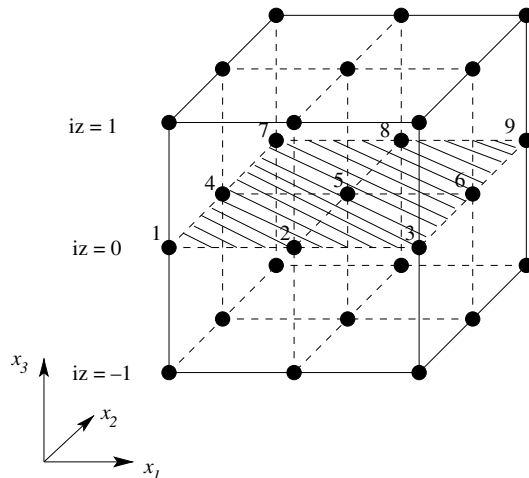


Fig. 5. The 27-point stencil with numbering.

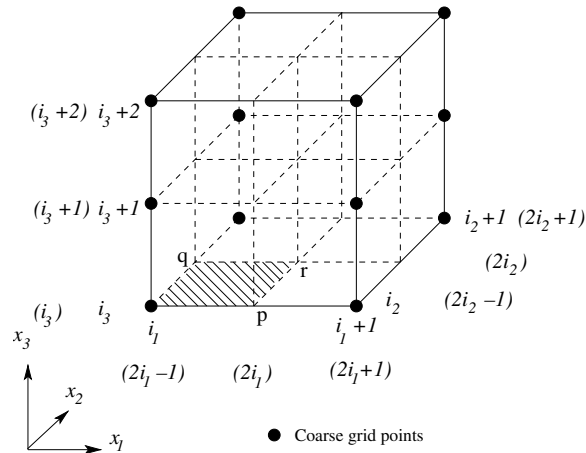


Fig. 6. Coarse and fine grid cells for (x_1, x_2) -semi-coarsening.

Based on the lumped 9-point stencil, the coarse-to-fine grid operator can be determined as:

$$\begin{aligned} \bar{\phi}_{2i_1-1, 2i_2-1, i_3} &= \tilde{\phi}_{i_1, i_2, i_3}, \\ \bar{\phi}_{2i_1, 2i_2-1, i_3} &= w1_{2i_1, 2i_2-1, i_3} \tilde{\phi}_{i_1, i_2, i_3} + w2_{2i_1, 2i_2-1, i_3} \tilde{\phi}_{i_1+1, i_2, i_3}, \\ \bar{\phi}_{2i_1-1, 2i_2, i_3} &= w1_{2i_1-1, 2i_2, i_3} \tilde{\phi}_{i_1, i_2, i_3} + w3_{2i_1-1, 2i_2, i_3} \tilde{\phi}_{i_1, i_2+1, i_3}, \\ \bar{\phi}_{2i_1, 2i_2, i_3} &= w1_{2i_1, 2i_2, i_3} \tilde{\phi}_{i_1, i_2, i_3} + w2_{2i_1, 2i_2, i_3} \tilde{\phi}_{i_1+1, i_2, i_3} + w3_{2i_1, 2i_2, i_3} \tilde{\phi}_{i_1, i_2+1, i_3} + w4_{2i_1, 2i_2, i_3} \tilde{\phi}_{i_1+1, i_2+1, i_3}, \end{aligned}$$

with the weights $w1, \dots, w4$ determined from the 2D interpolation weights [38], see also [37].

Fig. 6 shows coarse and fine grid cells for (x_1, x_2) -coarsening.

In Appendix A, we explain the matrix-dependent interpolation weights in more detail (for bilinear interpolation in multigrid we refer to, for example, [35]).

Using a single processor machine, we can only compute discrete problems with grid sizes up to approximately $100 \times 100 \times 100$. This is still far from realistic in geophysical applications. They require resolutions between 150 and 1000 grid points in each coordinate direction. Because of this, we parallelized the solver as explained in the previous section.

4. Parallelization of the solver

For realistic problems, we need to scale the size of the model up to $500 \times 500 \times 500$ unknowns or more. A single processor cannot handle these problems, so we focus on an efficient parallelization of the 3D algorithm. In [19], the parallel performance of the 2D version of the iterative method was reported. Here, we generalize the 2D parallelization strategy to 3D. The preconditioned iterative method can be well parallelized. The code is supplemented by MPI (Message-Passing Interface)-routines [27,34]. Let p_1, p_2, p_3 be the number of processors in the x_1 -, x_2 - and x_3 -directions, with $p_1 \times p_2 \times p_3 = p$, and let n_1, n_2 and n_3 be the number of grid points in x_1 -, x_2 - and x_3 -coordinates, respectively. For the present 3D solver based on semi-coarsening we choose $p_3 = 1$. So, parallelization will not be done in the x_3 -direction, only along the x_1 - and x_2 -directions. The line-wise relaxation takes place on one processor. The grid points in the (x_1, x_2) -plane are partitioned in a fashion as depicted in Fig. 7.

There are two operations which dominate the parallel computation. The first one is the computation of matrix-vector multiplications within the Bi-CGSTAB method. This operation is parallelized by using the standard MPI-subroutines [34] (for example MPI_Send_Init, MPI_Recv_Init, MPI_Startall, MPI_Waitall), and it requires communication between processors corresponding to neighboring subregions. The second one is the approximate inversion of the preconditioner using the multigrid method. This operation consists of the prolongation and restriction operations, line-wise Jacobi pre-and post-smoothing operations. These operations

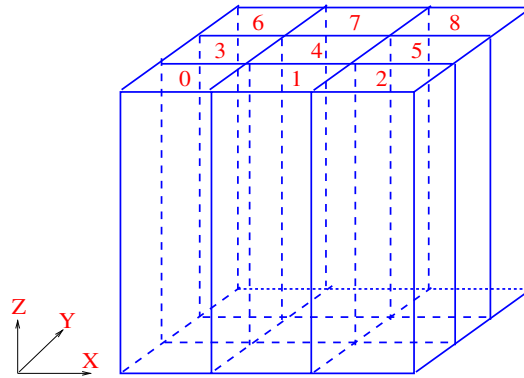


Fig. 7. Parallel configuration for $p = 9$ ($p_1 = 3$, $p_2 = 3$ and $p_3 = 1$).

are structurally similar to the matrix-vector multiplication and possess similar communication patterns. All these operations are performed fully in parallel, except for the coarsest multigrid levels, in which the program execution switched from a parallel to a sequential mode. This is done to optimize the performance: communication times on very coarse grids would dominate calculation times with only a few grid points and many processors. The program flow from parallel to sequential is governed by a special variable, which stores the current multigrid level and controls the behavior of the parallel program.

Further parallel operations, like the dot product and similar operations, effectively need global communication between all MPI-processes. This has been implemented using MPI_Allgather subroutine. Vector updates do not require any communication between neighboring processors. The solution output is also performed in a parallel fashion.

5. Numerical performance for larger models

5.1. Numerical results using a single processor

In [12], the parameters β_1 and β_2 in the preconditioner (10) have been evaluated, without an absorption boundary layer (i.e., with $\alpha = 0$). In this paper, we perform a parameter study for the 3D preconditioner, including the extra damping layer along the physical domain. We first consider a constant wavenumber model, with $k = 20$ and a unit-cube computational domain, $D = [0, 1]^3$. The source is located at $(0.5, 0.5, 0)$. The number of grid points is about 12 per wavelength, so that we have $kh = 0.625$, where h is the mesh width. We add 20 points to each side for the absorbing boundary. A 32-bit server (AMD Athlon(TM) XP 2600+ with 1.03GB memory) was used. The solution process is started with a zero-initial guess, and the iteration is terminated when the residual is less than 10^{-6} . The convergence results of the preconditioned Bi-CGSTAB method, in terms of numbers of iterations and CPU times, for varying values of β_1 and β_2 are summarized in Table 2.

In Table 2, the value of β_1 is taken between -1 and 1 , while β_2 ranges from 0 to 1 . From the table, we see that the values of $(\beta_1, \beta_2) = (0.0, 1.0)$ and $(\beta_1, \beta_2) = (-1.0, 0.0)$ ([20]) are not optimal for the preconditioner which is comparable to 2D. From the results in Table 2, we observe that, as in 2D, the values $\beta_1 = 1.0$ and $\beta_2 = 0.5$ are suitable parameter choices. We will also employ these parameters, $(\beta_1, \beta_2) = (1.0, 0.5)$ in the preconditioner, when solving the 3D wave equation in heterogeneous media.

Next, we also present experiments that evaluate the influence of a different semi-coarsening strategy on the convergence of the preconditioned Bi-CGSTAB method for the six-layer model (Fig. 8a) and the 3D wedge model (Fig. 8b).

In Table 3 (second and fourth columns) (x_1, x_3) -semi-coarsening in which a direction without variation in k (i.e. x_2) is kept uncoarsened is considered. Compared with the convergence results obtained using (x_1, x_2) -semi-coarsening (first and third columns in Table 3), we observe that for this test the results of the semi-coarsening strategies is geometry dependent. The (x_1, x_3) -semi-coarsening is a proper choice for both cases.

Furthermore, we investigate the behavior of the frequency-domain iterative solver by considering the more realistic models. Here we take the domain of $D = 304 \times 304 \times 304 \text{ m}^3$. Three different models are considered

Table 2

Performance of the solution method for different values of β_1, β_2 in the preconditioner M , number of iterations (Nit) and CPU time (in s). The total grid points for all cases above is 32^3

β_1	β_2	Nit	Times (s)
1.0	0.25	56	204.41
1.0	0.5	10	48.03
1.0	0.75	12	54.81
1.0	1.0	16	98.24
0.0	1.0	24	137.24
-1.0	0.0	68	352.35

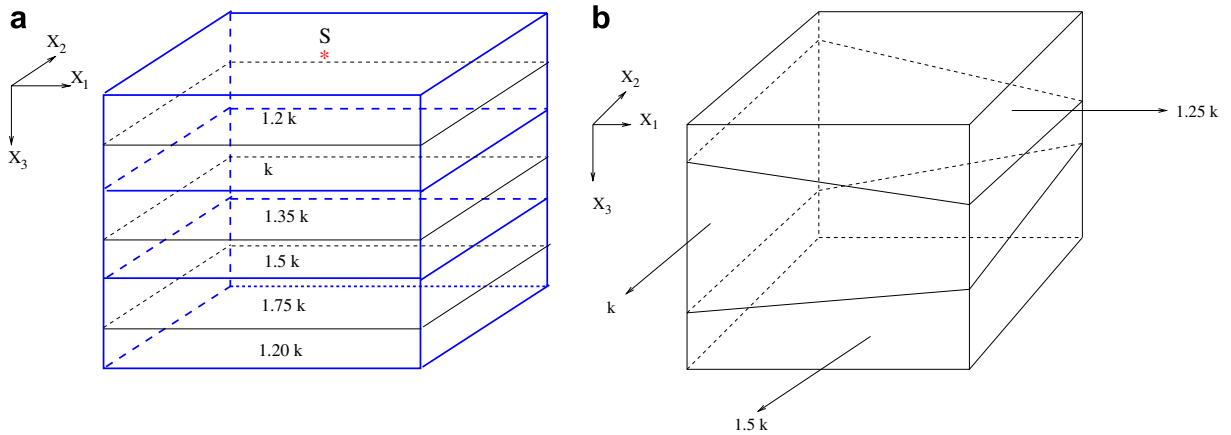


Fig. 8. The geometry of a 6-layer model (a) and 3D wedge model (b).

Table 3

Solver performance for different semi-coarsening strategies for the 6 layers and the 3D wedge models in terms of number of iterations and CPU time on one processor in s (in brackets)

Model:	6 layers		3D wedge	
	(x_1, x_2)	(x_1, x_3)	(x_1, x_2)	(x_1, x_3)
k :/Grid:				
10/16 ³	19 (7.03)	19 (7.03)	20 (7.77)	16 (6.11)
20/32 ³	30 (29.23)	26 (25.09)	27 (26.20)	23 (23.31)
30/48 ³	38 (77.89)	36 (72.45)	38 (78.87)	31 (65.05)
40/64 ³	47 (160.36)	43 (145.26)	49 (165.28)	39 (135.78)
50/80 ³	62 (371.48)	56 (332.22)	58 (354.66)	52 (319.72)

i.e., a constant velocity model ($c = 2000$ m/s), three layers model with c varying from 2000 m/s in the first layer to 5000 m/s in the third layer, and a 3D wedge model with three layers with velocities varying from 1500 m/s in the first layer to 3000 m/s in the third layer (as depicted in Fig. 8b). In Fig. 9, we have plotted the number of iterations versus frequency. In this experiment, the number of grid points per wavelength has been kept constant, so the number of grid points n in each coordinate is proportional to the frequency used. From this figure, it is shown that the number of iterations varies linearly with the frequency.

5.2. Parallel performance for different models

In order to illustrate the performance of the parallel approach, we again consider three different models, i.e., the constant model ($c = 2000$ m/s), only 3 layers model and the 3D wedge model (as depicted in Fig. 8a–b). The computations are performed on an SGI Altix 3700 system (www.sara.nl). We use 25 proces-

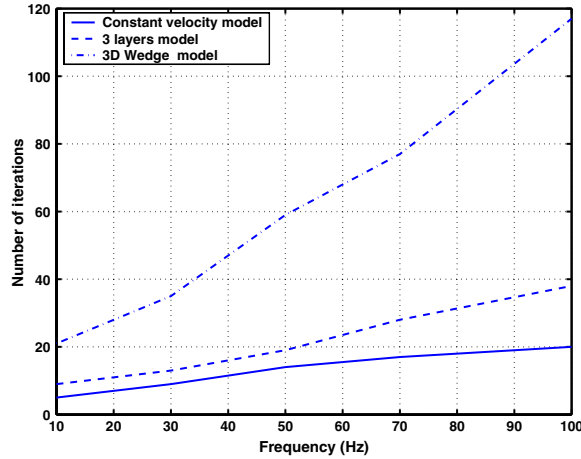


Fig. 9. The number of iterations of the iterative solver versus frequency.

sors. An effective way to assess the performance of a parallel program is to measure its execution time T as a function of the number of processors N_p . Additionally the so-called *parallel efficiency* can be evaluated that indicates how well the parallel performance scales. It is defined as follows ([34]):

$$E(N_p) = \frac{T(1)}{N_p T(N_p)}, \tag{22}$$

where $T(1)$ indicates the execution time for a single processor and $T(N_p)$ denotes the execution time for N_p processors. Ideally, if the program scales well then its efficiency is ≈ 1 .

Fig. 10a shows the speedup factor for three different models. The speedup factor is the wall clock time of N_p processors divided by the wall clock time for single processor in the same model. Fig. 10b shows the efficiency of the parallel approach for three different methods.

5.3. More complex model: a 3D salt dome model

In this section, we evaluate the multigrid preconditioner and Bi-CGSTAB solver for a 3D realistic model that can be found in the geophysical applications, a salt dome model typical for geological structures in the

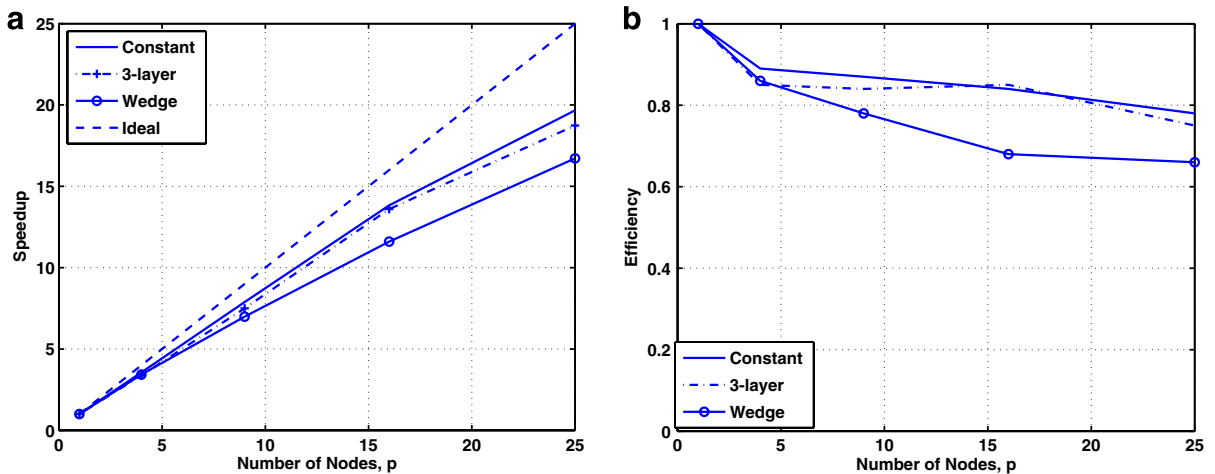


Fig. 10. The parallel efficiency for three different models.

North sea. The changes in velocity throughout the domain are given in Fig. 11. The velocity (c) varies from 1700 m/s to 4900 m/s. The size of the model is $8920 \times 4440 \times 5100 \text{ m}^3$. The source is an explosive source located in the center at point of (4460, 2220, 20) m. We consider the frequencies of 2.5 Hz, 5 Hz and 10 Hz.

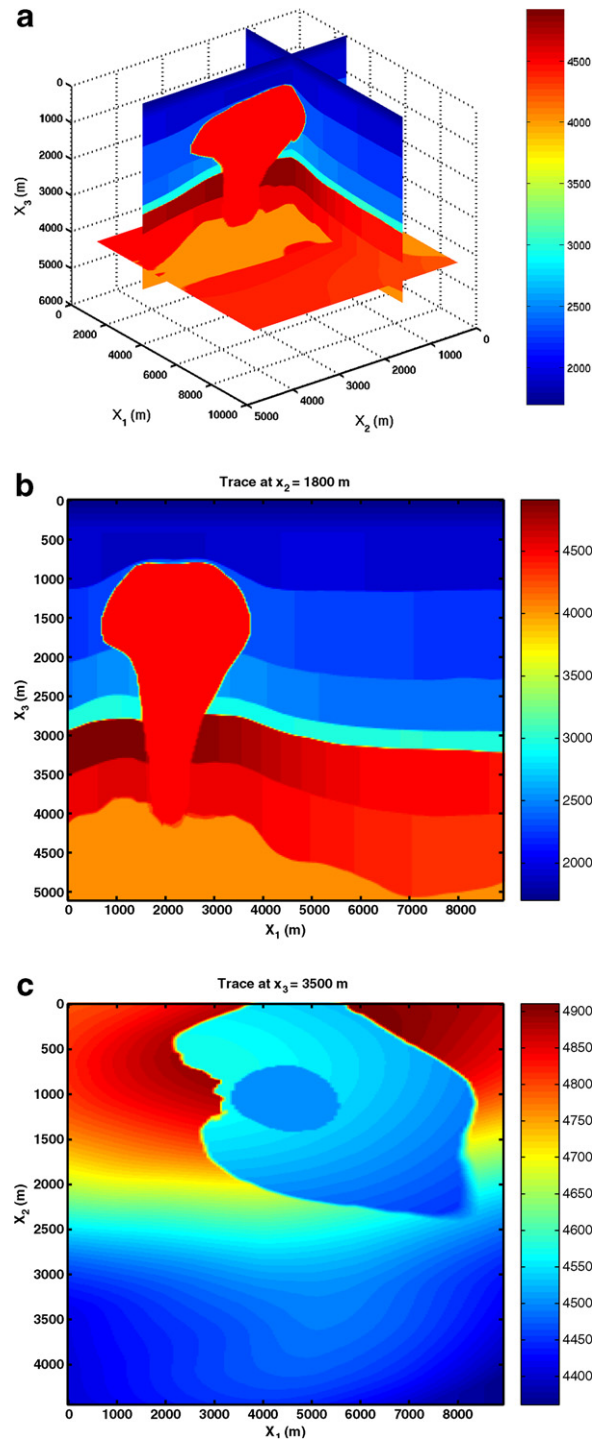


Fig. 11. (a) The 3D salt dome velocity model. The velocity varies from 1700 m/s to 4900 m/s. Cross-section of the model at $x_2 = 1800$ m (b) and $x_3 = 3500$ m (c), respectively.

The minimum wavelength, λ_{\min} , can be calculated as the minimum velocity divided by the frequency. In this case λ_{\min} is about 680 m, 340 m and 170 m for 2.5 Hz, 5 Hz, and 10 Hz, respectively.

We also investigate two different semi-coarsening strategies, i.e. (x_1, x_2) -semi-coarsening and (x_1, x_3) -semi-coarsening for the frequency of 2.5 Hz, 5 Hz and 10 Hz.

For $f = 2.5$ Hz, the wavenumber ($k = \frac{2\pi f}{c}$) varies from 0.0024 to 0.009. The grid spacing is about 80 m by taking 10 points per minimum wavelength for the discretization. With an absorbing layer of 35 points the discrete model contains $182 \times 91 \times 101$ points, amounting to about 1.7 millions unknowns.

For the frequencies 5 and 10 Hz the wavenumber k varies from 0.006 to 0.018 and from 0.012 to 0.036. When we take 10 points per minimum wavelength for the discretization, the grid spacing is about 40 m and 20 m for 5 Hz and 10 Hz, respectively. With an absorbing layer of 35 points the discrete model contains $294 \times 182 \times 199$ points, amounting to about 11 millions unknowns for 5 Hz and $517 \times 293 \times 326$ points in total (approx. 50 million unknowns) for 10 Hz.

We evaluate the parallel code using 4, 9, and 18 processors of the SGI Altix 3700 system for the frequencies 2.5, 5 and 10 Hz. The performance of the convergence of the numerical results for the 3D salt dome model using different semi-coarsening strategies for $f = 2.5, 5$ and 10 Hz in terms of the number of iterations and the relative residuals are present in Figs. 12a–c.

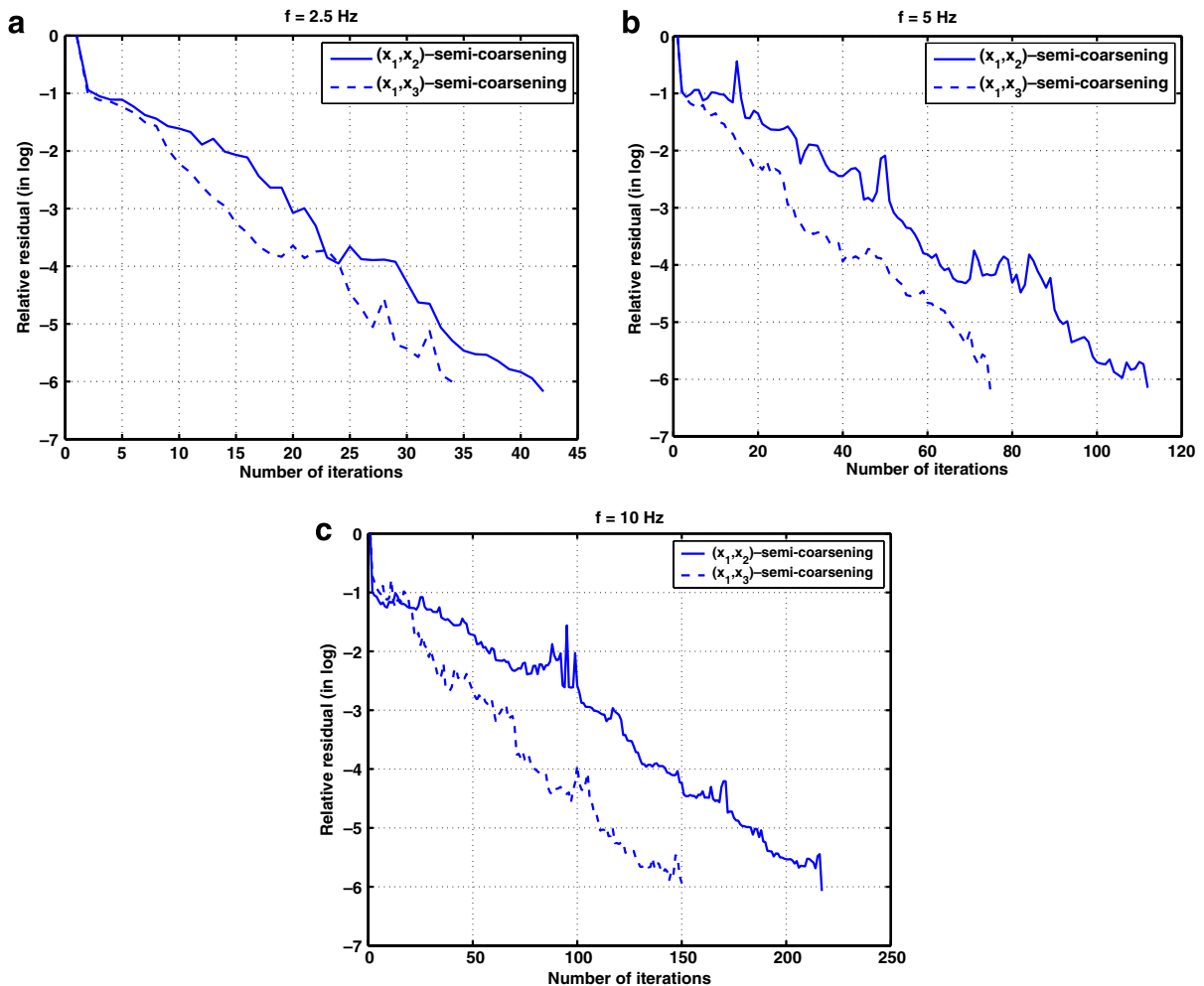


Fig. 12. The performance of the convergence of the numerical results for the 3D salt dome model using different coarsening strategy for: $f = 2.5$ Hz (a); $f = 5$ Hz (b) and $f = 10$ Hz (c) in terms of the number of iterations and the relative residual.

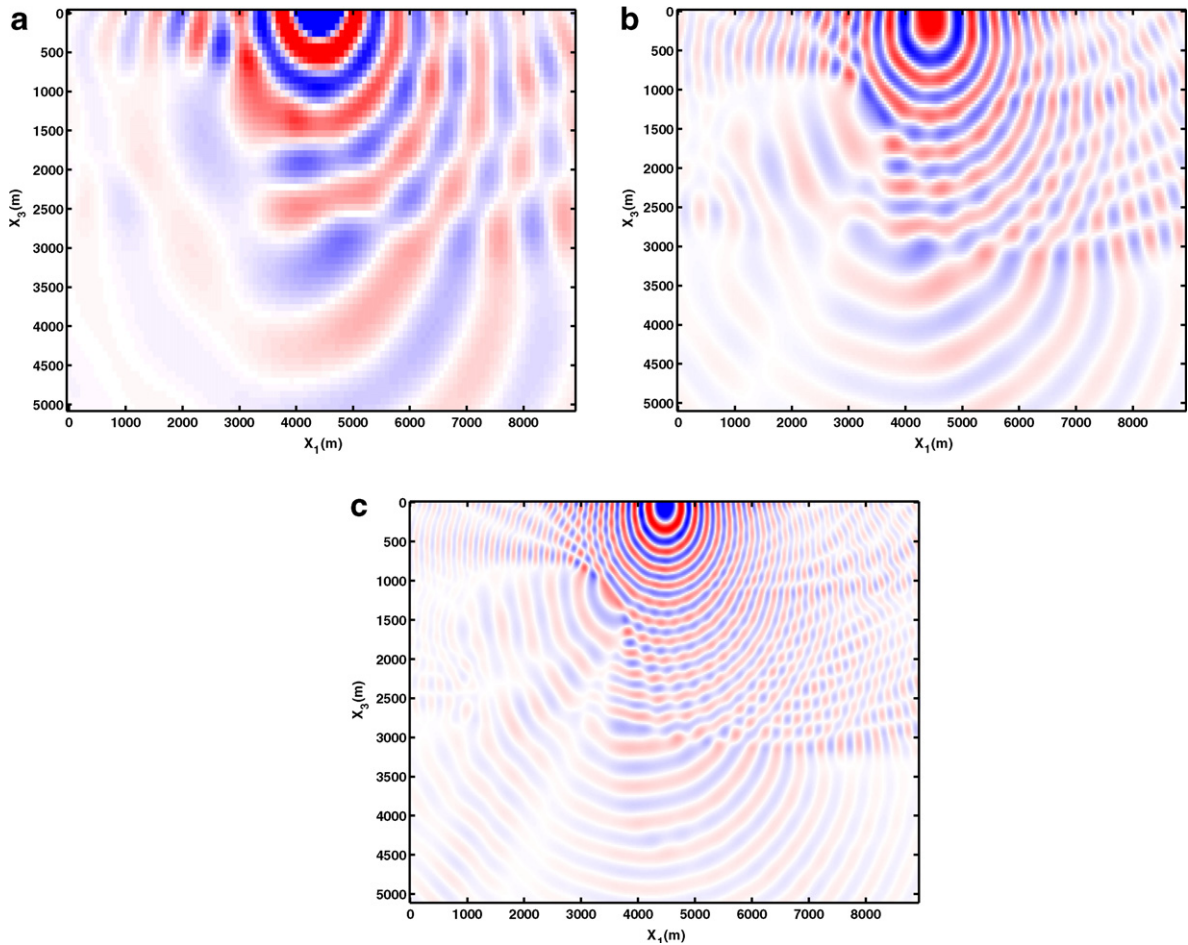


Fig. 13. The snapshots of the real part of the numerical results at $x_2 = 1800$ m for (a) $f = 2.5$ Hz; (b) $f = 5$ Hz and (c) $f = 10$ Hz, respectively.

In Figs. 12a–c, the solid line denotes the convergence of the preconditioned Bi-CGSTAB method using (x_1, x_2) -semi-coarsening strategy and the dashed line denotes the convergence of the preconditioned Bi-CGSTAB method using (x_1, x_3) -semi-coarsening strategy. From these results, we conclude that the (x_1, x_3) -semi-coarsening performs best. More generally, we again choose the direction in which k varies least for line-wise smoother.

In Figs. 13a–c, we display the snapshots of the real parts of the numerical solution of the wave field at $y = 1800$ m for 2.5 Hz, 5 Hz and 10 Hz, respectively. From Figs. 13a–c, one can observe the wave field propagating from the source at (4460, 2220) m through the model. We can also see the complexity of the wave field due to the heterogeneous regions. The faster velocities in the salt and in deeper sediment layers cause a larger wavelength. Because of the strong contrast with the embedding medium less energy penetrates the salt dome especially for higher frequencies (see Fig. 13c).

6. Conclusions

In this paper, we have presented a parallel iterative solution method for 3D high frequency Helmholtz problems in media with heterogeneities. The iterative method is based on Bi-CGSTAB, preconditioned by a complex-shifted Laplacian. The preconditioning operator is approximately inverted by a multigrid variant with 2D-semi-coarsening and line-wise smoothing. We have shown that the 2D-semi-coarsening strategy is

geometry dependent. By choosing the lines in the direction in which wavenumber k varies least we observe the best overall convergence. Furthermore, we validated the numerical accuracy by a comparison with an analytical result for the constant wavenumber model. Numerical results that indicate the robustness and efficiency of the method have been presented for several test problems. The number of iterations increases linearly with the frequency for a fixed number of grid points per wavelength.

Using the parallel approach, we have been able to compute a 3D salt dome model arising in the geophysical applications for high frequency.

Acknowledgement

This work is part of a joint Project between Philips and Shell on “Rigorous modeling of 3D wave propagation in inhomogeneous media for geophysical and optical problems”. This Project was financially supported by the Dutch Ministry of Economic Affairs under Project number BTS01044. We thank SARA Computing and Networking Services (www.sara.nl) in Amsterdam, the Netherlands for giving permission to implement and execute our parallel code using the SGI Altix 3700 Aster and SGI Origin 3800 Teras systems.

Appendix A. The derivation of the matrix-dependent prolongation weights

In this appendix, we present the derivation of the matrix-dependent prolongation weights (23). We start by splitting \tilde{M} (20) into a symmetric and an antisymmetric part:

$$\tilde{M}_S = \frac{1}{2}(\tilde{M} + \tilde{M}^T), \quad \tilde{M}_T = \frac{1}{2}(\tilde{M} - \tilde{M}^T). \quad (23)$$

The elements of the symmetric and the antisymmetric parts are denoted by \tilde{m}_s and \tilde{m}_t , respectively. In our case, $\tilde{M}_T = 0$ on the finest grid only. On the coarse grids, due to the lumping, asymmetry enters the definition of the matrices. Therefore, we adopt the asymmetric strategy as presented in [37] in the definition of the interpolation weights. For the symmetric part, we have

$$\begin{aligned} d_w &= \max(|\tilde{m}_{s,2i_1,2i_2-1,i_3}^1 + \tilde{m}_{s,2i_1,2i_2-1,i_3}^4 + \tilde{m}_{s,2i_1,2i_2-1,i_3}^7|, |\tilde{m}_{s,2i_1,2i_2-1,i_3}^1|, |\tilde{m}_{s,2i_1,2i_2-1,i_3}^7|), \\ d_e &= \max(|\tilde{m}_{s,2i_1,2i_2-1,i_3}^3 + \tilde{m}_{s,2i_1,2i_2-1,i_3}^6 + \tilde{m}_{s,2i_1,2i_2-1,i_3}^9|, |\tilde{m}_{s,2i_1,2i_2-1,i_3}^3|, |\tilde{m}_{s,2i_1,2i_2-1,i_3}^9|), \\ d_n &= \max(|\tilde{m}_{s,2i_1-1,2i_2,i_3}^7 + \tilde{m}_{s,2i_1-1,2i_2,i_3}^8 + \tilde{m}_{s,2i_1-1,2i_2,i_3}^9|, |\tilde{m}_{s,2i_1-1,2i_2,i_3}^7|, |\tilde{m}_{s,2i_1-1,2i_2,i_3}^9|), \\ d_s &= \max(|\tilde{m}_{s,2i_1-1,2i_2,i_3}^1 + \tilde{m}_{s,2i_1-1,2i_2,i_3}^2 + \tilde{m}_{s,2i_1-1,2i_2,i_3}^3|, |\tilde{m}_{s,2i_1-1,2i_2,i_3}^1|, |\tilde{m}_{s,2i_1-1,2i_2,i_3}^3|), \\ \sigma_1 &= \frac{1}{2} \min \left(1, \left| 1 - \frac{\sum_{p=1}^9 |\tilde{m}_{s,2i_1,2i_2-1,i_3}^p|}{\tilde{m}_{s,2i_1,2i_2-1,i_3}^5} \right| \right), \\ \sigma_2 &= \frac{1}{2} \min \left(1, \left| 1 - \frac{\sum_{p=1}^9 |\tilde{m}_{s,2i_1-1,2i_2,i_3}^p|}{\tilde{m}_{s,2i_1-1,2i_2,i_3}^5} \right| \right). \end{aligned}$$

For the unsymmetric part, two parameters c_1 and c_2 are defined as:

$$\begin{aligned} c_1 &= \tilde{m}_{t,2i,2j-1,k}^3 + \tilde{m}_{t,2i,2j-1,k}^6 + \tilde{m}_{t,2i,2j-1,k}^9 - (\tilde{m}_{t,2i,2j-1,k}^1 + \tilde{m}_{t,2i,2j-1,k}^4 + \tilde{m}_{t,2i,2j-1,k}^7), \\ c_2 &= \tilde{m}_{t,2i-1,2j,k}^7 + \tilde{m}_{t,2i-1,2j,k}^8 + \tilde{m}_{t,2i-1,2j,k}^9 - (\tilde{m}_{t,2i-1,2j,k}^1 + \tilde{m}_{t,2i-1,2j,k}^2 + \tilde{m}_{t,2i-1,2j,k}^3). \end{aligned}$$

As mentioned, the elements of the symmetric and the antisymmetric parts are denoted by subscript s and t , respectively. Using these quantities the matrix-dependent weights on the west, east, north and south are determined as follows:

$$\begin{aligned} w_w &= \sigma_1 \left(1 + \frac{d_w - d_e}{d_w + d_e} + \frac{c_1}{d_w + d_e + d_n + d_s} \right), \quad w_e = 2\sigma_1 - w_w, \\ w_n &= \sigma_2 \left(1 + \frac{d_s - d_n}{d_s + d_n} + \frac{c_2}{d_w + d_e + d_n + d_s} \right), \quad w_s = 2\sigma_2 - w_n. \end{aligned}$$

The weights w_1, \dots, w_4 can now be computed, i.e.,

- for $(2i_1, 2i_2 - 1, i_3)$

$$w_{1_{2i_1, 2i_2-1, i_3}} = \min(2\sigma_1, \max(0, w_w)), w_{2_{2i_1, 2i_2-1, i_3}} = \min(2\sigma_1, \max(0, w_e)).$$

- for $(2i_1 - 1, 2i_2, i_3)$

$$w_{1_{2i_1-1, 2i_2, i_3}} = \min(2\sigma_1, \max(0, w_s)), w_{3_{2i_1-1, 2i_2, i_3}} = \min(2\sigma_1, \max(0, w_n)).$$

- for $(2i_1, 2i_2, i_3)$

$$w_{1_{2i_1, 2i_2, i_3}} = \frac{\tilde{m}_{2i_1, 2i_2, i_3}^1 + \tilde{m}_{2i_1, 2i_2, i_3}^2 \cdot w_{1_{2i_1, 2i_2-1, i_3}} + \tilde{m}_{2i_1, 2i_2, i_3}^4 \cdot w_{1_{2i_1-1, 2i_2, i_3}}}{\tilde{m}_{2i_1, 2i_2, i_3}^5},$$

$$w_{2_{2i_1, 2i_2, i_3}} = \frac{\tilde{m}_{2i_1, 2i_2, i_3}^3 + \tilde{m}_{2i_1, 2i_2, i_3}^2 \cdot w_{2_{2i_1, 2i_2-1, i_3}} + \tilde{m}_{2i_1, 2i_2, i_3}^6 \cdot w_{1_{2i_1+1, 2i_2, i_3}}}{\tilde{m}_{2i_1, 2i_2, i_3}^5},$$

$$w_{3_{2i_1, 2i_2, i_3}} = \frac{\tilde{m}_{2i_1, 2i_2, i_3}^7 + \tilde{m}_{2i_1, 2i_2, i_3}^4 \cdot w_{3_{2i_1-1, 2i_2, i_3}} + \tilde{m}_{2i_1, 2i_2, i_3}^8 \cdot w_{1_{2i_1, 2i_2+1, i_3}}}{\tilde{m}_{2i_1, 2i_2, i_3}^5},$$

$$w_{4_{2i_1, 2i_2, i_3}} = \frac{\tilde{m}_{2i_1, 2i_2, i_3}^9 + \tilde{m}_{2i_1, 2i_2, i_3}^6 \cdot w_{3_{2i_1+1, 2i_2, i_3}} + \tilde{m}_{2i_1, 2i_2, i_3}^8 \cdot w_{2_{2i_1, 2i_2+1, i_3}}}{\tilde{m}_{2i_1, 2i_2, i_3}^5}.$$

References

- [1] I. Babushka, F. Ihlenburg, E.T. Paik, S.A. Sauter, A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution, *Comput. Methods Appl. Mech. Engrg.* 128 (1995) 325–359.
- [2] A. Bayliss, C.I. Goldstein, E. Turkel, On accuracy conditions for the numerical computation of waves, *J. Comput. Phys.* 59 (1985) 396–404.
- [3] A. Bamberger, Second-order absorbing boundary conditions for the wave equation: a solution for the corner problem, *SIAM J. Numer. Anal.* 27 (1990) 323–352.
- [4] W.L. Briggs, V.E. Henson, S.F. McCormick, *A multigrid tutorial*, SIAM, Philadelphia, USA, 2000.
- [5] E. Brigham, *The fast Fourier transform and its application*, Prentice-Hall Inc., NJ, USA, 1988.
- [6] C.G. Broyden, M.T. Vespucchi, *Krylov solvers for linear algebraic system: Krylov solvers*, Elsevier, Amsterdam, 2004.
- [7] R. Clayton, B. Engquist, Absorbing boundary conditions for acoustic and elastic wave equations, *Bull. Seis. Soc. Am.* 67 (1977) 1529–1540.
- [8] A.T. De Hoop, *Handbook of radiation and scattering of waves; acoustic waves in fluids, elastic waves in solids, electromagnetic waves*, Academic Press, London, UK, 1995.
- [9] H.C. Elman, O.G. Ernst, D.P. O’Leary, A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations, *SIAM J. Sci. Comput.* 31 (2001) 1291–1315.
- [10] B. Engquist, A. Majda, Absorbing boundary conditions for the numerical simulation of waves, *Math. Comput.* 31 (1977) 629–651.
- [11] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, On a class of preconditioners for solving the Helmholtz equation, *Appl. Numer. Math.* 50 (2004) 409–425.
- [12] Y.A. Erlangga, C.W. Oosterlee, C. Vuik, A novel multigrid-based preconditioner for the heterogeneous Helmholtz equation, *SIAM J. Sci. Comput.* 27 (4) (2006) 1471–1492.
- [13] A. George, J.W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, NJ, USA, 1981.
- [14] N.A. Gumerov, R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*, Elsevier, Amsterdam, 2004.
- [16] C.-H. Jo, C. Shin, J.H. Suh, An optimal 9-point, finite difference, frequency space, 2-D scalar wave extrapolator, *Geophysics* 61 (1996) 529–537.
- [17] S. Kim, S. Kim, Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media, *SIAM J. Sci. Comput.* 24 (2002) 684–701.
- [18] K.R. Kelly, K.J. Marfurt, *Numerical modeling of seismic wave propagation*, Society of Exploration Geophysicists, No. 13, Tulsa, Oklahoma, USA, 1990.
- [19] A. Kononov, S.W. de Leeuw, C.D. Riyanti, C.W. Oosterlee, C. Vuik, Numerical performance of a parallel solution method for a heterogeneous 2D Helmholtz equation, in: P. Wesseling, C.W. Oosterlee, P. Hemker, (Eds.), *Proceeding of the Eighth European Multigrid Conference*, 2005.
- [20] A.L. Laird, M.B. Giles, *Preconditioning techniques for the solution of the 2D Helmholtz equation*, Report 02/12, Oxford Computer Laboratory, Oxford, UK, 2002.

- [21] B. Lee, T.A. Manteuffel, S.F. McCormick, J. Ruge, First-order system least-squares for the Helmholtz equation, *SIAM J. Sci. Comput.* 21 (2000) 1927–1949.
- [22] Q. Liao, G.A. McMechan, Multifrequency viscoacoustic modeling and inversion, *Geophysics* 61 (1996) 1371–1378.
- [24] R.M.M. Mattheij, S.W. Rienstra, J.H.M. Thije Boonkkamp, *Partial differential equation: modeling analysis computation*, SIAM, Philadelphia, 2005.
- [25] W.A. Mulder, R.-R. Plessix, How to choose a subset of frequencies in frequency-domain finite-difference migration, *Geophys. J. Int.* 158 (2004) 801–812.
- [26] W.A. Mulder, R.-E. Plessix, One-way and two-way wave-equation migration, *Geophysics* 69 (2004) 1491–1504.
- [27] P.S. Pacheco, *Parallel programming with MPI*, Morgan Kaufman Publisher Inc., CA, USA, 1997.
- [28] R.-E. Plessix, W.A. Mulder, Separation of variables as a preconditioner for an iterative Helmholtz solver, *Appl. Num. Math.* 44 (2003) 385–400.
- [29] R.-E. Plessix, W.A. Mulder, Frequency-domain finite-difference amplitude-preserving migration, *Geophys. J. Int.* 157 (2004) 975–987.
- [30] R.G. Pratt, Seismic waveform inversion in frequency domain. Part I: Theory and verification in a physical scale domain, *Geophysics* 64 (1999) 888–901.
- [31] C.D. Riyanti, Y.A. Erlangga, R.-E. Plessix, W.A. Mulder, C.W. Oosterlee, C. Vuik, A new iterative solver for the time-harmonic wave equation, *Geophysics* 71 (2006) E57–E63.
- [32] Y. Saad, *Iterative Methods for Sparse Linear Systems*, second ed., SIAM, Philadelphia, 2003.
- [33] I. Singer, E. Turkel, A perfectly matched layer for the Helmholtz equation in a semi-infinite strip, *J. Comput. Phys.* 201 (2004) 439–465.
- [34] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI-The Complete Reference, The MPI Core*, second ed., vol. 1, MIT Press, Cambridge, USA, 1998.
- [35] U. Trottenberg, C.W. Oosterlee, A. Schüller, *Multigrid*, Academic Press Inc., London, UK, 2000.
- [36] H.A. Van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 13 (1992) 631–644.
- [37] T. Washio, C.W. Oosterlee, Flexible multiple semi-coarsening for three-dimensional singularly perturbed problems, *SIAM J. Sci. Comput.* 19 (1998) 1646–1666.
- [38] P.M. De Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.* 33 (1990) 1–27.