Journal of Computational Physics 227 (2008) 8685–8698

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A domain decomposition solver for acoustic scattering by elastic objects in layered media

Kazufumi Ito, Zhonghua Qiao *, Jari Toivanen

Center for Research in Scientific Computation, North Carolina State University, Box 8205, Raleigh, NC 27695-8205, United States

ARTICLE INFO

Article history:
Received 15 February 2008
Received in revised form 16 June 2008
Accepted 18 June 2008
Available online 27 June 2008

Keywords:
Fast Helmholtz solver
Scattering from elastic targets
Domain decomposition preconditioner

ABSTRACT

A finite element solution procedure is presented for accurately computing time-harmonic acoustic scattering by elastic targets buried in sediment. An improved finite element discretization based on trilinear basis functions leading to fourth-order phase accuracy is described. For sufficiently accurate discretizations 100 million to 1 billion unknowns are required. The resulting systems of linear equations are solved iteratively using the GMRES method with a domain decomposition preconditioner employing a fast direct solver. Due to the construction of the discretization and preconditioner, iterations can be reduced onto a sparse subspace associated with the interfaces. Numerical experiments demonstrate capability to evaluate the scattered field with hundreds of wavelengths.

1. Introduction

We develop an efficient numerical method for computing time-harmonic acoustic wave scattering by an elastic object in three-dimensional layered media. It is capable of modeling the scattering of sonar signals by undersea targets located in or near the seabed in littoral environments with plane or rippled interface between water and sediment. One application for such problems is the detection of hazardous or/and lost objects buried in sediment. For this purpose it is essential to have a numerical approximation which can accurately predict the scattered field by such targets. Our model problem in this paper is the scattering by a solid aluminum cylinder buried in sediment. A similar experimental test setup was used in [30] to perform measurements. This is an exterior problem which has to be truncated into a bounded domain for a finite element discretization; see Fig. 1.

We consider scattering problems in a frequency range in which the size of the computational domain is of the order of 100 wavelengths. The discretization has to have sufficiently many nodes per wavelength. In the considered frequency range the phase (pollution) error of the solution with a second-order accurate discretization is large unless tens of nodes are used per wavelength [20]. This would lead to huge systems of linear equations which are computationally intractable. Many ways have been proposed to reduce the phase error; see [20], for example. We employ a generalization of the approach proposed in [13] in which standard bilinear finite elements are used for two-dimensional Helmholtz problems together with a modified quadrature rule. This leads to fourth-order phase accuracy and a higher order approximation on orthogonal uniform meshes for a constant speed of sound. Actually it coincides with the nine-point fourth-order compact finite difference scheme [29] in the homogeneous medium. A related fourth-order finite difference method for the Helmholtz equation was considered in [12]. The resulting matrix has exactly the same structure and complexity as the one obtained by using the standard Gauss quadrature rule. As shown in Section 5.1 it drastically increases the performance and capability of our finite element modeling. The scheme with eight points per wavelength sufficiently well achieves the resolution we need and thus leads to a linear system with much smaller numbers of unknowns.

* Corresponding author. Tel.: +1 919 760 5359.
E-mail addresses: kito@ncsu.edu (K. Ito), zqiao@ncsu.edu, mathqiao@hotmail.com (Z. Qiao), jatoivan@ncsu.edu (J. Toivanen).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.06.015

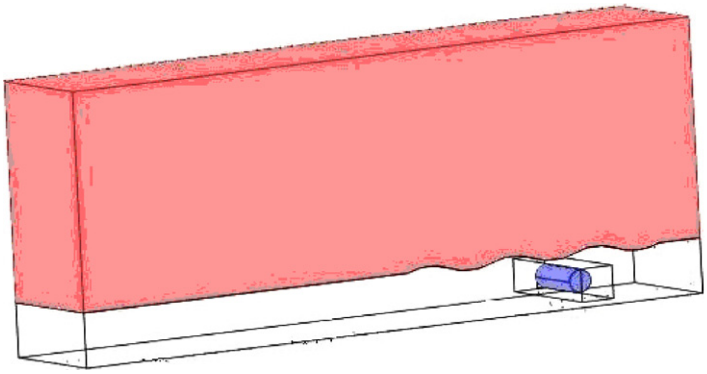

Fig. 1. The computational domain (Π) and domain decomposition to the near-field (Ω) and far-field domains ($\Pi \backslash \Omega$).

The resulting system of linear equations is too large to be solved using a direct method. We propose an iterative solution procedure with a decomposition preconditioner based on a domain embedding approach [15–17,21,22,24,26]. In our solution procedure the computational domain is decomposed into the near field subdomain which encloses an elastic target (the interior box in Fig. 1) and the far-field subdomain (the rest of the rectangle computational domain Π). In the far-field subdomain the discretization is based on an orthogonal uniform mesh which is locally adapted to the interface between the water and sediment. We use a multiplicative domain decomposition, i.e. we develop a preconditioning method for the resulting linear equation that partitions the matrix into the decoupled blocks. We will use a separable preconditioner based on the perfectly vertically layered media in our solution procedure for the far-field subdomain. The far-field preconditioner is based on a fast direct solver [16,33]. Since the media is vertically layered with the wavy interface, our preconditioner coincides with the system matrix except the rows corresponding to unknowns near the interfaces. Thus, we can reduce iterations on a small sparse subspace as has been shown in [22,24], for example, and Section 4.3. This procedure reduces GMRES iterates onto a sparse subspace which leads to much reduced storage and computational requirements. The GMRES iteration is evaluated by the partial solution method based on a cyclic reduction type fast direct solver. This reduction makes our iterative method extremely efficient and enables us to solve systems with billions of unknowns as our numerical examples demonstrate. With multigrid preconditioners like in [1,9,10] and with nonorthogonal grids in [28] such reduction cannot be made and the solution of the model problem would require much more memory and computation. With the separation-of-variables preconditioner in [31] the reduction of the iterations onto a sparse subspace can be made with the model problem.

A similar domain decomposition approach to the one considered here was used for two-dimensional problems in [4,5,23]. Specifically, the approach in [23] is extended to the three-dimensional case. Such extension is nontrivial and involves the use of higher order discretization in the layered media and efficient direct solver for the 3D separable preconditioner. Another preconditioning technique for scattering problems in layered media without an object has been considered in [1,10,28,31], for example, and with an object in [19].

The outline of the paper is the following. We begin by defining a model problem in Section 2. For the finite element discretization a weak formulation, meshes, a domain decomposition, the forming of linear systems and a modified quadrature rule are described in Section 3. The iteration with a domain decomposition preconditioner is consider in Section 4. Also, the solution procedures for the far-field and near-field preconditioners are given and how iterations can be performed in a sparse subspace is described. Numerical experiments are presented in Section 5. They include an accuracy study for the discretizations and an efficiency study of the iterative solution. The paper ends with the conclusions in Section 6.

2. Model problem

We truncate the exterior domain into a rectangular parallelepiped domain Π shown in Fig. 1. We pose an absorbing boundary condition $\mathcal{B}p = 0$ on the truncation boundary $\partial\Pi$. For the water and sediment fluid region ($\Pi \backslash \Omega$) we use the Helmholtz equation to model the time-harmonic pressure variations p. In the elastic object (Ω) the displacement u is described by the time-harmonic solution for the linear elastic wave equation. Coupling these equations leads to the partial differential equation model:

$$
\begin{aligned}
&\nabla \cdot \frac{1}{\rho} \nabla p + \frac{k^2}{\rho} p = g \quad \text{in} \quad \Pi \setminus \Omega \\
&\frac{1}{\rho} \frac{\partial p}{\partial n} = \omega^2 u \cdot n, \quad -pn = \sigma(u) n \quad \text{on} \quad \partial\Omega \\
&\nabla \cdot \sigma(u) + \omega^2 \rho u = 0 \quad \text{in} \quad \Omega \\
&\mathcal{B}p = 0 \quad \text{on} \quad \partial\Pi,
\end{aligned}
\tag{1}
$$

where ρ is a piecewise constant density, ω is the angular frequency, $k = \omega/c$ is the wavenumber and c is a piecewise constant speed of sound. The function g is a compactly supported sound source and n denotes the outward unit normal. The stress tensor σ and the strain tensor ϵ are defined by

$$\sigma(u) = 2\mu\epsilon(u) + \lambda\nabla\cdot uI \quad \text{and} \quad \epsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^{\mathrm{T}}).$$

In our model problem the Lame constants μ and λ are constants in the elastic object. They are defined in terms of the compressional speed c_{c} and the shear speed c_{s} by

$$\lambda = \rho(c_{\mathrm{c}}^2 - 2c_{\mathrm{s}}^2) \quad \text{and} \quad \mu = \rho c_{\mathrm{s}}^2.$$

As the boundary condition $\mathcal{B}p = 0$ in (1) we employ the second-order absorbing boundary condition described in [2,16,23]. For practical purposes this usually leads to sufficiently small reflections from the truncation boundary $\partial\Pi$. An alternative would be to use a perfectly matched layer (PML) next to the truncation boundary $\partial\Pi$. A properly chosen PML would allow to use slightly smaller computational domain without reducing the accuracy. Using the formulation and implementation of the PML described in [16,18] the computational efficiency would increase due to a smaller computational domain. The six rectangular faces of the boundary $\partial\Pi$ are denoted by $\Gamma_{\pm j}$, $j = 1, 2, 3$, whose outward normal directions are given by the coordinate directions $\pm x_j$. The second-order absorbing boundary conditions on these faces reads

$$\pm\frac{\partial p}{\partial x_j} - \mathrm{i}kp - \frac{\mathrm{i}}{2k}\sum_{1\leqslant l\neq j\leqslant 3}\frac{\partial^2 p}{\partial x_l^2} = 0. \tag{2}$$

Furthermore, we have the conditions

$$-\frac{3}{2}k^2 p - \mathrm{i}k\left(\pm\frac{\partial p}{\partial x_m} \pm \frac{\partial p}{\partial x_l}\right) - \frac{1}{2}\frac{\partial^2 p}{\partial x_j^2} = 0 \tag{3}$$

on each edge denoted by $\Gamma_{(\pm m,\pm l)}$ between the faces $\Gamma_{\pm m}$ and $\Gamma_{\pm l}$ and the conditions

$$-2\mathrm{i}kp + \sum_{l=1}^{3}\pm\frac{\partial p}{\partial x_l} = 0 \tag{4}$$

on the eight corners of the rectangular parallelepiped Π denoted by Ψ.

3. Finite element discretization

In this section, we describe our finite element discretization of the model problem.

3.1. Weak form

For the finite element discretization, we have the weak formulation for the coupled partial differential equation (1) with the absorbing boundary condition defined by (2)–(4): Find

$$p \in V = \{v \in H^1(\Pi\setminus\overline{\Omega}) : v|_{\partial\Pi} \in H^1(\partial\Pi), v|_{\Gamma_{(m,j)}} \in H^1(\Gamma_{(m,j)})\forall\Gamma_{(m,j)} \in \Phi\}$$

and $u \in H^1(\Omega)^d$ such that

$$\begin{aligned}&\int_{\Pi\setminus\Omega}\frac{1}{\rho}(\nabla p\cdot\nabla q - k^2 pq)\mathrm{d}x - \mathrm{i}k\int_{\partial\Pi}\frac{1}{\rho}pq\,\mathrm{d}s + \frac{\mathrm{i}}{2k}\sum_{k=1}^{3}\sum_{j\neq k}\left(\int_{\Gamma_k}\frac{1}{\rho}\frac{\partial p}{\partial x_j}\frac{\partial q}{\partial x_j}\mathrm{d}s + \int_{\Gamma_{-k}}\frac{1}{\rho}\frac{\partial p}{\partial x_j}\frac{\partial q}{\partial x_j}\mathrm{d}s\right)\\ &\quad + \frac{3}{4}\sum_{\Gamma_{(m,j)}\in\Phi}\int_{\Gamma_{(m,j)}}\frac{1}{\rho}pq\,\mathrm{d}l - \frac{1}{4k^2}\sum_{\Gamma_{(m,j)}\in\Phi}\int_{\Gamma_{(m,j)}}\frac{1}{\rho}\frac{\partial p}{\partial x_k}\frac{\partial q}{\partial x_k}\bigg|_{k\neq m,j}\mathrm{d}l + \frac{i}{2k}\sum_{x\in\Psi}\frac{1}{\rho}p(x)q(x) + \int_{\partial\Omega}(\omega^2 u\cdot nq + v\cdot np)\mathrm{d}s\\ &\quad + \int_{\Omega}(\sigma(u):\epsilon(v) - \omega^2\rho u\cdot v)\mathrm{d}x = \int_{\Pi\setminus\overline{\Omega}}gq\,\mathrm{d}x\end{aligned} \tag{5}$$

for all $(q,v) \in V\times H^1(\Omega)^d$. Here n denotes the unit outward normal vector of $\partial\Omega$ and Φ is the set of all edges of $\partial\Pi$.

3.2. Domain decomposition and meshes

For the finite element mesh generation and also for the solution procedure, the computational domain Π is divided into two parts: the near-field domain Ω_1 which includes the elastic object Ω and the far-field domain $\Omega_2 = \Pi\setminus\Omega_1$. The near-field domain Ω_1 is a rectangular parallelepiped which is only slightly larger than the elastic object Ω. Thus, the far-field domain is vastly larger than the near-field one. For the model problem the domain decomposition is shown in Fig. 1.

We first describe the generation of the mesh for the far-field domain Ω_2. We start by defining a uniform orthogonal grid in such a way that the corners of the near-field domain Ω_1 are grid points. We adapt it to the interface between water and sediment in order to have a second-order accurate approximation for the interface. We assume that the interface is defined by the equation $x_3 = f(x_1,x_2)$ with a given function f. For each grid line in the x_3-direction we find the point which is closest to the interface and we move it in the x_3-direction onto the interface. Then we define hexahedral elements given by the grid cells. The hexahedral elements which are cut into two parts by the interface are replaced by a few polyhedral elements, for example, prism elements. This is performed so that the interface does not cut any elements and on the cell faces which are not cut by the interface the element faces are quadrilaterals. A similar mesh generation method for two-dimensional problems is described in [6].

For the near-field domain Ω_1 we use an unstructured mesh which is conforming on the interface between the near-field and far-field domains, that is, the faces of elements on this interface match for elements in both domains. Fig. 2 shows a crosscut of a coarse mesh.

3.3. Forming the system of linear equations

On hexahedral and prism elements we use trilinear and bilinear Lagrangian basis functions, respectively. For the Helmholtz operator the coefficient matrix can be assembled from the element stiffness matrices K_e and the element mass matrices M_e defined by

$$K_e = \int_{\Omega_e} \nabla N(x)^T \nabla N(x) dx \quad \text{and} \quad M_e = \int_{\Omega_e} N(x)^T N(x) dx, \tag{6}$$

where $N(x)$ is a vector-valued function containing the values of nonzero basis functions in the finite element Ω_e. The elasticity operator and boundary terms can be assembled in a similar manner element by element.

It is easier to perform integrations on a simple reference element instead of an element Ω_e. This is performed in the following way for a hexahedral element. Other elements can be treated in the same way. We choose the hexahedral reference element to be $[-1, 1] \times [-1, 1] \times [-1, 1]$. The function $\widehat{N}(\xi) \in \mathbb{R}^8$ gives the values of Lagrangian basis functions associated with the corners of the reference element. We denote the mapping from the reference element into the element Ω_e by $x(\xi)$, where $\xi \in [-1, 1]^3$. Furthermore, we denote the Jacobian of $x(\xi)$ by $J(\xi)$. The local element matrices in (6) can be computed as

$$\begin{aligned} M_e &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \widehat{N}(\xi)^T \widehat{N}(\xi) \det J(\xi) d\xi_1 d\xi_2 d\xi_3 \quad \text{and} \\ K_e &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} G(\xi)^T G(\xi) \det J(\xi) d\xi_1 d\xi_2 d\xi_3, \end{aligned} \tag{7}$$

where $G(\xi) = (J(\xi)^T)^{-1} \nabla \widehat{N}(\xi)$. Usually the integrals in (7) are approximated using a quadrature rule

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \psi(\xi) d\xi_1 d\xi_2 d\xi_3 \approx \sum_{j=1}^{n_q} w_j \psi(\xi^j) \tag{8}$$

instead of evaluating them analytically. A commonly used tensor product Gauss quadrature is defined by: $w_j = 1$, $\xi^j = (\pm\sqrt{1/3}, \pm\sqrt{1/3}, \pm\sqrt{1/3})^T$, $j = 1,\ldots,n_q = 8$. This rule leads to the exact local element matrices in (7) when the mapping $x(\xi)$ is linear.

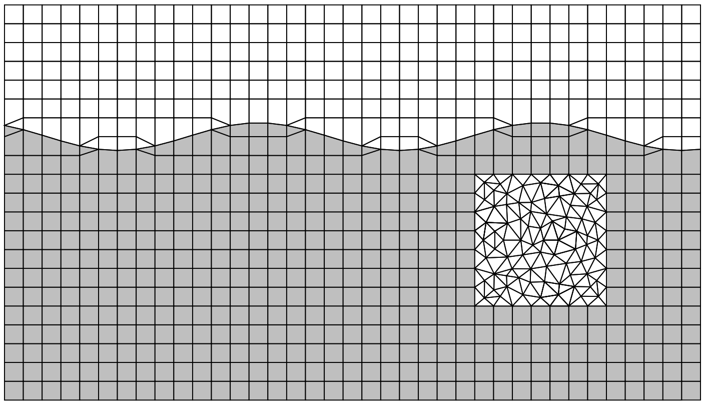

Fig. 2. A crosscut of a small coarse mesh in the x_1x_3-plane.

The assembling process leads to a system of linear equations

$$Ax = b, \tag{9}$$

where the matrix A has entries with nonzero imaginary part due to absorption and it is non-Hermitian.

3.4. Modified quadrature rule

The finite element discretizations of acoustic scattering problems have some error in the speed of sound. The size of this error depends on the order of the finite element, the number of mesh nodes per wavelength, and on the propagation direction. This error causes a phase error in the solution which can accumulate to be large when the domain is large in terms of wavelengths. The phase error is also called the pollution error and it has been considered in [20], for example. For the trilinear elements the discrete wavelength is second-order accurate with respect to the number of nodes per wavelength. There are several ways to reduce this error. Here we study the use of an extension of a modified quadrature rule proposed for two-dimensional problems in [13]. This approach is well-suited for our discretizations and solution procedure.

Using the generic form of the quadrature in (8), the modified rule is defined by: $w_j = 1$, $\xi^j = (\pm\sqrt{2/3}, \pm\sqrt{2/3}, \pm\sqrt{2/3})^{\mathrm{T}}$, $j = 1, \ldots, n_q = 8$. That is, the quadrature points $\pm\sqrt{1/3}$ are replaced by $\pm\sqrt{2/3}$. It was shown in [13] that for two-dimensional problems this modification leads the discrete wavelength to be fourth-order accurate with respect to the number of nodes per wavelength on uniform orthogonal meshes.

4. Iterative solution

4.1. Preconditioned iteration

The domain decomposition and discretization in Section 3.2 leads the system of linear equations (9) to have the block form

$$Ax = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = b, \tag{10}$$

where the first block row corresponds to the interior of the near-field subdomain Ω_1 and the second block row corresponds to the far-field subdomain Ω_2 and the interface between the subdomains formed by four faces of a rectangular parallelepiped.

Based on the block structure in (10), we define an upper triangular domain decomposition preconditioner

$$B = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & B_{22} \end{pmatrix}. \tag{11}$$

Thus, the first block row is taken from the matrix A. The preconditioner block B_{22} is a Schur complement matrix which will be described in Section 4.2. It is constructed in such a way that it corresponds to a scattering problem in the far-field domain Ω_2 with a nonlocal boundary condition on the interface between the subdomains. The preconditioner block A_{11} corresponds to a Dirichlet boundary value problem in the near-field domain Ω_1. For these reasons, B is a Neumann–Dirichlet-type domain decomposition preconditioner. They lead to a well conditioned matrix $B^{-1}A$ for the Poisson equation and, thus, a rapid convergence of a preconditioned iteration; see [35], for example. Based on this we can expect the conditioning to be good for low frequency problems. See the discussion in Section 5.2.

We use B as a right preconditioner and, thus, we solve iteratively the system of linear equations

$$AB^{-1}v = b \tag{12}$$

instead of the original system $A\,x = b$ in (9). Once we have obtained the solution v of (12), we get x by solving the system $B\,x = v$. For the iterations we use the GMRES method [34] without restart.

4.2. Far-field preconditioner

By discretizing the Helmholtz equation in the domain Π, without the object Ω and with a perfectly horizontal surface of the sediment, on a fully orthogonal $n_1 \times n_2 \times n_3$ mesh we obtain a matrix

$$\widetilde{C} = \begin{pmatrix} C_{ee} & C_{e2} \\ C_{2e} & C_{22} \end{pmatrix}, \tag{13}$$

where the blocks correspond to our domain decomposition. The matrix block C_{ee} corresponds to an acoustic scattering problem without a scatterer in the near-field subdomain Ω_1. The far-field preconditioner is the Schur complement matrix

$$B_{22} = C_{22} - C_{2e}C_{ee}^{-1}C_{e2}. \tag{14}$$

The solution y_2 of the linear system $B_{22}y_2 = f_2$ is obtained as the second block of $y = (y_e\, y_2)^T$ after solving the linear system

$$\widetilde{C}y = \begin{pmatrix} C_{ee} & C_{e2} \\ C_{2e} & C_{22} \end{pmatrix}\begin{pmatrix} y_e \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ f_1 \end{pmatrix}.$$

By renumbering the unknowns first in the x_3-direction, then in the x_2-direction, and last in the x_1-direction the matrix $\widetilde{C}$ has a tensor product form

$$C = H_1 \otimes M_2 \otimes M_3 + M_1 \otimes H_2 \otimes M_3 + M_1 \otimes M_2 \otimes (H_3 - \widetilde{M}_3). \tag{15}$$

The matrices H_1 and H_2 are stiffness matrices for one-dimensional problems in the x_1 and x_2 directions, respectively, with modifications on the boundaries due to the absorbing boundary condition. Similarly, M_1 and M_2 are mass matrices for one-dimensional problems in the x_1 and x_2 directions, respectively, with modifications on the boundaries. The matrices M_3 and H_3 are stiffness and mass matrices in the x_3-direction which are scaled by the inverse of the density ρ. Furthermore, $\widetilde{M}_3$ is a mass matrix in the x_3-direction which is scaled by the wave number squared k^2 divided by the density ρ. More precisely these matrices are as follows.

The dimension of the matrices H_1 and M_1 is n_1 and they are given by

$$H_1 = \frac{1}{h}\begin{pmatrix} 1 - \mathrm{i}hk/2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 - \mathrm{i}hk/2 \end{pmatrix},$$

and

$$M_1 = \frac{h}{6}\begin{pmatrix} 2 + 3\mathrm{i}/(hk) & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 2 + 3\mathrm{i}/(hk) \end{pmatrix},$$

where h denotes the mesh step size in all directions and k is the wavenumber in the sediment. The $n_2 \times n_2$ matrices H_2 and M_2 are defined in the same way.

The $n_3 \times n_3$ matrices H_3, M_3 and $\widetilde{M}_3$ correspond to one-dimensional problems in the x_3-direction. They can be assembled from the element matrices as

$$\begin{aligned} H_3 &= \mathrm{diag}\left\{-\frac{\mathrm{i}k_1}{2\rho_1}, 0, \ldots, 0, -\frac{\mathrm{i}k_{n_3-1}}{2\rho_{n_3-1}}\right\} + \sum_{j=1}^{n_3-1} \frac{1}{\rho_j} P_j^T \frac{1}{h}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} P_j, \\ M_3 &= \mathrm{diag}\left\{\frac{\mathrm{i}}{2k_1\rho_1}, 0, \ldots, 0, \frac{\mathrm{i}}{2k_{n_3-1}\rho_{n_3-1}}\right\} + \sum_{j=1}^{n_3-1} \frac{1}{\rho_j} P_j^T \frac{h}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} P_j, \quad \text{and} \\ \widetilde{M}_3 &= \mathrm{diag}\left\{\frac{\mathrm{i}k_1}{2\rho_1}, 0, \ldots, 0, \frac{\mathrm{i}k_{n_3-1}}{2\rho_{n_3-1}}\right\} + \sum_{j=1}^{n_3-1} \frac{k_j^2}{\rho_j} P_j^T \frac{h}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} P_j, \end{aligned} \tag{16}$$

where P_j is a $2 \times n_3$ matrix with zeros everywhere except in columns j and $j + 1$ which form a 2×2 identity block matrix. The coefficients ρ_j and k_j are the density and wavenumber on the jth one-dimensional element in the x_3-direction.

With the modified quadrature rule in Section 3.4 the matrices H_1, H_2, and H_3 stay the same. The matrix M_1 is replaced by

$$M_{1m} = \frac{h}{12}\begin{pmatrix} 5 + 6\mathrm{i}/(hk) & 1 & & & \\ 1 & 10 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 10 & 1 \\ & & & 1 & 5 + 6\mathrm{i}/(hk) \end{pmatrix}$$

and similarly M_2 is replaced by M_{2m} defined the same way as M_{1m}. The matrices M_3 and $\widetilde{M}_3$ are replaced by M_{3m} and $\widetilde{M}_{3m}$, respectively, which are obtained by replacing the matrices

$$\frac{h}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad \text{by} \quad \frac{h}{12}\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$$

in (16).

4.3. Iteration on a sparse subspace

Here, we consider the structure of the vectors needed during the iterative solution of (12). Particularly, we show that the vectors are very sparse. This reduces memory usage by orders of magnitude and due to this it is usually possible to use the GMRES method without restarts. For a more detailed discussion on iterations on sparse subspaces we refer to [22,24].

The jth GMRES iteration for solving the right preconditioned system $AB^{-1}v = b$ finds a vector v^j from the Krylov subspace

$$\text{span}\ \{b, AB^{-1}b, \ldots, (AB^{-1})^{j-1}b\}, \tag{17}$$

which minimizes the norm of the residual $AB^{-1}v^j - b$. We denote the vectors in the definition of the Krylov subspace as $b^1 = AB^{-1}b, \ldots, b^{j-1} = (AB^{-1})^{j-1}b$. The vector v^j is a linear combination of these vectors. Let us consider their structure. For b^1, we have the identity

$$b^1 = AB^{-1}b = b + (A - B)B^{-1}b. \tag{18}$$

From this we see that $b^1 \in X = \text{span}\{b\} + \text{range}(A - B)$. Similarly we can observe that for $l = 2, \ldots, j - 1$,

$$b^l = (AB^{-1})^l b = AB^{-1}b^{l-1} = b^{l-1} + (A - B)B^{-1}b^{l-1}.$$

From this we see by induction that $b^l \in X$ for $l = 2, \ldots, j - 1$, and, thus, the Krylov subspace in (17) is a subspace of X.

Let us now consider the sparsity structure of the subspace X. Based on the above conclusion this will be also the sparsity structure of the vectors needed during the GMRES iteration. We make an assumption that the vector b has at most $\mathcal{O}(n_1 n_2)$ nonzero components. This means that the sound source can be a point, a line, or a surface. In our model problem the source is a point. The matrix $A - B$ has the block structure

$$A - B = \begin{pmatrix} 0 & 0 \\ A_{21} & A_{22} - B_{22} \end{pmatrix}. \tag{19}$$

From this it follows that the first block of vectors in range$(A - B)$ is a zero block, that is, the part corresponding to the near-field is zero. Due to the block A_{21} in $A - B$ the vector components associated to the interface between the near-field and far-field are nonzero in general.

The structure of the matrix block $A_{22} - B_{22}$ in (19) is the most important one when considering sparsity of vectors in range$(A - B)$ as this far-field block is usually orders of magnitude larger than the near-field block. Due to our construction of A_{22} and B_{22} they differ in the neighborhood of the surface of the sediment and on the interface between the far-field and near-field domains. Near the surface of the sediment there are two reasons for differences. One is that the mesh used to compute A is locally adapted to the surface while the mesh for B_{22} is not. From this it follows that the vector components corresponding to nodes on adapted elements are nonzero in general. The other reason is that in the construction of B_{22} the wavy sediment surface was replaced by a straight one. This means that the material properties in the original problem and the problem defining the preconditioner mismatch in the volume between the wavy and straight surfaces. Hence, the vector components associated to mesh nodes in this volume are nonzero. Assuming that the deviation of the wavy surface from the straight one is order of the mesh step size the number of nonzero components of the vectors is $\mathcal{O}(n_1 n_2)$. Fig. 3 shows the sparse subspace X for a crosscut of a problem.

Under the above assumptions vectors in the sparse subspace X have $\mathcal{O}(n_1 n_2)$ nonzero components. Thus, they can be stored as $\mathcal{O}(n_3)$ times fewer floating point numbers than full vectors. Taking advantage of this in the GMRES iteration reduces the memory and computational requirements essentially. More precisely, the needed memory is reduced by the factor $\mathcal{O}(n_3)$ while the computational cost is reduced by the factor $\mathcal{O}(jn_3)$, where j is the number of GMRES iterations.

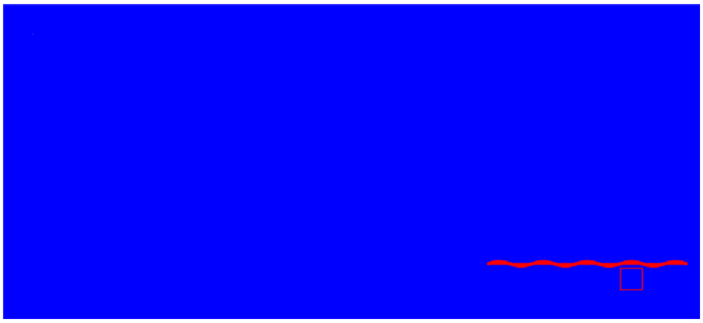

Fig. 3. The sparse subspace (denoted by red in color print or gray in b/w print) on a x_1x_3-crosscut for a partly rippled sediment.

4.4. Partial solution technique for far-field preconditioner

During the iterative solution it is necessary to solve linear systems $C\,v = y$, where y belongs to a sparse subspace Y and, furthermore, only the components of the solution y corresponding to this subspace are needed. These kinds of problems are called partial solution problems and for problems with a tensor product matrix C a partial solution technique has been developed in [3,27]. With this technique the solution requires $\mathcal{O}(n_1 n_2 n_3 \log n_3)$ operations when assuming that the dimension of Y is $\mathcal{O}(n_2 n_3)$ and the following implementation is used.

The partial solution technique is based on the diagonalization of the matrices H_1 and M_1 in the tensor product form (15). For this purpose let us consider the generalized eigenvalue problem

$$H_1 w_j = \lambda_j M_1 w_j,$$

where (λ_j,w_j) is the jth eigenpair, $j = 1,\ldots,n_1$, and the eigenvector w_j is normalized so that $w_j^{\mathrm{T}} M_1 w_j = 1$. The properties of these eigenvalue problems have been studied in [11]. Based on these pairs we define the matrices $\Lambda = \operatorname{diag}\{\lambda_1,\ldots,\lambda_{n_1}\}$ and $W = (\,w_1 \quad \cdots \quad w_{n_1}\,)$. Then we have that $W^{\mathrm{T}} H_1 W = \Lambda$ and $W^{\mathrm{T}} M_1 W = I$. Here and in the following, we denote the $n_j \times n_j$ identity matrix by I_j. With these identities and properties of tensor products, we obtain

$$\begin{aligned}\widehat{C} &= (W^{\mathrm{T}} \otimes I_2 \otimes I_3) C (W \otimes I_2 \otimes I_3)\\ &= \Lambda \otimes M_2 \otimes M_3 + I_1 \otimes H_2 \otimes M_3 + I_1 \otimes M_2 \otimes (H_3 - \widetilde{M}_3),\end{aligned} \tag{20}$$

which is a block diagonal matrix having n_1 diagonal blocks of size $n_2 n_3 \times n_2 n_3$. Using the diagonalization (20) the linear system $C\,v = y$ can be solved in the following three steps:

1. Compute $\hat{y} = (W^{\mathrm{T}} \otimes I_2 \otimes I_3)y$.
2. Solve $\widehat{C}\hat{v} = \hat{y}$.
3. Compute $v = (W \otimes I_2 \otimes I_3)\hat{v}$.

Let us consider the steps 1 and 3. The straightforward implementation of the multiplications in these steps require $\mathcal{O}(n_1^2 n_2 n_3)$ operations. This can be reduced by taking advantage of the sparsity of y and the required components of v. In the step 1. only the nonzero terms of y need to be multiplied and in the step 3. only the needed components of v are computed. We assumed in the above that y has $\mathcal{O}(n_2 n_3)$ nonzero components and the same amount of the components of v are required. Then the computational cost can be reduced to $\mathcal{O}(n_1 n_2 n_3)$ operations for both steps.

Due to the block diagonal structure of $\widehat{C}$, the step 2 decouples to n_1 solutions of linear systems with $n_2 n_3 \times n_2 n_3$ block tridiagonal matrices. Each of these systems can be solved efficiently using, for example, the cyclic reduction type fast direct solver introduced in [36] and further considered in [16,25,33]. This method recursively solves partial solution problems. When this is based on a split in the x_3-direction there are $\mathcal{O}(\log n_3)$ recursion levels and at each level the solution requires $\mathcal{O}(n_2 n_3)$ operations. For the detailed description of this method we refer to [16,32,33]. Thus, the computational cost to solve each of these n_1 problems is $\mathcal{O}(n_2 n_3 \log n_3)$ operations and the total cost of the step 2 is $\mathcal{O}(n_1 n_2 n_3 \log n_3)$ operations.

A memory efficient way to implement the above three step procedure is to perform the steps for one eigenpair (λ_j,w_j) at a time. Thus, the steps are to multiply y by $w_j^{\mathrm{T}} \otimes I_2 \otimes I_3$, to solve the $n_2 n_3 \times n_2 n_3$ linear system corresponding to λ_j, and compute only the required component of the multiplication by $w_j \otimes I_2 \otimes I_3$. This way, storing the intermediate vectors requires only $\mathcal{O}(n_2 n_3)$ double precision numbers. A well-suited data structure for the sparsity pattern is to have three integer arrays storing the index triplet for each nonzero component. Using these arrays the above steps 1. and 3. can be implemented easily.

4.5. Multiplications in each iteration

During the GMRES iteration we need to perform multiplications of the form $g = A\,B^{-1}v$ with a given vector v belonging to the subspace X. Each of these multiplications can be performed in three steps:

1. Solve

$$B_{22} y_2 = (C_{22} - C_{2e} C_{ee}^{-1} C_{e2}) y_2 = v_2$$

using the partial solution method as described in Sections (4.2) and (4.4).

2. Solve

$$A_{11} y_1 = -A_{12} y_2 \tag{21}$$

using the near-field solver as described in Section 4.6.

3. Compute

$$g_2 = A_{21} y_1 + A_{22} y_2.$$

As the resulting vector g belongs on the sparse subspace X the first vector block y_1 of y is zero block and the second block y_2 is very sparse as described in Section 4.3.

4.6. Near-field solver

We form and solve the near-field system of linear equations in (21) using the COMSOL Multiphysics software [7]. This problem is a discretized coupled fluid-structure problem in the near-field subdomain Ω_1 with a Dirichlet boundary condition on the interface between the subdomains. In the solution we employ the GMRES solver with a two level geometric multigrid preconditioner [9,14]. The coarse grid problem is solved using UMFPACK (unsymmetric multifrontal sparse LU factorization package) [8]. The projection of the residual vector onto the coarser level is obtained by multiplying it by the transpose of the prolongation (interpolation) matrix. The presmoothing is performed using SOR and the postsmoothing is performed also by SOR, but with reverse ordering of unknowns. The relaxation parameter in SOR is $\omega = 0.25$.

5. Numerical experiments

5.1. Grid refinement analysis

In this section, we present a grid refinement analysis for the second-order and fourth-order discretization. Our analysis is conducted for an acoustic scattering problem in the layered media with the straight sediment and no target. The test frequency is 3.75 kHz. The computational domain is [0 m,12 m] × [−0.4 m, 0.4 m] × [−0.8 m, 4 m] and the point source is located at [0.4 m, 0 m, 3.6 m]. The plane interface between the water and sediment is at $x_3 = 0$ m. The density of water is 1000 kg/m^3 and the speed of sound in it is 1482 m/s. In the sediment the density is 2000 kg/m^3 and the speed of sound is (1668–16.8i)m/s, where the imaginary part attenuates waves.

For this problem, we do not have the analytic solution. In order to study the order of convergence, we compared the computed solutions against the solution $q_{2401\times161\times961}$ obtained using the fourth-order discretization on the finest mesh given by a 2401 × 161 × 961 grid. The l_2-norm approximate error is given by

$$E_N(\cdot) = \frac{\|\cdot_N - q_{2401\times161\times961}\|_2}{\|q_{2401\times161\times961}\|_2}.$$

We define the order of convergence as

$$\mathrm{Order}(\cdot) = \log_2 \frac{E_{N/2}(\cdot)}{E_N(\cdot)}.$$

In Table 1, we report the results for the second-order solution (p) and the fourth-order solution (q) for meshes with 5, 10, 20, 40 and 80 points per wavelength (λ/h). We can see that the method is second-order accurate for the second-order discretization. On the sediment interface, the accuracy of the fourth-order scheme is reduced to second-order due to the jump in the material properties. The error on the interface starts to dominate the phase error on finer meshes eventually leading to second-order convergence. Nevertheless, with the fourth-order scheme the error is much smaller for practical mesh step sizes. The accuracy for the 10 points per wavelength with the fourth-order scheme is comparable to the 80 points per wavelength with the second-order scheme. Thus we use eight points per wavelength with the fourth-order scheme for our exterior domain discretization. In Fig. 4 we show the real part of the total field 8 cm (one mesh step for the coarsest grid) above the sediment for the coarsest (five points) and the finest (80 points) grids with the fourth-order discretization. The five points scheme captures the phase of the scattering field reasonably well. Fig. 5 shows the results for 10 and 80 points second-order scheme, and 10 points fourth-order scheme. The second-order solution with 10 points is inaccurate while the two other solutions are fairly accurate and comparable to each other. This clearly demonstrates the effectiveness of the fourth-order discretization.

5.2. Acoustic scattering by a buried elastic target in wavy sediment

Our scattering test problem is similar to the experimental test setup used for measurements in [30]. The scatterer is an 0.5 m long aluminum cylinder with one feet diameter (0.3048 m). The axis of the cylinder is the line segment with endpoints (0 m, 0 m, 0.18 m) and (0 m, 0.5 m, 0.18 m). The computational domain is [−10.5 m, 1.35 m] × [−0.25 m, 0.75 m] × [−0.5 m,

Table 1
The grid refinement analysis for the test problem

N	λ/h	$E_N(p)$	Order(p)	$E_N(q)$	Order(q)
151 × 11 × 61	5	1.6771		6.2599×10^{-1}	
301 × 21 × 121	10	1.8543		4.2801×10^{-2}	3.87
601 × 41 × 241	20	5.4960×10^{-1}	1.75	7.2334×10^{-3}	2.56
1201 × 81 × 481	40	1.3834×10^{-1}	1.99	1.4550×10^{-3}	2.31
2401 × 161 × 961	80	3.4498×10^{-2}	2.00	0	

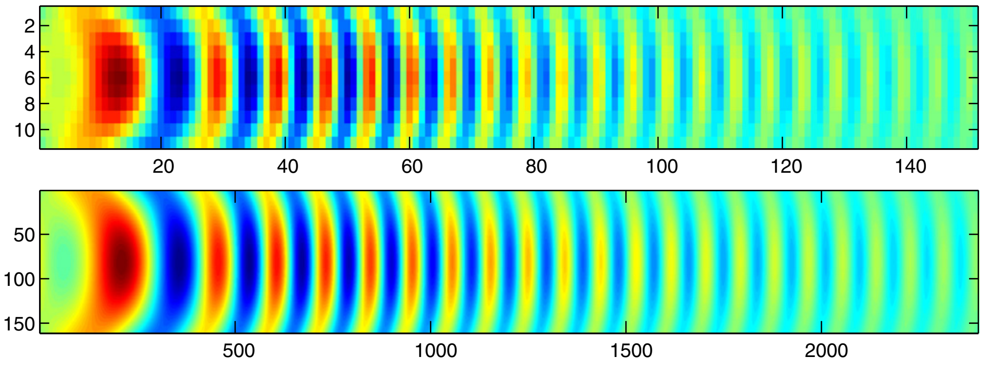

Fig. 4. The total field solutions for the fourth-order discretization on the meshes 151 × 11 × 61 (5 points) and 2401 × 161 × 961 (80 points).

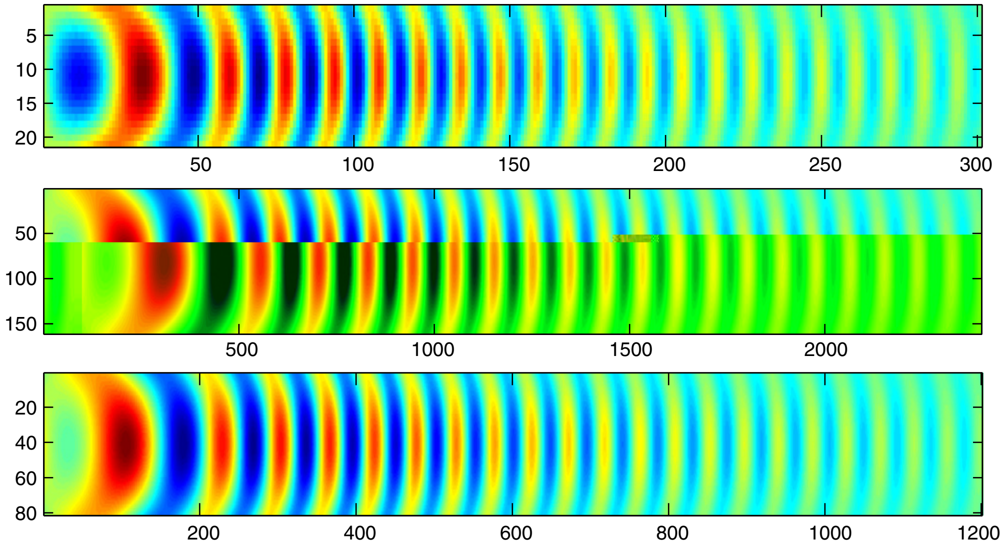

Fig. 5. The total field solutions for the second-order discretization on the meshes 301 × 21 × 121 (10 points) and 2401 × 161 × 961 (80 points), and the fourth-order discretization on the mesh 301 × 21 × 121 (10 points).

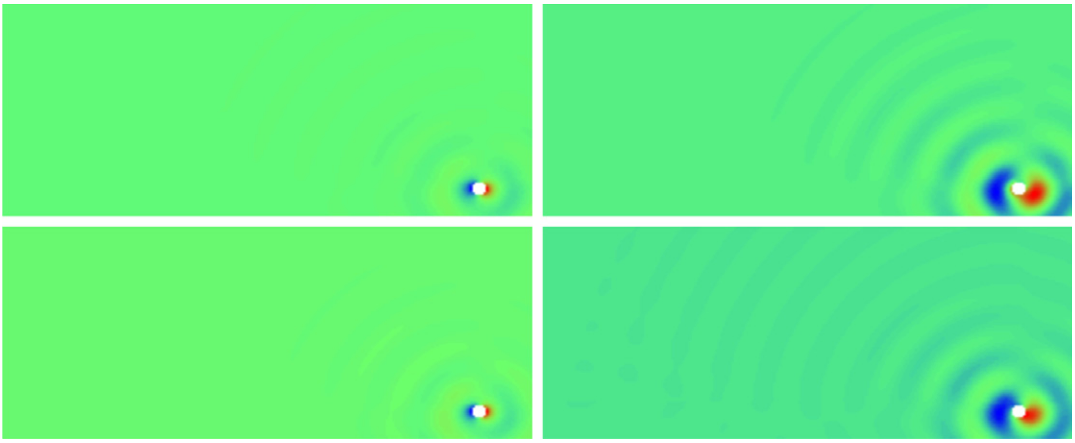

Fig. 6. Real (left) and imaginary (right) part of the scattered field from the COMSOL computation (up) and the coupled code (down).

Table 2
Iteration counts and CPU times for computing the scattering by the aluminum cylinder

f (kHz)	*N*	Iteration	CPU (min)
5	1186 × 101 × 535	17	48
10	1186 × 101 × 535	35	102
15	1186 × 101 × 535	51	199
20	1186 × 101 × 535	62	900
25	1581 × 121 × 713	52	4020

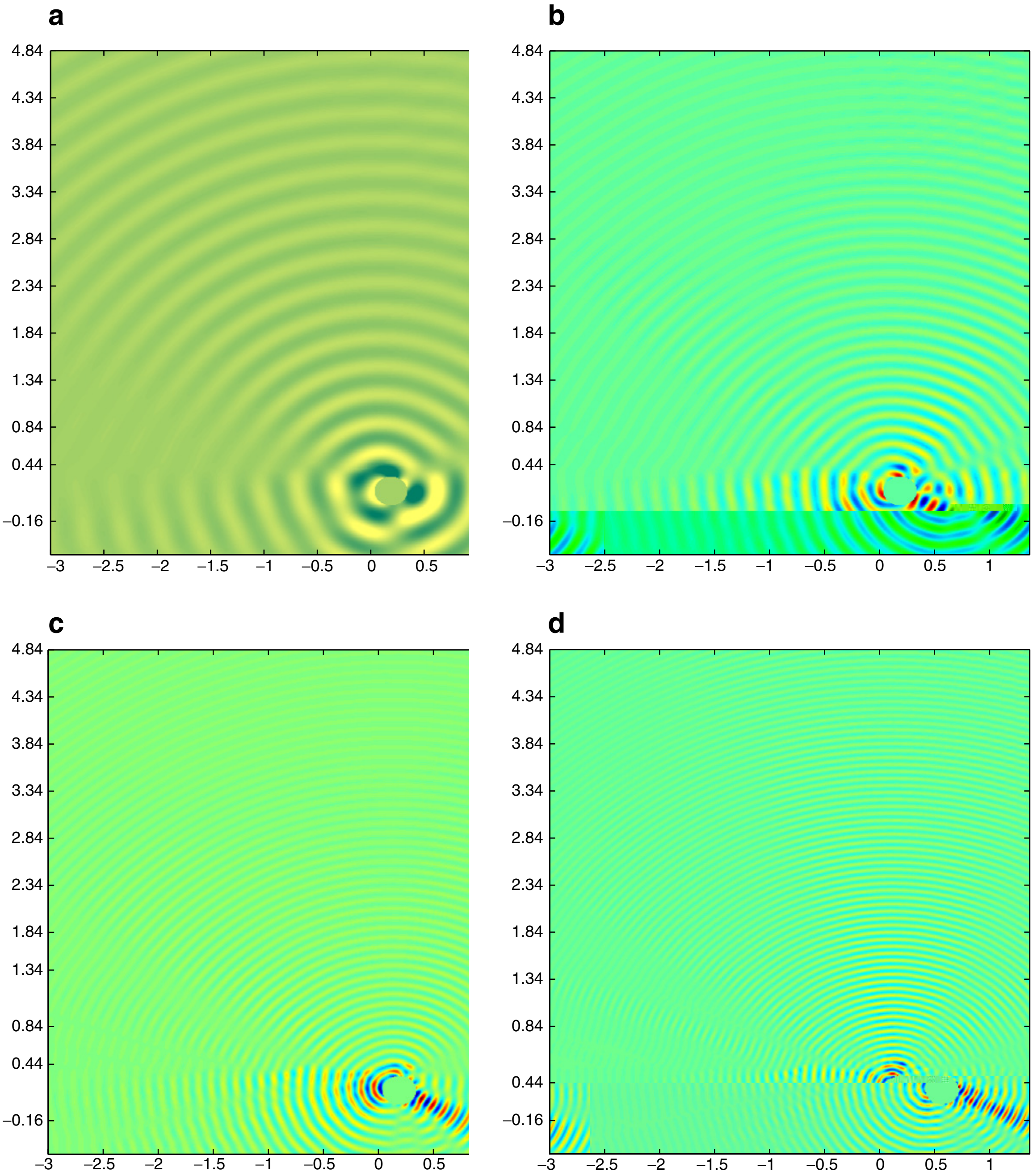

Fig. 7. Scattering by the aluminum cylinder at different frequencies: (a) 5 kHz, (b) 10 kHz, (c) 15 kHz and (d) 20 kHz.

Table 3
Iteration counts and CPU times for the near-field preconditioned GMRES solver

f (kHz)	Iteration	CPU (min)
5	9	1.1
10	14	1.3
15	47	2.0
20	190	14.3
25	300	62.0

Table 4
Iteration counts and CPU times for computing the wave propagation in water and sediment without a target

f (kHz)	N	Iteration	CPU (min)
5	$1186 \times 101 \times 535$	12	16
10	$1186 \times 101 \times 535$	15	19
15	$1186 \times 101 \times 535$	20	25
20	$1186 \times 101 \times 535$	25	30
25	$1581 \times 121 \times 713$	30	89

4.84 m] and the point source is located at [−10 m, 0.25 m, 4.34 m]. The partial rippled interface between the water and sediment is defined by

$$x_3 = f(x_1) = \begin{cases} 0.44\ \text{m} + (0.0381\ \text{m})\cos(2\pi(x_1 - 0.18)/(0.75\ \text{m})), & -2.2575\ \text{m} \leqslant x_1 \leqslant 1.1175\ \text{m} \\ 0.44\ \text{m}, & \text{otherwise}. \end{cases}$$

Thus, the top of the cylinder is 0.108 m below the mean level of the sediment surface. The properties of the water and sediment are the same as in the above section. The density of the cylinder aluminum is 2730 kg/m^3 and its compressional speed and shear speed are c_c = 6568 m/s and c_s = 3149 m/s, respectively. The damping parameter is set as Raleigh damping with the mass damping parameter 1 and the stiffness damping parameter 0.001. The near-field subdomain Ω_1 is [0 m, 0.36 m] × [−0.25 m, 0.75 m] × [0 m, 0.36 m].

We compared the solutions from our coupled far-field and near-field solver and the COMSOL Multiphysics solver for a low frequency 1.5 kHz. The plots of the real and imaginary parts of the solutions in Fig. 6 show the agreement of the solutions to be good. The COMSOL solver required 128 CPU seconds and our coupled solver required 48 CPU seconds. The COMSOL solver cannot solve high frequency problems (>4 kHz) as it either runs out of memory or requires unreasonable amount of time.

For the higher frequencies with required finer discretizations the problem has from 100 million to 1 billion of unknowns. In Table 2, the iteration counts and CPU times for problems with different incident frequencies are given. For the frequencies 5 kHz, 10 kHz, 15 kHz and 20 kHz we use a 1186 × 101 × 535 mesh leading to 30, 15, 10 and 7.5 points per wavelength, respectively. For 25 kHz problem we use 1581 × 121 × 713 mesh having eight points per wavelength. Times in the table are given in minutes on a PC with an Intel Xeon 3.20 GHz processor with 32 GBytes of memory. The GMRES iterations were terminated when the norm of the residual vector was reduced by the factor 10^{-6}. In Fig. 7, the amplitude of the scattered fields at 5–20 kHz are shown.

In order to understand the behavior of the domain decomposition solver better, we performed numerical experiments with the near-field and far-field solvers. The iteration counts and CPU times for the near-field GMRES solver are given in Table 3. The multigrid preconditioner in the near-field solver is reasonably efficient for low frequency problems (5–15 kHz) while for higher frequencies (20–25 kHz) it is inefficient. The convergence of the solver without an elastic target is studied in Table 4. In this case the scattering is only due to the interface between water and sediment, and it is not necessary to solve near-field problems. The number of iterations grows approximately linearly with respect to the frequency. The iteration counts with the target in Table 2 are roughly twice those without a target in Table 4. The CPU time with the target is three times larger at 5 kHz while it is about 45 times larger at 25 kHz. Based on these observations we conclude that the growth of the CPU time is mainly due to the near-field solver for our particular implementation. Tables 2 and 4 also indicate that the GMRES iterations number increases as the frequency increases. It coincides with our discussion in Section 4.1, that is, the conditioning is good for low frequency problems. We are currently working on improving the near-field discretization and solver.

6. Conclusions

We have developed an efficient numerical method for computing time-harmonic acoustic scattering by an elastic object in three-dimensional layered media. The infinite domain is truncated to a rectangular parallelepiped and a second-order absorbing boundary condition is posed on the truncation boundary which usually leads to sufficient accuracy for practical purposes. Our discretization uses a modified trilinear finite element discretization with fourth-order phase accuracy. We studied the accuracy of this discretization in a 30 wavelengths long computational domain with layered media and without a target. In the far-field 10 nodes per wavelength lead to about 4% error while the standard finite element discretization required about 80 nodes per wavelength to reach the same accuracy level.

The discretization of our model problem leads to order of 100 million unknowns. In order to be able to solve these problems on a PC with 32 Gbytes of memory a domain decomposition preconditioner was developed. The domain was decomposed into a small near-field subdomain enclosing the target and a vastly larger far-field subdomain. The preconditioner for the far-field subdomain employs a fast direct solver based on the diagonalization of matrices in tensor product form. This together with reducing the GMRES iterations onto a small sparse subspace associated with the interfaces enabled the solution on the PC. The iterations converged in some tens of iterations with the iteration count growing roughly linearly with the frequency. The near-field subdomain problems were solved iteratively using the COMSOL multiphysics software. The number of these inner iterations seems to grow exponentially with the frequency. Even though the near-field problem was three orders of magnitude smaller it required much more computational effort. Thus, improvements on the near-field solver would increase the efficiency of the solution procedure the most.

Acknowledgments

The authors like to thank Dr. David Burnett, Dr. Quyen Huynh, and Dr. Joseph Lopes for helpful discussions on littoral acoustic scattering problems. The research was supported by the Office of Naval Research Grant N00014-06-1-0067, the National Science Foundation Grant DMS-0610661, and the Academy of Finland Grant #207089.

References

[1] T. Airaksinen, E. Heikkola, A. Pennanen, J. Toivanen, An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation, J. Comput. Phys. 226 (2007) 1196–1210.

[2] A. Bamberger, P. Joly, J.E. Roberts, Second-order absorbing boundary conditions for the wave equation: a solution for the corner problem, SIAM J. Numer. Anal. 27 (1990) 323–352.

[3] A. Banegas, Fast Poisson solvers for problems with sparsity, Math. Comp. 32 (1978) 441–446.

[4] H.T. Banks, K. Ito, G.M. Kepler, J.A. Toivanen, Material surface design to counter electromagnetic interrogation of targets, SIAM J. Appl. Math. 66 (2006) 1027–1049.

[5] H.T. Banks, K. Ito, J.A. Toivanen, Determination of interrogating frequencies to maximize electromagnetic backscatter from objects with material coatings, Commun. Comput. Phys. 1 (2006) 362–382.

[6] C. Börgers, A triangulation algorithm for fast elliptic solvers based on domain imbedding, SIAM J. Numer. Anal. 27 (1990) 1187–1196.

[7] Comsol AB, COMSOL Multiphysics 3.3 Command Reference. Stockholm, 2006.

[8] T.A. Davis, Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method, ACM Trans. Math. Software 30 (2004) 196–199.

[9] H.C. Elman, O.G. Ernst, D.P. O'Leary, A multigrid method enhanced by Krylov subspace iteration for discrete Helmhotz equations, SIAM J. Sci. Comput. 23 (2001) 1291–1315.

[10] Y.A. Erlangga, C.W. Oosterlee, C. Vuik, A novel multigrid based preconditioner for heterogeneous Helmholtz problems, SIAM J. Sci. Comput. 27 (2006) 1471–1492.

[11] G. Fibich, S. Tsynkov, Numerical solution of the nonlinear Helmholtz equation using nonorthogonal expansions, J. Comput. Phys. 210 (2005) 183–224.

[12] Y. Fu, Compact fourth-order finite difference schemes for Helmholtz equation with high wave numbers, J. Comput. Math. 26 (2008) 98–111.

[13] M.N. Guddati, B. Yue, Modified integration rules for reducing dispersion in finite element methods, Comput. Methods Appl. Mech. Eng. 193 (2004) 275–287.

[14] W. Hackbusch, Multigrid methods and applications, in: Springer Series in Computational Mathematics, vol. 4, Springer-Verlag, Berlin, 1985.

[15] E. Heikkola, Y.A. Kuznetsov, P. Neittaanmäki, J. Toivanen, Fictitious domain methods for the numerical solution of two-dimensional scattering problems, J. Comput. Phys. 145 (1998) 89–109.

[16] E. Heikkola, T. Rossi, J. Toivanen, Fast direct solution of the Helmholtz equation with a perfectly matched layer/an absorbing boundary condition, Int. J. Numer. Methods Eng. 57 (2003) 2007–2025.

[17] E. Heikkola, T. Rossi, J. Toivanen, A parallel fictitious domain method for the three-dimensional Helmholtz equation, SIAM J. Sci. Comput. 24 (2003) 1567–1588.

[18] E. Heikkola, T. Rossi, J. Toivanen, A domain embedding method for scattering problems with an absorbing boundary or a perfectly matched layer, J. Comput. Acoust. 11 (2003) 159–174.

[19] Q. Huynh, K. Ito, J. Toivanen, A fast Helmholtz solver for scattering by a sound-soft target in sediment, in: Domain Decomposition Methods in Science and Engineering XVI, Lect. Notes Comput. Sci. Eng., vol. 55, Springer, Berlin, 2007, pp. 595–602.

[20] F. Ihlenburg, Finite element analysis of acoustic scattering, in: Applied Mathematical Sciences, vol. 132, Springer-Verlag, New York, 1998.

[21] K. Ito, J. Toivanen, Efficient domain decomposition method for acoustic scattering in multi-layered media, in: Proceedings of the Eccomas CFD 2006 Conference, Eccomas, Barcelona, 2006.

[22] K. Ito, J. Toivanen, Preconditioned iterative methods on sparse subspaces, Appl. Math. Lett. 19 (2006) 1191–1197.

[23] K. Ito, J. Toivanen, A fast iterative solver for scattering by elastic objects in layered media, Appl. Numer. Math. 57 (2007) 811–820.

[24] Y.A. Kuznetsov, Matrix iterative methods in subspaces, in: Proceedings of the International Congress of Mathematicians, vol. 1, 2 (Warsaw, 1983), Warsaw, 1984, PWN, pp. 1509–1521.

[25] Y.A. Kuznetsov, Numerical methods in subspaces, in: G.I. Marchuk (Ed.), Vychislitel'nye Processy i Sistemy II, Nauka, Moscow, 1985, pp. 265–350 (in Russian).

[26] Y.A. Kuznetsov, K.N. Lipnikov, 3 D Helmholtz wave equation by fictitious domain method, Russ. J. Numer. Anal. Math. Model. 13 (1998) 371–387.

[27] Y.A. Kuznetsov, A.M. Matsokin, Partial solution of systems of linear algebraic equations, in: Numerical methods in applied mathematics (Paris, 1978), "Nauka" Sibirsk. Otdel., Novosibirsk, 1982, pp. 143–163.

[28] E. Larsson, A domain decomposition method for the Helmholtz equation in a multilayer domain, SIAM J. Sci. Comput. 20 (1999) 1713–1731.

[29] Z. Li, K. Ito, The immersed interface method-numerical solution of PDEs involving interfaces and irregular domains, SIAM, Philadelphia, 2006.

[30] C.L. Nesbitt, J.L. Lopes, Subcritical detection of an elongated target buried under a rippled interface, in: Proceedings of Oceans'04, vol. 4, IEEE, 2004, pp. 1945–1952.

[31] R.E. Plessix, W.A. Mulder, Separation-of-variables as a preconditioner for an iterative Helmholtz solver, Appl. Numer. Math. 44 (2003) 385–400.

[32] T. Rossi, J. Toivanen, A nonstandard cyclic reduction method, its variants and stability, SIAM J. Matrix Anal. Appl. 20 (1999) 628–645.

[33] T. Rossi, J. Toivanen, A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary dimension, SIAM J. Sci. Comput. 20 (1999) 1778–1796.

[34] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.

[35] B.F. Smith, P.E. Bjørstad, W.D. Gropp, Domain Decomposition, Cambridge University Press, Cambridge, 1996.

[36] P.S. Vassilevski, Fast algorithm for solving a linear algebraic problem with separable variables, C.R. Acad. Bulg. Sci. 37 (1984) 305–308.