

A PARALLEL DOMAIN DECOMPOSITION METHOD FOR THE
HELMHOLTZ EQUATION IN LAYERED MEDIA*ERKKI HEIKKOLA[†], KAZUFUMI ITO[‡], AND JARI TOIVANEN[§]

Abstract. An efficient domain decomposition method and its parallel implementation for the solution of the Helmholtz equation in three-dimensional layered media are considered. A modified trilinear finite element discretization scheme is applied to the equation system leading to fourth-order phase accuracy and thereby reducing the pollution error considerably. The resulting linear system is solved with the GMRES method using a multiplicative nonoverlapping domain decomposition preconditioner with layers defining the subdomains. This right preconditioner is constructed by embedding each layer into a rectangular domain and by employing a fast direct solver. Due to the construction of the preconditioner the iterations can be reduced to a subspace corresponding to the interfaces between the layers. Numerical experiments with several test cases demonstrate the effectiveness and scalability of the proposed method and ability to solve large-scale problems with up to billions of unknowns.

Key words. Helmholtz equation, domain decomposition method, preconditioned iterative method, ultrasonic tomography, geological survey

AMS subject classifications. 65N22, 65F08

DOI. 10.1137/18M1230906

1. Introduction. The modeling of acoustic waves in medical acoustics or seismic imaging requires the solution of the Helmholtz equation in heterogeneous media. Potential application examples for an efficient Helmholtz solver with variable parameters are ultrasound waveform tomography and the full waveform inversion procedure, which involve inverse problems related to the material parameters in the Helmholtz equation [21, 27, 31]. The computationally most demanding component of such procedures is a solver for the forward problem.

This article considers an efficient numerical method to solve the Helmholtz equation in layered media with piecewise constant speed of sound. Numerous formulations and numerical methods have been introduced for the solution of the Helmholtz equation during the past decades, and some of the recent approaches are reviewed, for example, in [6, 33]. Our method is based on a domain decomposition preconditioner for the iterative solution of the discrete equations arising from finite element discretization.

The finite element discretization of the Helmholtz equation leads to indefinite, complex-valued, and large-scale linear system of equations, which is challenging to solve [6]. The number of nodes per wavelength needs to be sufficiently high to achieve reasonable accuracy, and the accuracy is deteriorated with increasing frequency by the pollution effect [13]. The linear systems to be solved in practical applications often involve billions of unknowns the solution of which requires parallelization of

*Submitted to the journal's Software and High-Performance Computing section December 19, 2018; accepted for publication (in revised form) July 10, 2019; published electronically October 1, 2019.

<https://doi.org/10.1137/18M1230906>

Funding: This work was supported by the Academy of Finland under project 295897.

[†]Numerola Oy, University of Jyväskylä, 40101 Jyväskylä, Finland (erkki.heikkola@numerola.fi).

[‡]Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (kito@math.ncsu.edu).

[§]Faculty of Information Technology, University of Jyväskylä, 40101 Jyväskylä, Finland (jari.a.toivanen@jyu.fi).

algorithms and massively parallel computers [30]. Furthermore, standard iterative methods for Helmholtz problems converge slowly without a good preconditioner. Therefore, the simulation of high-frequency waves is often limited by computational resources.

Research work on efficient solvers for the Helmholtz equation with variable coefficients has concentrated on various preconditioning techniques, which are often based on domain decomposition methods, multigrid methods, and fast direct solvers. By fast direct methods, we mean direct solution methods based on FFT or cyclic reduction, for example, which require order of $m \log m$ or $m(\log m)^2$ operations to solve a system with m unknowns. For Helmholtz problems in layered media such techniques have been studied by Plessix and Mulder in [22]. They applied tensor product form preconditioners with fast direct solvers and obtained good efficiency for low-frequency problems. However, the performance of the method deteriorated with increasing frequency. Ito and Toivanen used a similar approach successfully for a problem with almost perfectly layered media [17]. Larsson and Holmgren introduced a domain decomposition preconditioner employing a fast direct solver and demonstrated some parallel computations [20]. An effective multigrid preconditioner for geological survey problems was developed by Erlangga, Oosterlee, and Vuik [7]. More recently, Engquist and Ying introduced a new type of preconditioner based on perfectly matched layers, named the sweeping preconditioner [5]. Similar methods have been proposed for the Helmholtz equation also by Stolk [29] as well as Eslamnia and Guddati [8]. Also the polarized traces method [32] by Zepeda-Núñez and Demanet belongs to this class of methods. These new preconditioning techniques based on domain decomposition for the Helmholtz equation have been reviewed in the article [10].

In this work, we introduce a multiplicative nonoverlapping domain decomposition preconditioner for the iterative solution of the Helmholtz equation in layered media with layers defining the subdomains. This right preconditioner and method was originally introduced for two-dimensional problems in [15]. Here we extend the same idea to three-dimensional problems and develop a parallel implementation of the method. Furthermore, the finite element discretization of the equation is here performed with modified trilinear elements leading to fourth-order phase accuracy and thereby reducing the pollution error considerably [11].

Each layer is embedded into a rectangular domain, and the preconditioner is based on employing a fast direct solver separately in each rectangle. With constant wave number in each layer, the preconditioner can be constructed such that all rows of the system matrix and the preconditioner corresponding to the interior of the layer coincide. Due to this fact, the iterative solution with the GMRES method [26] can be reduced to a subspace corresponding to the layer interfaces and their neighboring grid points. This essentially reduces the memory consumption. The solution procedure can be considered as a preconditioned iterative method in a subspace [16, 18]. Numerical tests in a parallel environment demonstrate the effectiveness of the method. The number of required GMRES iterations depends fairly weakly on the frequency with logarithmic growth with respect to the frequency in the most challenging case.

The paper is organized as follows. The mathematical problem is formulated in section 2. Construction of the discrete system of equations with the modified quadrature rule is described in section 3. The domain decomposition preconditioner for the discrete equations is considered in section 4 with the subdomain preconditioners introduced in subsection 4.2. Implementation and parallelization of the iterative solution procedure in a subspace is described in section 5. The parallel performance of the method is studied with three test cases from medical physics and geophysics.

Results of the numerical experiments are reported in section 6. The conclusions follow in section 7.

2. Mathematical formulation. We consider the propagation of time-harmonic acoustic waves in unbounded and layered three-dimensional fluid media. The unbounded fluid domain is confined into a rectangular domain Π . A schematic two-dimensional illustration of the modeled fluid domain consisting of four horizontal layers is given in Figure 2.1. The number of layers, denoted by L , is not limited and the interfaces between the layers Ω_j are not restricted to be planar:

$$(2.1) \quad \Pi = \bigcup_{j=1}^L \bar{\Omega}_j.$$

In each layer Ω_j , the pressure variations p satisfy the Helmholtz equation

$$(2.2) \quad -\nabla \cdot \left(\frac{1}{\rho_j} \nabla p \right) - \frac{k_j^2}{\rho_j} p = g \quad \text{in } \Omega_j,$$

where $k_j = \frac{\omega}{c_j}$ is the wave number, ω the angular frequency, and c_j and ρ_j the constant speed of sound and density in layer j . The function g corresponds to the sound source. On the truncation boundaries $\partial\Pi$ we impose the second-order absorbing boundary condition introduced in [1] to minimize spurious reflections. Mathematical analysis on the existence, uniqueness, stability, and regularity of the solutions to Helmholtz problems of type (2.2) is given in [3] in the case with the first-order absorbing boundary condition.

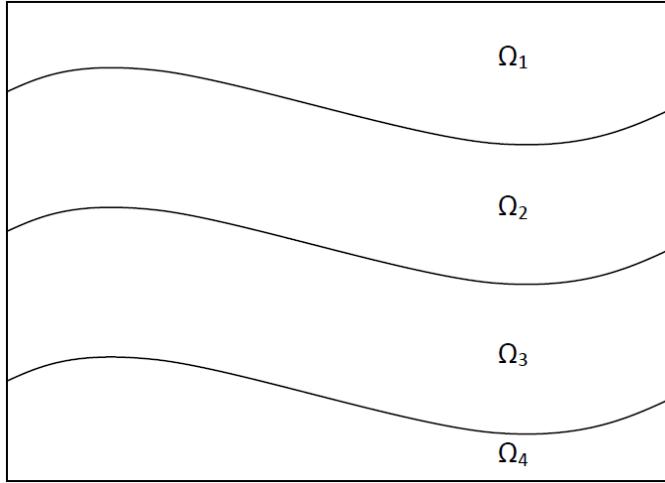


FIG. 2.1. Illustration of a layered domain with four layers.

In the three-dimensional case, the boundary $\partial\Pi$ consists of six rectangular faces denoted by $\Gamma_{\pm j}$, $j = 1, 2, 3$, whose outward normal directions are the coordinate directions $\pm x_j$, respectively. Then, the second-order boundary conditions on the faces are of the form

$$(2.3) \quad \pm \frac{1}{\rho} \frac{\partial p}{\partial x_j} - \frac{ikp}{\rho} - \frac{i}{2k\rho} \sum_{1 \leq l \neq j \leq 3} \frac{\partial^2 p}{\partial x_l^2} = 0.$$

The condition on the edge between the faces $\Gamma_{\pm m}$ and $\Gamma_{\pm l}$, denoted by $\Gamma_{(\pm m, \pm l)}$, is given by

$$(2.4) \quad -\frac{3k^2 p}{2\rho} - \frac{ik}{\rho} \left(\pm \frac{\partial p}{\partial x_m} \pm \frac{\partial p}{\partial x_l} \right) - \frac{1}{2\rho} \frac{\partial^2 p}{\partial x_j^2} = 0, \quad j \neq m, l,$$

while the conditions on the eight corners are

$$(2.5) \quad -\frac{2ikp}{\rho} + \sum_{l=1}^3 \pm \frac{1}{\rho} \frac{\partial p}{\partial x_l} = 0.$$

The sets of edges and corners of $\partial\Pi$ are denoted by Φ and Ψ .

For the weak formulation of the equations (2.2) with the boundary conditions (2.3), (2.4), and (2.5) we introduce the space V :

$$(2.6) \quad V = \{v \in H^1(\Pi) : v|_{\partial\Pi} \in H^1(\partial\Pi), v|_{\Gamma_{(m,j)}} \in H^1(\Gamma_{(m,j)}) \forall \Gamma_{(m,j)} \in \Phi\}.$$

Then the weak formulation is given as follows: Find $p \in V$ such that

$$(2.7) \quad \begin{aligned} & \int_{\Pi} \left(\frac{1}{\rho} \nabla p \cdot \nabla q - \frac{k^2}{\rho} pq \right) dx - i \frac{k}{\rho} \int_{\partial\Pi} pq ds \\ & + \frac{i}{2k\rho} \sum_{j=1}^3 \sum_{l \neq j} \left(\int_{\Gamma_j} \frac{\partial p}{\partial x_l} \frac{\partial q}{\partial x_l} ds + \int_{\Gamma_{-j}} \frac{\partial p}{\partial x_l} \frac{\partial q}{\partial x_l} ds \right) \\ & + \frac{3}{4\rho} \sum_{\Gamma_{(m,j)} \in \Phi} \int_{\Gamma_{(m,j)}} pq dl - \frac{1}{4k^2\rho} \sum_{\Gamma_{(m,j)} \in \Phi} \int_{\Gamma_{(m,j)}} \frac{\partial p}{\partial x_n} \frac{\partial q}{\partial x_n} \Big|_{n \neq m, j} dl \\ & + \frac{i}{2k\rho} \sum_{x \in \Psi} p(x)q(x) = \int_{\Pi} gq dx \end{aligned}$$

for all $q \in V$.

3. Finite element discretization. The finite element discretization of (2.7) is performed with hexahedral elements and trilinear basis functions. A uniform hexahedral mesh with $n_1 \times n_2 \times n_3$ nodes and the grid step size h is first generated into the domain Π . The mesh nodes are not shifted to match the layer interfaces, and each element is assigned to a certain layer depending on the location of the center point of the element. Therefore, the discretization gives a staircase approximation to the layer interfaces as illustrated in Figure 3.1.

More precisely, let us denote the set of element midpoints by M . Then, the discretized subdomains $\hat{\Omega}_j$ are defined by

$$(3.1) \quad \hat{\Omega}_j = \bigcup_{\substack{c(x_1, x_2, x_3) = c_j \\ (x_1, x_2, x_3) \in M}} \prod_{i=1}^3 \left[x_i - \frac{h}{2}, x_i + \frac{h}{2} \right].$$

The linear system of equations arising from the finite element discretization of (2.7) can be assembled from the elementwise stiffness and mass matrices. Usually the integrations required to compute these matrices are transformed into a simple reference element which for hexahedral elements is $[-1, 1] \times [-1, 1] \times [-1, 1]$. We denote by $\hat{N}(\xi) \in \mathbb{R}^8$ the vector-valued function giving the values of the Lagrangian

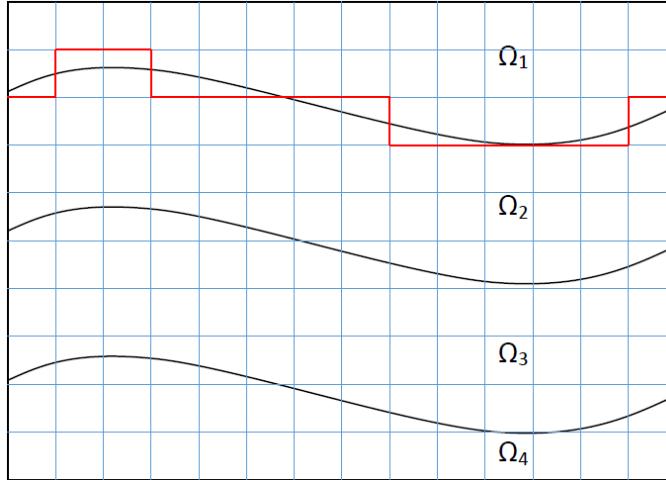


FIG. 3.1. An orthogonal grid in domain Π for the two-dimensional example case. Approximation of the interface between the layers Ω_1 and Ω_2 is shown with a red line.

basis functions associated with the corners of the reference element. The mapping from the reference element to the actual element Ω_e is denoted by $x(\xi)$ and its Jacobian by $J(\xi)$, where ξ is a point in the reference element. With this notation, the local mass and stiffness matrices of Ω_e can be represented in the form

$$(3.2) \quad M_e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \hat{N}(\xi)^T \hat{N}(\xi) \det J(\xi) d\xi_1 d\xi_2 d\xi_3,$$

$$K_e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \hat{G}(\xi)^T \hat{G}(\xi) \det J(\xi) d\xi_1 d\xi_2 d\xi_3,$$

where $G(\xi) = (J(\xi)^T)^{-1} \nabla \hat{N}(\xi)$.

Typically the matrices in (3.2) are evaluated numerically using the Gauss quadrature

$$(3.3) \quad \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \psi(\xi) d\xi_1 d\xi_2 d\xi_3 = \sum_{j=1}^{n_q} W_j \psi(\xi^j),$$

where n_q is the number of Gauss points, W_j are the weight coefficients, and ξ^j are the local coordinates of Gauss points. The most commonly used choices for trilinear elements are the Gauss rule ($n_q = 8$, $W_j = 1$, $\xi^j = (\pm 1/\sqrt{3}, \pm 1/\sqrt{3}, \pm 1/\sqrt{3})$) and the Gauss–Lobatto rule ($n_q = 8$, $W_j = 1$, $\xi^j = (\pm 1, \pm 1, \pm 1)$), which both provide similar accuracy with respect to the mesh size h . A major drawback of the traditional discretization is that it leads to a dispersion error, which can accumulate significantly if the domain is large in terms of wavelengths. This dispersion error is also called the pollution error [13].

Guddati and Yue studied a generalized integration rule of the form (3.3) with weights $W_j = 1$ and points $\xi^j = (\pm \alpha, \pm \alpha, \pm \alpha)$ [11]. They observed that with the integration points $\alpha = \sqrt{2}/3$ the dispersion error is reduced by an order of magnitude. With the conventional integration rules the discrete wavelength is second-order accurate with respect to the number of nodes per wavelength, but with the modified

rule it becomes fourth-order accurate. This has been shown using a Taylor expansion with respect to the wave number in [11]. This discretization technique has previously been applied to Helmholtz problems in [17, 14, 8]. Alternatively dispersion minimizing schemes described in [28] can be used after expressing them in the tensor product form considered in subsection 4.2

4. Construction of the preconditioner. We obtain the following system of linear equations from the finite element discretization:

$$(4.1) \quad Ax = b.$$

For the iterative solution of this system we introduce a right preconditioner matrix B and solve iteratively the system

$$(4.2) \quad AB^{-1}v = b$$

instead of the original system (4.1). We apply the GMRES method as the iterative solution procedure [26]. After obtaining v the solution x of the original system is given by $Bx = v$.

4.1. Domain decomposition. In order to define the matrix B , we denote by R_j the restriction operator into the subdomain $\hat{\Omega}_j$. Multiplication of a vector by R_j results in a vector containing only the components associated with the subdomain $\hat{\Omega}_j$ and its boundary. The subdomain preconditioners B_j for each subdomain Ω_j are defined later in subsection 4.2. Then, the matrix B is a multiplicative domain decomposition preconditioner given by the recursive formula:

$$(4.3) \quad \begin{aligned} P_1^{-1} &= R_1^T B_1^{-1} R_1, \\ P_j^{-1} &= P_{j-1}^{-1} + R_j^T B_j^{-1} R_j (I - AP_{j-1}^{-1}), \quad j = 2, \dots, L, \\ B &= P_L. \end{aligned}$$

Multiplication of a vector x by the inverse of the preconditioner B can thus be performed by the following sequence, which goes through the L layers of the domain:

$$(4.4) \quad \begin{aligned} y^{(1)} &= R_1^T B_1^{-1} R_1 x, \\ y^{(2)} &= y^{(1)} - R_2^T B_2^{-1} R_2 (x - Ay^{(1)}), \\ &\vdots \\ y^{(L)} &= y^{(L-1)} - R_L^T B_L^{-1} R_L (x - Ay^{(L-1)}) = B^{-1}x. \end{aligned}$$

4.2. Subdomain preconditioners. The subdomain preconditioners are based on embedding each layer into a rectangular domain and on employing a fast direct solver separately in each rectangle. We first define for each subdomain $\hat{\Omega}_j$ the minimum and maximum coordinate values:

$$(4.5) \quad x_{i,\min}^j = \underset{(x_1, x_2, x_3) \in \hat{\Omega}_j}{\operatorname{argmin}} x_i, \quad x_{i,\max}^j = \underset{(x_1, x_2, x_3) \in \hat{\Omega}_j}{\operatorname{argmax}} x_i, \quad i = 1, 2, 3.$$

Then, the values

$$(4.6) \quad x_{i,\min} = \min_{j=1, \dots, L} x_{i,\min}^j, \quad x_{i,\max} = \max_{j=1, \dots, L} x_{i,\max}^j, \quad i = 1, 2, 3,$$

give similar bounds for the domain Π , and we can define rectangular extensions of $\hat{\Omega}_j$ by

$$(4.7) \quad \Theta_j = \prod_{i=1}^3 \left[\max\{x_{i,\min}^j - Q h, x_{i,\min}\}, \min\{x_{i,\max}^j + Q h, x_{i,\max}\} \right],$$

where Q is a nonnegative integer. The construction of these extended domains for the two-dimensional example case are illustrated in Figure 4.1, which shows the domains Θ_j highlighted in red. We denote the number of nodes in the extended domain Θ_j in the coordinate directions x_1 , x_2 , and x_3 by n_1^j , n_2^j , and n_3^j , respectively.

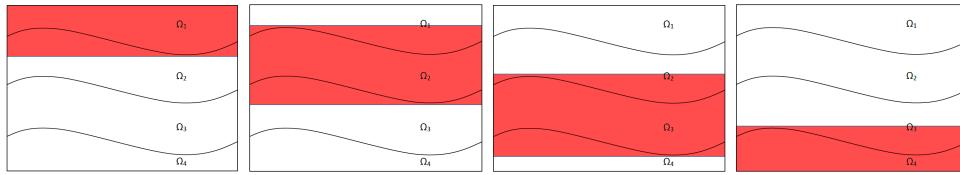


FIG. 4.1. The extended domains Θ_j for the four subdomains Ω_j in the two-dimensional example case.

The subdomain preconditioner B_j is based on the Helmholtz equation (2.2) in the extended domain Θ_j with the second-order absorbing boundary conditions (2.3), (2.4), and (2.5) on the boundaries $\partial\Theta_j$. We discretize this equation using the same orthogonal mesh as described in section 3, which leads to a matrix of the block form

$$(4.8) \quad C_j = \begin{pmatrix} C_{j,dd} & C_{j,de} \\ C_{j,ed} & C_{j,ee} \end{pmatrix}$$

with the degrees of freedom corresponding to the subdomain $\hat{\Omega}_j$ numbered first (subscript d) and the degrees of freedom corresponding to the extension $\Theta_j \setminus \hat{\Omega}_j$ numbered second (subscript e). The preconditioner B_j is now defined as the Schur complement matrix

$$(4.9) \quad B_j = C_{j,dd} - C_{j,de} C_{j,ee}^{-1} C_{j,ed}.$$

Solutions to linear systems of the form $B_j y_j = x_j$ can be obtained as the first block of the solution to larger systems

$$(4.10) \quad C_j y = \begin{pmatrix} C_{j,dd} & C_{j,de} \\ C_{j,ed} & C_{j,ee} \end{pmatrix} \begin{pmatrix} y_j \\ y_e \end{pmatrix} = \begin{pmatrix} x_j \\ 0 \end{pmatrix}.$$

By renumbering the degrees of freedom appropriately the block tridiagonal matrix C_j can be represented in the tensor product form

$$(4.11) \quad C_j = A_1^j \otimes M_2^j \otimes M_3^j + M_1^j \otimes A_2^j \otimes M_3^j + M_1^j \otimes M_2^j \otimes A_3^j - \frac{k_j^2}{\rho_j} M_1^j \otimes M_2^j \otimes M_3^j.$$

Linear systems with such matrices can be solved efficiently with fast direct methods such as the cyclic reduction method [12, 25]. The tridiagonal $n_i^j \times n_i^j$ matrices A_i^j and M_i^j , $i = 1, 2, 3$, are the stiffness and mass matrices for one-dimensional problems in the three coordinate directions with special terms on the boundaries due to the absorbing

boundary condition. We note that the computational efficiency of the subdomain preconditioner B_j deteriorates when the extended domain Θ_j is large compared to the subdomain $\hat{\Omega}_j$. For example, elongated submains can lead to this kind of case.

By applying the general Gauss quadrature rule (3.3) with the integration points $\xi^j = (\pm\alpha, \pm\alpha, \pm\alpha)$ the one-dimensional stiffness and mass matrices are of the form

$$(4.12) \quad A_l^j = \frac{1}{h\rho_j} \begin{pmatrix} 1 - i\frac{hk_j}{2} & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 & -1 \\ & & & -1 & 1 - i\frac{hk_j}{2} \end{pmatrix}$$

and

$$(4.13) \quad M_l^j = \frac{h(1 + \alpha^2)}{2} \begin{pmatrix} d & q & & & \\ q & 1 & q & & \\ & \ddots & \ddots & \ddots & \\ & & q & 1 & q \\ & & & q & d \end{pmatrix},$$

where $d = \frac{1}{2} + \frac{i}{hk_j(1+\alpha^2)}$ and $q = \frac{1}{2} \frac{1-\alpha^2}{1+\alpha^2}$. The quadrature rule leading to reduced dispersion error is achieved by $\alpha = \sqrt{2/3}$.

5. Iterative solution method.

5.1. Iterations on a subspace. The GMRES method for the preconditioned system (4.2) approximates the solution v by minimizing the norm of the residual with respect to a sequence of Krylov subspaces [26]. On the j th iteration the approximation v^j minimizes the norm of $AB^{-1}v^j - b$ in the Krylov subspace

$$(5.1) \quad \text{span}\{b, AB^{-1}b, (AB^{-1})^2b, \dots, (AB^{-1})^{j-1}b\}.$$

We can see by induction that the vectors in the Krylov subspace belong to $X = \text{span}\{b\} + \text{range}(A - B)$ [15, 17].

Due to the construction of the preconditioner most of the rows in matrices A and B coincide. Only the rows corresponding to the interfaces between the layers and their neighboring grid points are different from each other. In the three-dimensional case, the number of such rows is of order $m^{2/3}$ with $m = n_1 \times n_2 \times n_3$ being the dimension of the linear system (4.1). Furthermore, we assume that the vector b has at most $\mathcal{O}(m^{2/3})$ nonzero entries, which allows us to conclude that dimension of the subspace X is $\mathcal{O}(m^{2/3})$. This observation can be exploited in the implementation of the GMRES iterations to reduce the memory usage and computational work significantly. More precisely, the memory requirement is reduced by the factor $\mathcal{O}(m^{1/3})$ while the computational cost of matrix-vector operations are reduced by the factor $\mathcal{O}(jm^{1/3})$, where j is the number of GMRES iterations.

5.2. Partial solution technique. During the iterative solution it is necessary to solve linear systems with the subdomain preconditioners B_j , which require the solution of larger systems of the form (4.10). The dimension of these systems is $\mathcal{O}(m)$. The right-hand-side x_j is a sparse vector with nonzero components only near

the subdomain interfaces and only the corresponding components are required from the solution y_j . The solution of such systems with matrices of tensor product form can be performed with a specialized procedure called the partial solution algorithm [2, 19]. It allows us to solve the linear systems (4.10) in $\mathcal{O}(m \log m)$ arithmetical operations as opposed to $\mathcal{O}(m \log^2 m)$ required by fast direct methods which do not exploit the sparsity structure. The computational cost of this method is discussed in more detail after presenting it in the following. The method can be implemented such that the memory consumption is only $\mathcal{O}(m^{2/3})$. Note that storing the matrix C_j defining B_j in (4.9) as a generic sparse matrix would require $\mathcal{O}(m)$ floating point numbers. For comparison forming the factorization with the nested dissection method for C_j would require $O(m^{4/3})$ floating point numbers and $O(m^2)$ arithmetical operations [4]. Thus, the partial solution algorithm offers huge memory and computational saving for these problems having very specific structure.

The partial solution technique requires the diagonalization of the matrices A_1^j and M_1^j in the tensor product form (4.11). This is achieved by solving the generalized eigenvalue problem

$$(5.2) \quad A_1^j w_l = \lambda_l M_1^j w_l,$$

where (λ_l, w_l) is the l th eigenpair and the vectors w_l satisfy the condition

$$(5.3) \quad w_l^T M_1^j w_k = \delta_{lk}, \quad l, k = 1, \dots, n_1^j.$$

The properties of this eigenvalue problem have been studied in [9] and a solution procedure for this problem is described in [12]. The computational cost of this procedure is $\mathcal{O}((n_1^j)^2) = \mathcal{O}(m^{2/3})$.

If we now define the matrices

$$(5.4) \quad \begin{aligned} \Lambda &= \text{diag}\{\lambda_1, \dots, \lambda_{n_1^j}\}, \\ W &= [w_1 \dots w_{n_1^j}] \end{aligned}$$

and denote the $n_i^j \times n_i^j$ identity matrix by I_i^j we can transform the tensor product matrices C_j into a block-diagonal form

$$(5.5) \quad \begin{aligned} \hat{C}_j &= (W^T \otimes I_2^j \otimes I_3^j) C_j (W \otimes I_2^j \otimes I_3^j) \\ &= \Lambda \otimes M_2^j \otimes M_3^j + I_1^j \otimes A_2^j \otimes M_3^j + I_1^j \otimes M_2^j \otimes A_3^j - \frac{k_j^2}{\rho_j} I_1^j \otimes M_2^j \otimes M_3^j. \end{aligned}$$

Linear systems of the form $C_j y = x$ can then be solved using Algorithm 5.1. We denote a diagonal matrix with diagonal components corresponding to the solution subspace equal to 1 and all other components equal to 0 by S_j and we define the $n_2^j n_3^j \times n_2^j n_3^j$ block tridiagonal matrices

$$(5.6) \quad \hat{C}_j^l = \lambda_l M_2^j \otimes M_3^j + A_2^j \otimes M_3^j + M_2^j \otimes A_3^j - \frac{k_j^2}{\rho_j} M_2^j \otimes M_3^j, \quad l = 1, \dots, n_1^j.$$

Computing one \hat{x} in Algorithm 5.1 requires $\mathcal{O}(m^{2/3})$ operations as x has $\mathcal{O}(m^{2/3})$ nonzero entries. For solving the two-dimensional problem with \hat{C}_j^l the fast direct method called the partial solution variant of the cyclic reduction method can be used. It has been considered in [12, 25] and it requires $\mathcal{O}(n_2^j n_3^j \log n_2^j) = \mathcal{O}(m^{2/3} \log m)$

Algorithm 5.1. Partial solution method.

```

Set  $y = 0$ 
for  $l = 1, \dots, n_1^j$  do
   $\hat{x} = (w_l^T \otimes I_2^j \otimes I_3^j)x$ 
   $\hat{C}_j^l \hat{u} = \hat{x}$ 
   $y = y + S_j(w_l \otimes I_2^j \otimes I_3^j)\hat{u}$ 
end for

```

operations. As the diagonal matrix S_j has $\mathcal{O}(m^{2/3})$ nonzero entries computing the term $S_j(w_l \otimes I_2^j \otimes I_3^j)\hat{u}$ requires $\mathcal{O}(m^{2/3})$ operations. As all the above steps are performed $n_1^j = \mathcal{O}(m^{1/3})$ times the computational cost of Algorithm 5.1 is $\mathcal{O}(m \log m)$ operations.

5.3. Parallel implementation. The parallel implementation of the preconditioned GMRES method is performed using the MPI library. It is based on distributing the following matrices and vectors equally among the processors:

- columns of the finite element matrix A in (4.1),
- components of the basis vectors of the Krylov subspace (5.1),
- eigenvectors w_m of (5.2).

It is then straightforward to distribute the matrix-vector operations related to the matrix A and Krylov subspace vectors among the processors. The partial solution algorithm Algorithm 5.1 is parallelized according to the distribution of the eigenvectors w_m . Each processor computes the part of y corresponding to its eigenvectors, and the global solution is finally distributed to all processors by an allreduce-operation.

6. Experimental results.

6.1. Test cases. We define three schematic test cases, which correspond to applications in geophysics and ultrasonic imaging. The first test case involves four planar layers of varying thickness corresponding to the characterization of marine sediments [23]. The second test case is a three-dimensional wedge problem with three layers. A similar test case has been considered also in [24]. The third test case imitates a typical setting in the ultrasonic tomography of breast tissue [21, 27]. The numerical experiments with the three test cases were performed in the Sisu supercomputer managed by the CSC-IT Center for Science. Sisu is a massively parallel processor supercomputer produced by Cray, Inc., belonging to the XC40 family. In each test case, the GMRES iterations for the preconditioned system (4.2) are terminated when the norm of the residual is reduced by a factor of 10^{-6} .

In the first test case the rectangular domain Π is given by

$$(6.1) \quad \Pi = [0, 20] \text{ m} \times [0, 20] \text{ m} \times [0, 20] \text{ m},$$

and it is divided into four layers along the x_3 -coordinate. A cross section of the domain along the plane $x_2 = 10$ is shown on the left in Figure 6.1. The thicknesses of the layers is $h_1 = 8$ m and $h_2 = h_3 = h_4 = 4$ m. Material parameters in the four layers are given by

$$(6.2) \quad \begin{aligned} c_1 &= 1500 \text{ m/s}, \rho_1 = 1000 \text{ kg/m}^3, \\ c_2 &= 1600 \text{ m/s}, \rho_2 = 1300 \text{ kg/m}^3, \\ c_3 &= 1725 \text{ m/s}, \rho_3 = 1800 \text{ kg/m}^3, \\ c_4 &= 2200 \text{ m/s}, \rho_4 = 2000 \text{ kg/m}^3, \end{aligned}$$

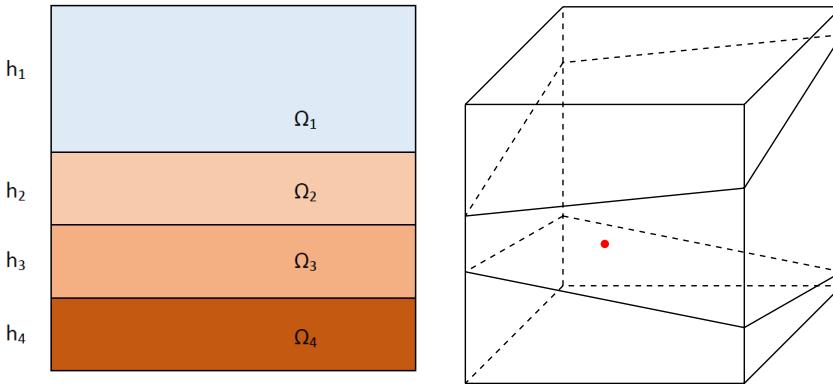


FIG. 6.1. On the left: Cross section of the first test domain with four planar layers. On the right: Illustration of the three-dimensional wedge geometry with the source point indicated by a red dot.

and the frequencies of the considered time-harmonic problem are 2.5 and 5 kHz. The source term g of the equation is defined by a point source located at $x_1 = x_2 = 10$, $x_3 = 0$.

In the second test case the rectangular domain Π is given by $\Pi = [0, 1000]^3 \text{ m}^3$ and the three layers are separated by the two planes

$$(6.3) \quad \begin{aligned} x_3 &= 0.1x_1 + 0.2x_2 + 0.6, \\ x_3 &= -0.2x_1 - 0.15x_2 + 0.4. \end{aligned}$$

Material parameters in the three layers are given by

$$(6.4) \quad \begin{aligned} c_1 &= 833 \text{ m/s}, \rho_1 = 900 \text{ kg/m}^3, \\ c_2 &= 1000 \text{ m/s}, \rho_2 = 1000 \text{ kg/m}^3, \\ c_3 &= 500 \text{ m/s}, \rho_3 = 1100 \text{ kg/m}^3, \end{aligned}$$

and the frequency of the considered time-harmonic problems varies between 10 and 80 kHz. There is a point source at $x_1 = x_3 = 0.5$, $x_2 = 0$.

In the third test case, domain Π is given by $\Pi = [0, 0.1]^3 \text{ m}^3$, and there are five separate layers as illustrated in Figure 6.2. The planar bottom layer Ω_5 corresponds to muscle tissue and its thickness is 1 cm. The domain Ω_4 represents fibroglandular tissue of breast and it is defined as an ellipsoid with diameter 3 cm in the x_1 - and x_2 -directions and diameter 1.5 cm in the x_3 -direction. The domain Ω_3 (fat tissue) is limited by a hemisphere of radius 3.75 cm and the muscle layer Ω_5 . The skin layer Ω_2 is located between two hemispheres of radius 3.75 and 4 cm. The rest of the domain Π contains water (layer Ω_1). Material parameters in the five layers are given by [21]

$$(6.5) \quad \begin{aligned} c_1 &= 1510 \text{ m/s}, \rho_1 = 995 \text{ kg/m}^3 \quad (\text{water}), \\ c_2 &= 1537 \text{ m/s}, \rho_2 = 1200 \text{ kg/m}^3 \quad (\text{skin}), \\ c_3 &= 1430 \text{ m/s}, \rho_3 = 928 \text{ kg/m}^3 \quad (\text{fat}), \\ c_4 &= 1510 \text{ m/s}, \rho_4 = 1020 \text{ kg/m}^3 \quad (\text{fibroglandular}), \\ c_5 &= 1580 \text{ m/s}, \rho_5 = 1041 \text{ kg/m}^3 \quad (\text{muscle}), \end{aligned}$$

and the frequencies of the considered ultrasonic problem are 250 and 500 kHz. There is a point source at $x_1 = x_3 = 0.05$, $x_2 = 0$.

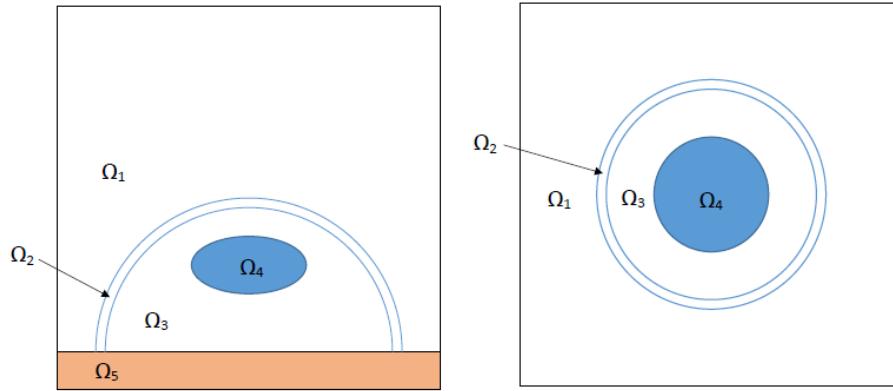


FIG. 6.2. On the left: Cross section along the xz -plane of the third test geometry. On the right: Cross section of the third test geometry along the plane $x_3 = 3.25$.

6.2. Results in one compute node. The Sisu supercomputer contains several computing nodes, each containing two 12-core Intel (Xeon) Haswell processors. In the first set of tests, the second and third test problems (wedge and muscle layers) were used to study the scalability of the parallel implementation within one node. The grid step size h was chosen according to the minimal wavelength of the problem such that there were at least 20 nodes per wavelength, that is, $\frac{c_{\min}}{f h} = 20$ with $f = \frac{\omega}{2\pi}$. The size of the corresponding linear system (4.1) is denoted by m .

The parallel efficiency of a particular case was evaluated by comparing the total execution time to the case with lowest number of cores. The ratio of these execution times was multiplied by the ratio of the number of cores. Value 1 would indicate perfect scalability.

Tables 6.1 and 6.2 show the results with increasing number of cores in one node. The parallel efficiency is reduced with increasing number of cores, and with 8 cores the efficiency is still relatively good. Therefore, in the subsequent tests the number of cores per node is fixed at 8 and only the number of nodes is increased.

TABLE 6.1
Test results with the wedge problem using one compute node.

f (kHz)	Size of linear system, m	Number of cores	Number of iterations	Total time (s)	Parallel efficiency
10	6.5e7	1	19	558.5	1.00
		2		286.0	0.98
		4		155.2	0.90
		8		85.3	0.82
		12		71.7	0.65
		16		53.6	0.65
		20		45.1	0.62
		24		38.4	0.61
20	5.1e8	4	23	1528.2	1.0
		8		840.0	0.91
		12		674.6	0.76
		16		509.5	0.75
		20		415.7	0.74
		24		350.1	0.73

TABLE 6.2
Test results with the third problem (muscle layers) using one compute node.

f (kHz)	Size of linear system, m	Number of cores	Number of iterations	Total time (s)	Parallel efficiency
250	4.3e7	1	37	690.1	1.0
		2		358.7	0.96
		4		201.4	0.86
		8		120.2	0.72
		12		104.6	0.55
		16		81.4	0.53
		20		69.3	0.50
		24		62.4	0.46
500	3.4e8	4	27	1183.9	1.0
		8		684.2	0.87
		12		566.0	0.70
		16		440.2	0.67
		20		365.6	0.65
		24		317.7	0.62

6.3. Strong scalability of the parallel implementation. In the second set of tests, the first and third test problems were used to study the strong scalability of the parallel implementation. This means that the size of the system is kept fixed while the number of nodes is increased. The number of cores per node was fixed at 8 as indicated by the results of subsection 6.2. The parallel efficiency was evaluated by comparing the total execution time to the previous case with half the number of cores. Value 2 would indicate perfect scalability. For the third test problem, the number of GMRES iterations drops when the frequency is increased.

Table 6.3 shows the results with increasing number of nodes for the first test case with planar layers. Here, the grid step size h was chosen such that there were at least 20 nodes per wavelength, that is, $\frac{c_{\min}}{f h} = 20$. For the first test problem, the number of GMRES iterations seems to be nearly independent of the frequency.

Table 6.4 shows the results with increasing number of nodes for the third test case (muscle layers). The number of grid steps per wavelength was at least 20. Illustration of the real and imaginary parts of the solution with frequency $f = 500$ kHz is given in Figure 6.3.

TABLE 6.3
Results with the first test problem using increasing number of compute nodes.

f (kHz)	Size of linear system, m	Number of nodes	Number of cores	Number of iterations	Total time (s)	Speed-up
2.5	3.0e8	1	8	11	171.3	
		2	16		91.8	1.87
		4	32		51.2	1.79
		8	64		31.6	1.62
		16	128		22.3	1.42
		32	256		15.6	1.45
5	2.4e9	1	8	12	1584.0	
		2	16		839.8	1.89
		4	32		445.3	1.89
		8	64		247.5	1.80
		16	128		152.9	1.62
		32	256		98.4	1.55

TABLE 6.4
Results with the third test problem using increasing number of compute nodes.

f (kHz)	Size of linear system, m	Number of nodes	Number of cores	Number of iterations	Total time (s)	Speed-up
250	4.3e7	1	8	37	120.2	
		2	16		66.8	1.80
		4	32		41.1	1.63
		8	64		28.9	1.42
		16	128		24.3	1.19
		32	256		22.0	1.10
500	3.4e8	1	8	27	684.2	
		2	16		368.5	1.86
		4	32		208.3	1.77
		8	64		126.9	1.64
		16	128		91.0	1.39
		32	256		66.8	1.36

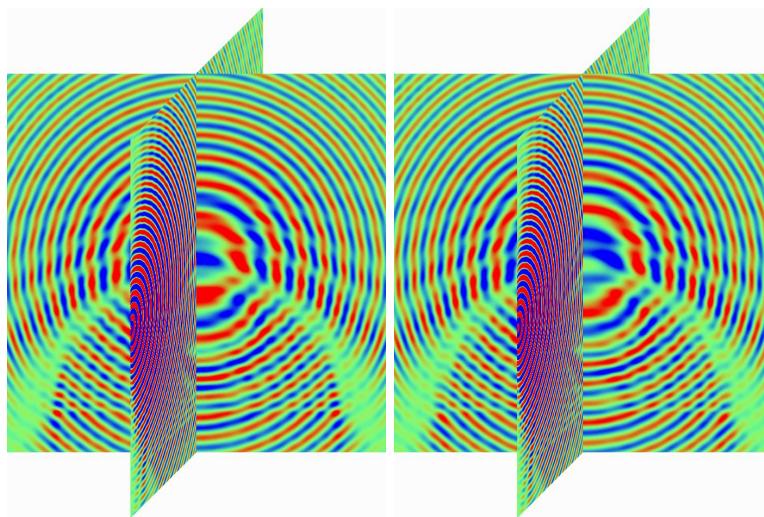


FIG. 6.3. *Real (left) and imaginary (right) parts of the solution to the third test problem with the frequency 500 kHz.*

6.4. Weak scalability of the parallel implementation. In the third set of tests, the second test problem was used to study the weak scalability of the parallel implementation, which means that the size of the linear system and the number of nodes are increased simultaneously. If the scalability were perfect, doubling of the system size and the number of nodes would keep the total time unchanged. The solution of the subdomain preconditioners is, however, not completely parallelized and requires $\mathcal{O}(m \log m)$ operations. Also the number of iterations to solve the linear system increases with system size, and therefore perfect scalability cannot be achieved. However, by dividing the total execution time by the number of iterations and by the time to solve one subdomain preconditioner we can evaluate the efficiency of the parallel part of the method.

Table 6.5 shows the results with increasing system size and number of nodes for the second test case (wedge). The number of cores per node was again fixed at 8. The grid step size h was chosen such that there were at least 10 nodes per wavelength and the frequency was increased from 10 to 80 kHz. The ratio of the total time with the product of the number of iterations and time to solve one subdomain preconditioner remains almost constant, which indicates that the parallel part of the method is

TABLE 6.5

Test results with the wedge problem using increasing system size and number of compute nodes.

f (kHz)	Size of linear system, m	Number of nodes	Number of iterations	Total time (s)	Ratio
10	8.1e6	1	19	9.42	74.0
20	6.4e7	2	23	54.40	71.7
40	5.1e8	4	29	296.10	72.9
80	4.1e9	8	39	1808.40	70.3

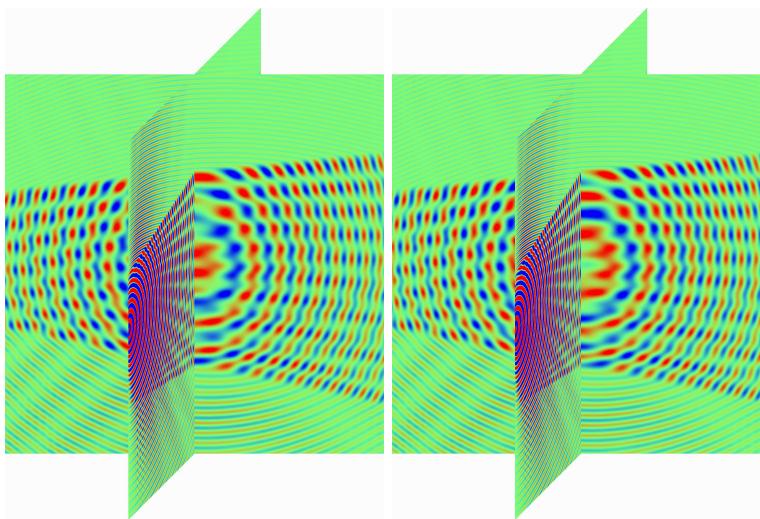


FIG. 6.4. *Real (left) and imaginary (right) parts of the solution to the second test problem with the frequency 40 kHz.*

scalably parallelized. The real and imaginary parts of the solution p to the second test problem with frequency $f = 40$ kHz are illustrated in Figure 6.4. With respect to the frequency f the number of iterations is roughly proportional to $\log f$.

7. Conclusions. An efficient domain decomposition based preconditioner for the finite element solution of the Helmholtz equation in three-dimensional layered media was considered. The discretization accuracy was improved by using a modified discretization scheme leading to fourth-order phase accuracy. Computational efficiency is achieved by using a fast direct solver for subdomain preconditioning, by the reduction of the iteration in a subspace, and by the good conditioning of the preconditioned system. An efficient parallel implementation was introduced for the iterative method, and numerical experiments with realistic test cases demonstrate the efficiency of the method. The experiments show the ability to solve problems with up to billions of unknowns.

Acknowledgment. The computational experiments of the article were performed using the computational resources of the CSC-IT Center for Science Ltd.

REFERENCES

- [1] A. BAMBERGER, P. JOLY, AND J. E. ROBERTS, *Second-order absorbing boundary conditions for the wave equation: A solution for the corner problem*, SIAM J. Numer. Anal., 27 (1990), pp. 323–352, <https://doi.org/10.1137/0727021>.

- [2] A. BANEGAS, *Fast Poisson solvers for problems with sparsity*, Math. Comp., 32 (1978), pp. 441–446.
- [3] H. BARUCQ, T. CHAUMONT-FRELET, AND C. GOUT, *Stability analysis of heterogeneous Helmholtz problems and finite element solution based on propagation media approximation*, Math. Comp., 86 (2017), pp. 2129–2157, <https://doi.org/10.1090/mcom/3165>.
- [4] T. A. DAVIS, *Direct Methods for Sparse Linear Systems*, Fundam. Algorithms 2, SIAM, Philadelphia, 2006, <https://doi.org/10.1137/1.9780898718881>.
- [5] B. ENGQUIST AND L. YING, *Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers*, Multiscale Model. Simul., 9 (2011), pp. 686–710, <https://doi.org/10.1137/100084644>.
- [6] Y. A. ERLANGGA, *Advances in iterative methods and preconditioners for the Helmholtz equation*, Arch. Comput. Methods Eng., 15 (2008), pp. 37–66, <https://doi.org/10.1007/s11831-007-9013-7>.
- [7] Y. A. ERLANGGA, C. W. OOSTERLEE, AND C. VUIK, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput., 27 (2006), pp. 1471–1492, <https://doi.org/10.1137/040615195>.
- [8] M. ESLAMINIA AND M. N. GUDDATI, *A double-sweeping preconditioner for the Helmholtz equation*, J. Comput. Phys., 314 (2016), pp. 800–823, <https://doi.org/10.1016/j.jcp.2016.03.022>.
- [9] G. FIBICH AND S. TSYNKOV, *Numerical solution of the nonlinear Helmholtz equation using nonorthogonal expansions*, J. Comput. Phys., 210 (2005), pp. 183–224, <https://doi.org/10.1016/j.jcp.2005.04.015>.
- [10] M. J. GANDER AND H. ZHANG, *A class of iterative solvers for the Helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods*, SIAM Rev., 61 (2019), pp. 3–76.
- [11] M. N. GUDDATI AND B. YUE, *Modified integration rules for reducing dispersion error in finite element methods*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 275–287, <https://doi.org/10.1016/j.cma.2003.09.010>.
- [12] E. HEIKKOLA, T. ROSSI, AND J. TOIVANEN, *Fast direct solution of the Helmholtz equation with a perfectly matched layer or an absorbing boundary condition*, Internat. J. Numer. Methods Engrg., 57 (2003), pp. 2007–2025, <https://doi.org/10.1002/nme.752>.
- [13] F. IHLENBURG, *Finite Element Analysis of Acoustic Scattering*, Appl. Math. Sci. 132, Springer-Verlag, New York, 1998, <https://doi.org/10.1007/b98828>.
- [14] K. ITO, Z. QIAO, AND J. TOIVANEN, *A domain decomposition solver for acoustic scattering by elastic objects in layered media*, J. Comput. Phys., 227 (2008), pp. 8685–8698, <https://doi.org/10.1016/j.jcp.2008.06.015>.
- [15] K. ITO AND J. TOIVANEN, *Efficient domain decomposition method for acoustic scattering in multi-layered media*, in Proceedings of the European Conference on Computational Fluid Dynamics, P. Wesseling, E. Oñate, and J. Périault, eds., TU Delft, The Netherlands, 2006, <http://resolver.tudelft.nl/uuid:6795c655-b37f-4bff-9803-9e213388610e>.
- [16] K. ITO AND J. TOIVANEN, *Preconditioned iterative methods on sparse subspaces*, Appl. Math. Lett., 19 (2006), pp. 1191–1197, <https://doi.org/10.1016/j.aml.2005.11.027>.
- [17] K. ITO AND J. TOIVANEN, *A fast iterative solver for scattering by elastic objects in layered media*, Appl. Numer. Math., 57 (2007), pp. 811–820, <https://doi.org/10.1016/j.apnum.2006.07.020>.
- [18] Y. A. KUZNETSOV, *Numerical Methods in Subspaces*, in Vychislitel'nye Processy i Sistemy II, G. I. Marchuk, ed., Nauka, Moscow, 1985.
- [19] Y. A. KUZNETSOV AND A. M. MATSOKIN, *On partial solution of systems of linear algebraic equations*, Russian J. Numer. Anal. Math. Modelling, 4 (1989), pp. 453–467, <https://doi.org/10.1515/mam.1989.4.6.453>.
- [20] E. LARSSON AND S. HOLMGREN, *Parallel solution of the Helmholtz equation in a multilayer domain*, BIT, 43 (2003), pp. 387–411, <https://doi.org/10.1023/A:1026087112862>.
- [21] N. OZMEN, *Ultrasound Imaging Methods for Breast Cancer Detection*, Ph.D. thesis, TU Delft, 2015, <http://resolver.tudelft.nl/uuid:d53a71d8-013b-4328-a0ac-e3d5a5b84e51>.
- [22] R. E. PLESSIX AND W. A. MULDER, *Separation-of-variables as a preconditioner for an iterative Helmholtz solver*, Appl. Numer. Math., 44 (2003), pp. 385–400, [https://doi.org/10.1016/S0168-9274\(02\)00165-4](https://doi.org/10.1016/S0168-9274(02)00165-4).
- [23] S. RAKOTONARIVO, M. LEGRIS, R. DESMARIE, J.-P. SESSARÉGO, AND J.-F. BOURILLET, *Forward modeling for marine sediment characterization using chirp sonars*, Geophys., 76 (2011), pp. T91–T99, <https://doi.org/10.1190/1.3590717>.
- [24] C. D. RIYANTI, A. KONONOV, Y. A. ERLANGGA, C. VUIK, C. W. OOSTERLEE, R.-E. PLESSIX, AND W. A. MULDER, *A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation*, J. Comput. Phys., 224 (2007), pp. 431–448, <https://doi.org/10.1016/j.jcp.2007.03.033>.

- [25] T. ROSSI AND J. TOIVANEN, *A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary dimension*, SIAM J. Sci. Comput., 20 (1999), pp. 1778–1796, <https://doi.org/10.1137/S1064827597317016>.
- [26] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869, <https://doi.org/10.1137/0907058>.
- [27] G. S. SANDHU, *Frequency Domain Ultrasound Waveform Tomography Breast Imaging*, Ph.D. thesis, Wayne State University, 2015, http://digitalcommons.wayne.edu/oa_dissertations/1295/.
- [28] C. C. STOLK, *A dispersion minimizing scheme for the 3-D Helmholtz equation based on ray theory*, J. Comput. Phys., 314 (2016), pp. 618–646, <https://doi.org/10.1016/j.jcp.2016.03.023>.
- [29] C. C. STOLK, *An improved sweeping domain decomposition preconditioner for the Helmholtz equation*, Adv. Comput. Math., 43 (2017), pp. 45–76, <https://doi.org/10.1007/s10444-016-9475-y>.
- [30] J. TOIVANEN, P. AVERY, AND C. FARHAT, *A multilevel FETI-DP method and its performance for problems with billions of degrees of freedom*, Internat. J. Numer. Methods Engrg., 116 (2018), pp. 661–682, <https://doi.org/10.1002/nme.5938>.
- [31] J. VIRIEUX AND S. OPERTO, *An overview of full-waveform inversion in exploration geophysics*, Geophysics, 74 (2009), pp. WCC1–WCC26, <https://doi.org/10.1190/1.3238367>.
- [32] L. ZEPEDA-NÚÑEZ AND L. DEMANET, *The method of polarized traces for the 2D Helmholtz equation*, J. Comput. Phys., 308 (2016), pp. 347–388, <https://doi.org/10.1016/j.jcp.2015.11.040>.
- [33] L. A. ZEPEDA-NÚÑEZ, *Fast and Scalable Solvers for the Helmholtz Equation*, Ph.D. thesis, Department of Mathematics, Massachusetts Institute of Technology, 2015, <http://hdl.handle.net/1721.1/99061>.