WCCM V
Fifth World Congress on
Computational Mechanics
July 7–12, 2002, Vienna, Austria
Eds.: H.A. Mang, F.G. Rammerstorfer,
J. Eberhardsteiner

Fast solvers for the Helmholtz equation with a perfectly matched layer / an absorbing boundary condition

Erkki Heikkola, Tuomo Rossi*, and Jari Toivanen

Department of Mathematical Information Technology
University of Jyväskylä, Jyväskylä, Finland
e-mail: Erkki.Heikkola@mit.jyu.fi, Tuomo.Rossi@mit.jyu.fi, Jari.Toivanen@mit.jyu.fi

Key words: Helmholtz equation, acoustic scattering, domain imbedding method, fast direct solver, perfectly matched layer

Abstract

The efficient numerical solution of exterior boundary value problems for the Helmholtz equation modeling acoustic scattering by an obstacle is considered. This exterior problem is truncated to a rectangular domain and either a perfectly matched layer (PML) or a second-order absorbing boundary condition (ABC) is posed to reduce reflections from the artificial boundary. The PML leads to a variable coefficient Helmholtz equation. The scattering problem is solved iteratively by using a domain imbedding method. This method extends the problem inside the scatterer leading to an equivalent problem in the rectangular domain. The Helmholtz operator in the rectangle with the PML or the ABC is used as a preconditioner. The partial solution variant of the cyclic reduction (PSCR) algorithm is a fast direct solver which can solve efficiently problems with this preconditioner. The separable tensor product form for the preconditioner is given and the implementation of the PML is described. The PSCR-algorithm is then described in a way that can be used in practical implementation. The numerical experiments demonstrate the accuracy and the efficiency of the proposed solution technique.

1 Introduction

We consider the efficient numerical solution of three-dimensional acoustic scattering problems, which can be formulated as exterior boundary value problems for the Helmholtz equation:

$$-\Delta u - \omega^2 u = 0, \qquad \text{in } \mathbb{R}^3 \setminus \bar{\Omega}, \tag{1}$$

$$u = g, \qquad \text{on } \partial\Omega, \tag{2}$$

$$\lim_{r\to\infty} r\left(\frac{\partial u}{\partial r} - i\omega u\right) = 0. \tag{3}$$

Here, the bounded scatterer Ω is sound soft leading to the Dirichlet boundary condition (2). The asymptotic condition for u is the classical Sommerfeld radiation condition. We approximate the problem (1) - (3) by truncating the unbounded domain and either by imposing a local non-reflecting second-order boundary condition [1] on the artificial boundary or by reformulating the problem to a variable coefficient one with a perfectly matched layer (PML) [5]. In this work, we apply a domain imbedding method which is called the algebraic fictitious domain method [12] to the solution of the problem in both cases. These methods are based on the idea of embedding the original domain into another domain with a simple geometrical form, where efficient preconditioners can be applied. We use a special absorbing extension [6, 10] for the Helmholtz problem. The discretization is based on low-order finite elements.

We describe a special fast direct solver for the arising separable preconditioning matrices. This fast direct method solves cyclically reduced problems with the partial solution method [2, 11] and, thus, the method is called the partial solution variant of the cyclic reduction (PSCR) algorithm [9, 13, 14]. The novelty in our approach is that the discretized Helmholtz equation on a rectangular domain results in a separable matrix even when a perfectly matched layer is introduced to the original problem [8]. Hence, the same solution technique can be applied at the same computational cost in preconditioning if one chooses to use either local absorbing boundary conditions or PML.

In numerical experiments, we compare the radar cross sections (RCS) computed using the absorbing boundary condition and the PML with varying parameters. Numerical experiments also demonstrate the ability of the method to solve scattering problems in a wide frequency range where the linear system can have up to order of 10^9 unknowns.

The rest of the paper is organized as follows: The formulation with an absorbing boundary condition and with a PML as well as the discretization are considered in Section 2. The separable tensor product form for the discretized Helmholtz in a rectangle is described in Section 3. Then, the PSCR algorithm is outlined for such separable matrices in Section 4. The algebraic fictitious domain method is shortly described in Section 5. Several numerical experiments are performed with two test geometries to study the error caused by introducing an artificial boundary and to demonstrate the capability to solve high frequency scattering problems.

2 Formulation and Discretization

For the numerical solution, the exterior problem (1) - (3) is truncated to the d-dimensional rectangular domain $\Pi = \prod_{k=1}^{d} [\check{x}_k, \hat{x}_k]$. The radiation condition (3) is approximated either by using an absorbing boundary condition or a perfectly matched layer (PML). These both approaches lead to the Helmholtz equation

$$
\begin{aligned}
-\nabla\cdot(\mathrm{A}\nabla u)-\omega^2\gamma u &= 0, && \text{in } \Pi\setminus\bar{\Omega}, && (4)\\
u &= g, && \text{on } \partial\Omega, && (5)\\
\mathcal{B}u &= 0, && \text{on } \partial\Pi. && (6)
\end{aligned}
$$

Here, we have the diagonal matrix

$$
\mathrm{A}(x)=\operatorname*{diag}_{k=1,\dots,d}\left\{\frac{\gamma(x)}{\gamma_k^2(x_k)}\right\},\quad \gamma(x)=\prod_{k=1}^{d}\gamma_k(x_k),\quad \gamma_k(x_k)=1+\frac{i\,\sigma_k(x_k)}{\omega}. \tag{7}
$$

The functions $\sigma_k(x_k)$ are given by

$$
\sigma_k(x_k)=\begin{cases}\sigma_0\left(\frac{\check{x}_k+\delta-x_k}{\delta}\right)^p, & x_k<\check{x}_k+\delta,\\ 0, & \check{x}_k+\delta\le x_k\le\hat{x}_k-\delta,\\ \sigma_0\left(\frac{x_k+\delta-\hat{x}_k}{\delta}\right)^p, & x_k>\hat{x}_k-\delta,\end{cases} \tag{8}
$$

where δ, σ_0 and p are nonnegative constants. The operator $\mathcal{B}$ defines the boundary condition on $\partial\Pi$.

When we use a PML, the positive parameter δ gives the thickness of the layer and B gives the trace of u on the boundary $\partial\Pi$. Thus, the problem is a variable-coefficient Helmholtz equation with Dirichlet boundary conditions on all the boundaries. In the case $\delta=0$, we use a second-order absorbing boundary condition [1] on $\partial\Pi$ and the matrix A reduces to the identity matrix leading to a constant-coefficient Helmholtz equation. For notational simplicity we present the absorbing boundary condition only for the two-dimensional case, but it can be generalized in a straightforward manner to the three-dimensional case [1, 6]. This condition involves the equation

$$
\frac{\partial u}{\partial \mathrm{n}}-i\omega u-\frac{i}{2\omega}\frac{\partial^2 u}{\partial \mathrm{s}^2}=0 \tag{9}
$$

on the faces of $\partial\Pi$ together with the equation $\frac{\partial u}{\partial \mathrm{s}}=i\omega\frac{3}{2}u$ in the corners, denoted by C, of $\partial\Pi$. Here, $\frac{\partial\cdot}{\partial \mathrm{n}}$ and $\frac{\partial\cdot}{\partial \mathrm{s}}$ denote the derivative to the outward normal direction and to the tangential direction, respectively.

For the weak formulation of the equations (4) - (6), we introduce the function spaces $V^D=H_0^1(\Pi)$ and $V^S=\{v\in H^1(\Pi)\,:\,v|_{\partial\Pi}\in H^1(\partial\Pi)\}$ corresponding to the two different boundary conditions. The Dirichlet conditions lead to the sesquilinear form

$$
a_D(u,v)=\int_{\Pi\setminus\bar{\Omega}}\left(\mathrm{A}\nabla u\cdot\nabla\bar{v}-\omega^2\gamma u\bar{v}\right)dx, \tag{10}
$$

while the absorbing boundary condition corresponds to the forms

$$
a_S(u,v)=\int_{\Pi\setminus\bar{\Omega}}\left(\nabla u\cdot\nabla\bar{v}-\omega^2 u\bar{v}\right)dx-i\int_{\partial\Pi}\left(\omega u\bar{v}+\frac{1}{2\omega}\frac{\partial u}{\partial \mathrm{s}}\frac{\partial\bar{v}}{\partial \mathrm{s}}\right)ds-\frac{3}{4}\sum_{x\in C}u(x)\bar{v}(x). \tag{11}
$$

Now, the weak formulation of the equations (4) - (6) can be represented in the following general form for $m=D$ or S: Find $u\in V^m+g$ such that

$$
a_m(u-g,v)=0\quad\forall v\in V^m, \tag{12}
$$

where g is extended from $\partial\Omega$ to $\Pi \setminus \bar{\Omega}$ in V^m.

The problem (12) is discretized by using linear and bilinear (trilinear) finite elements on a mesh which is constructed from an orthogonal mesh. Let us denote the mesh points in the x_k-direction by $x_{k,l}$, $l = 1, \ldots, n_k$. Thus, we have $x_{k,1} = \check{x}_k$ and $x_{k,n_k} = \hat{x}_k$. By locally adapting the mesh to the boundary $\partial\Omega$ and dividing those cells, which are intersected by $\partial\Omega$, to two triangles (several tetrahedrons), we obtain a good approximation for the domain $\Pi \setminus \bar{\Omega}$ [3]. The finite element discretization leads to a system of linear equations

$$\mathrm{K}\mathrm{u} = \mathrm{f}. \tag{13}$$

3 Separable Tensor Product Form

In the case that we do not have a scatterer Ω in the computational domain, we can use the fully orthogonal mesh together with bilinear (trilinear) finite elements to discretize the Helmholtz equation. Such a problem leads to a system of linear equations, where the matrix has a separable tensor product form. Let us denote this separable matrix by B. Then after a suitable renumbering of unknowns it can be represented in the form

$$\mathrm{B} = \left(\mathrm{A}_1^m - \omega^2 \mathrm{M}_1^m\right) \otimes \mathrm{M}_2^m + \mathrm{M}_1^m \otimes \mathrm{A}_2^m, \quad m = D, S. \tag{14}$$

In the three-dimensional case, we obtain analogously the form

$$\mathrm{B} = \left(\mathrm{A}_1^m - \omega^2 \mathrm{M}_1^m\right) \otimes \mathrm{M}_2^m \otimes \mathrm{M}_3^m + \mathrm{M}_1^m \otimes \mathrm{A}_2^m \otimes \mathrm{M}_3^m + \mathrm{M}_1^m \otimes \mathrm{M}_2^m \otimes \mathrm{A}_3^m. \tag{15}$$

We calculate the entries of the matrices A_k^m and M_k^m by using the mid-point quadrature rule and the trapezoidal quadrature rule, respectively. For a general numerical quadrature, the matrices are given in [8]. Then, $\mathrm{A}_k^D \in \mathbb{C}^{(n_k-2)\times(n_k-2)}$ in (14) and (15) are given by

$$\mathrm{A}_k^D = \begin{pmatrix} b_{k,2} & a_{k,2} & & \\ a_{k,2} & b_{k,3} & \ddots & \\ & \ddots & \ddots & a_{k,n_k-2} \\ & & a_{k,n_k-2} & b_{k,n_k-1} \end{pmatrix}, \tag{16}$$

where

$$a_{k,l} = -\left(\gamma_k((x_{k,l} + x_{k,l+1})/2)(x_{k,l+1} - x_{k,l})\right)^{-1} \quad \text{and} \quad b_{k,l} = -\left(a_{k,l-1} + a_{k,l}\right). \tag{17}$$

The matrices $\mathrm{A}_k^S \in \mathbb{C}^{n_k \times n_k}$ corresponding to the absorbing boundary condition are

$$\mathrm{A}_k^S = \begin{pmatrix} b_{k,1}^S & \mathrm{a}_{k,1}^T & \\ \mathrm{a}_{k,1} & \mathrm{A}_k^D & \mathrm{a}_{k,n_k-1} \\ & \mathrm{a}_{k,n_k-1}^T & b_{k,n_k}^S \end{pmatrix}, \quad \mathrm{a}_{k,1} = \begin{pmatrix} a_{k,1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathrm{a}_{k,n_k-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{k,n_k-1} \end{pmatrix}, \tag{18}$$

where

$$b_{k,1}^S = -a_{k,1} - \frac{i\omega}{2}, \qquad b_{k,n_k}^S = -a_{k,n_k} - \frac{i\omega}{2} \tag{19}$$

and entries $a_{k,l}$ are given by (17).

The diagonal matrices $\mathrm{M}_k^D \in \mathbb{C}^{(n_k-2)\times(n_k-2)}$ are given by

$$\mathrm{M}_k^D = \mathrm{diag}\{d_{k,2}, d_{k,3}, \dots, d_{k,n_k-1}\}, \qquad d_{k,l} = \frac{1}{2}\gamma_k(x_{k,l})(x_{k,l+1} - x_{k,l-1}). \tag{20}$$

The diagonal matrices $\mathrm{M}_k^S \in \mathbb{C}^{n_k \times n_k}$ corresponding to the absorbing boundary condition are

$$\mathrm{M}_k^D = \mathrm{diag}\{d_{k,1}^S, d_{k,2}, d_{k,3}, \dots, d_{k,n_k-1}, d_{k,n_k}^S\}, \tag{21}$$

where

$$d_{k,1}^S = \frac{1}{2}\gamma_k(x_{k,1})(x_{k,2} - x_{k,1}) + \frac{i}{2\omega}, \qquad d_{k,n_k}^S = \frac{1}{2}\gamma_k(x_{k,n_k})(x_{k,n_k} - x_{k,n_k-1}) + \frac{i}{2\omega} \tag{22}$$

and entries $d_{k,l}$ are given by (21).

4 Fast Direct Solver

We apply the partial solution variant of the cyclic reduction (PSCR) algorithm to solve linear systems with the separable matrix B. This method was introduced in [14] and further developed in [9, 13]. It is a cyclic reduction algorithm which uses the partial solution method [2, 11] instead of matrix polynomials. Here, we describe the implementation of the PSCR-method in a general radix–r framework, while the parallel implementation of the radix–4 variant is described in [13]. Furthermore, it can be shown [8] that it requires $\mathcal{O}(n_1 n_2 \log n_1)$ operations for two-dimensional problems and $\mathcal{O}(n_1 n_2 n_3 \log n_1 \log n_2)$ operations for three-dimensional problems.

4.1 The partial solution variant of the cyclic reduction

The algorithm operates on $m = \lceil \log_r(n_1 + 1) \rceil$ levels with r^{k-1} subproblems on the kth level. These problems are defined by dividing the mesh nodes into r^{k-1} groups in the x_1-direction such that the dividing mesh coordinates are $x_{1,l_{k,j}}$, where the indices $l_{k,j}$ are given by

$$l_{k,j} = \begin{cases} 0, & j = 1, \\ \lfloor (j-1)n_1/r^{k-1} + 1 \rfloor, & 1 < j \le r^{k-1} + 1. \end{cases} \tag{23}$$

Then, the jth subproblem corresponds to the index set

$$S_{k,j} = \{l_{k,j} + 1, \dots, l_{k,j+1} - 1\} \tag{24}$$

in the x_1-direction. We note that on the level $k = m$, some of the subproblems might vanish.

In order to describe the algorithm, we first introduce the following index sets:

$$P_{k,j} = \{l_{k+1,r(j-1)+2}, l_{k+1,r(j-1)+3}, \dots, l_{k+1,r(j-1)+r}\}, \tag{25}$$

$$R_{k,j} = \begin{cases} \{l_{k,j+1}\}, & j = 1, \\ \{l_{k,j}, l_{k,j+1}\}, & 1 < j < r^{k-1}, \\ \{l_{k,j}\}, & j = r^{k-1}, \end{cases} \tag{26}$$

$$T_{k,j} = \begin{cases} \{l_{k,j+1} - 1\}, & j = 1, \\ \{l_{k,j} + 1, l_{k,j+1} - 1\}, & 1 < j < r^{k-1}, \\ \{l_{k,j} + 1\}, & j = r^{k-1}, \end{cases} \tag{27}$$

for $k = 1, \dots, m$ and $j = 1, \dots, r^{k-1}$. Furthermore, we define the renumbered sets

$$\hat{P}_{k,j} = \bigcup_{t \in P_{k,j}} \{t - l_{k,j}\} \qquad \text{and} \qquad \hat{T}_{k,j} = \bigcup_{t \in T_{k,j}} \{t - l_{k,j}\}. \tag{28}$$

We use a special notation for submatrices and subvectors. For defining the notation, let the members of the index sets P and T be $p_1, \dots, p_{n_p}$ and $t_1, \dots, t_{n_t}$ in ascending order and with the possible multiple appearances removed, respectively. Then, the subvector $\mathrm{f}(P)$ is defined to be

$$\mathrm{f}(P) = \begin{pmatrix} \mathrm{f}(p_1) \\ \mathrm{f}(p_2) \\ \vdots \\ \mathrm{f}(p_{n_p}) \end{pmatrix} \in \mathbb{C}^{n_p \cdot n}, \quad \text{where} \quad \mathrm{f}(p) = \begin{pmatrix} \mathrm{f}_{(p-1)n+1} \\ \mathrm{f}_{(p-1)n+2} \\ \vdots \\ \mathrm{f}_{pn} \end{pmatrix} \in \mathbb{C}^{n}. \tag{29}$$

Here, n is n_2 for two-dimensional problems and $n_2 n_3$ for three-dimensional problems. Similarly, we define the submatrix

$$\mathrm{B}(P, T) = \begin{pmatrix} \mathrm{B}(p_1, t_1) & \mathrm{B}(p_1, t_2) & \cdots & \mathrm{B}(p_1, t_{n_t}) \\ \mathrm{B}(p_2, t_1) & \mathrm{B}(p_2, t_2) & \cdots & \mathrm{B}(p_2, t_{n_t}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{B}(p_{n_p}, t_1) & \mathrm{B}(p_{n_p}, t_2) & \cdots & \mathrm{B}(p_{n_p}, t_{n_t}) \end{pmatrix} \in \mathbb{C}^{(n_p \cdot n) \times (n_t \cdot n)}, \tag{30}$$

where

$$\mathrm{B}(p, t) = \begin{pmatrix} \mathrm{B}_{(p-1)n+1,(t-1)n+1} & \mathrm{B}_{(p-1)n+1,(t-1)n+2} & \cdots & \mathrm{B}_{(p-1)n+1,tn} \\ \mathrm{B}_{(p-1)n+2,(t-1)n+1} & \mathrm{B}_{(p-1)n+2,(t-1)n+2} & \cdots & \mathrm{B}_{(p-1)n+2,tn} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{B}_{pn,(t-1)n+1} & \mathrm{B}_{pn,(t-1)n+2} & \cdots & \mathrm{B}_{pn,tn} \end{pmatrix} \in \mathbb{C}^{n \times n}. \tag{31}$$

In the following algorithm, a similar notation is used to denote submatrices of a submatrix defined by (30). For example, $(\mathrm{B}(S_{k,j}, S_{k,j}))^{-1}(\hat{T}_{k,j}, \hat{P}_{k,j})$ means that we take the submatrix defined by set pair $(\hat{T}_{k,j}, \hat{P}_{k,j})$ of the inverse of the matrix $\mathrm{B}(S_{k,j}, S_{k,j})$.

Now, we can present the PSCR-algorithm as follows:

Algorithm 1 *PSCR-method for computing* $\mathrm{f} = \mathrm{B}^{-1}\mathrm{f}$.

```
do $k = m, 2, -1$
    do $j = 1, r^{k-1}$
        if $l_{k,j+1} - l_{k,j} > 1$ then
            $\mathrm{f}(R_{k,j}) = \mathrm{f}(R_{k,j}) - \mathrm{B}(R_{k,j}, T_{k,j})(\mathrm{B}(S_{k,j}, S_{k,j}))^{-1}(\hat{T}_{k,j}, \hat{P}_{k,j})\mathrm{f}(P_{k,j})$
        end if
    end do
end do
$\mathrm{f}(P_{1,1}) = \mathrm{B}^{-1}(P_{1,1}, P_{1,1})\mathrm{f}(P_{1,1})$
do $k = 2, m$
    do $j = 1, r^{k-1}$
        if $l_{k,j+1} - l_{k,j} > 1$ then
```

```
            $\mathrm{f}(P_{k,j}) = (\mathrm{B}(S_{k,j}, S_{k,j}))^{-1}(\hat{P}_{k,j}, \hat{P}_{k,j} \cup \hat{T}_{k,j})$
                        $(\mathrm{I}(P_{k,j} \cup T_{k,j}, P_{k,j})\mathrm{f}(P_{k,j})$
                        $- \mathrm{I}(P_{k,j} \cup T_{k,j}, T_{k,j})\mathrm{B}(T_{k,j}, R_{k,j})\mathrm{f}(R_{k,j}))$
        end if
    end do
end do.
```

The identity matrix I in Algorithm 1 has the same size as the matrix B. The submatrices of I are used to assemble vectors to a suitable structure for subproblems. The multiplications by the inverse matrices in Algorithm 1 are considered in Section 4.2.

4.2 Partial solution method

The partial solution method for the direct solution of linear systems corresponding to elliptic equations in rectangular domains was introduced in [2, 11]. The method is a special implementation of the classical method of separation of variables, and its efficiency is based on the assumptions that only a sparse set of the solution components of the linear system is required and that the right-hand side vector has only a few nonzero components. Then, the partial solution procedure is obtained directly from the method of separation variables by neglecting arithmetical operations with the zero components. In the following, we consider the multiplication by $\mathrm{B}^{-1}(P, T)$. All the other subproblems in Algorithm 1 have a similar structure and, thus, they can be solved in the same way.

The method of separation of variables involves the solution of the generalized eigenvalue problem

$$\mathrm{A}_1^m \mathrm{W} = \mathrm{M}_1^m \mathrm{W} \Lambda, \tag{32}$$

where the matrix W contains the eigenvectors as its columns, and the diagonal matrix Λ contains the eigenvalues $\lambda_1, \ldots, \lambda_{n_1}$. It is necessary for the applicability of the method that the eigenvectors form a basis for the space $\mathbb{C}^{n_1}$ and satisfy the condition $\mathrm{W}^T \mathrm{M}_1^m \mathrm{W} = \mathrm{I}$. The eigenvalue problem (32) is solved once in the initialization stage of the solver. Now, it is straightforward to see that the inverse of B in (14) (two-dimensional problem) is

$$\mathrm{B}^{-1} = (\mathrm{W} \otimes \mathrm{I})((\Lambda - \omega^2 \mathrm{I}) \otimes \mathrm{M}_2^m + \mathrm{I} \otimes \mathrm{A}_2^m)^{-1}(\mathrm{W}^T \otimes \mathrm{I}) \tag{33}$$

and the inverse of B in (15) (three-dimensional problem) is

$$\mathrm{B}^{-1} = (\mathrm{W} \otimes \mathrm{I})(((\Lambda - \omega^2 \mathrm{I}) \otimes \mathrm{M}_2^m + \mathrm{I} \otimes \mathrm{A}_2^m) \otimes \mathrm{M}_3^m + \mathrm{I} \otimes \mathrm{M}_2^m \otimes \mathrm{A}_3^m)^{-1}(\mathrm{W}^T \otimes \mathrm{I}). \tag{34}$$

By exploiting the special sparsity of problem, we obtain the following partial solution algorithm for two-dimensional problems.

Algorithm 2 *Partial solution method for computing* $\mathrm{v} = \mathrm{B}^{-1}(P, T)\mathrm{g}$.

```
$\mathrm{v} = 0$
do $k = 1, n_1$
    $\hat{\mathrm{g}} = (\mathrm{W}^T \otimes \mathrm{I})(\{k\}, T)\mathrm{g}$
    $\hat{\mathrm{v}} = ((\lambda_k - \omega^2)\mathrm{M}_2^m + \mathrm{A}_2^m)^{-1}\hat{\mathrm{g}}$
    $\mathrm{v} = \mathrm{v} + (\mathrm{W} \otimes \mathrm{I})(P, \{k\})\hat{\mathrm{v}}$
end do
```

Note that FFT-transformation cannot be used to perform the multiplications with W and W^T, since these matrices are not of suitable form. Note also that the matrix $(\Lambda - \omega^2\mathrm{I}) \otimes \mathrm{M}_2^m + \mathrm{I} \otimes \mathrm{A}_2^m$ appearing in (33) is a block diagonal matrix and the diagonal blocks are tridiagonal matrices. Hence, the related linear systems appearing in Algorithm 2 are easily solved with the LU-decomposition tuned for tridiagonal matrices. Therefore, the solution of each such system requires $\mathcal{O}(n_2)$ floating point operations.

For three-dimensional problems, the partial solution method has the same structure as Algorithm 2. The matrix $((\Lambda-\omega^2\mathrm{I})\otimes\mathrm{M}_2^m+\mathrm{I}\otimes\mathrm{A}_2^m)\otimes\mathrm{M}_3^m+\mathrm{I}\otimes\mathrm{M}_2^m\otimes\mathrm{A}_3^m$ is again block diagonal. This time the diagonal blocks are separable block tridiagonal matrices and we can use the PSCR-method for two-dimensional problems to solve the corresponding linear systems.

5 Domain Imbedding Method

In Section 4, we described a fast direct solver for the Helmholtz equation in the rectangular domain Π. There are several domain imbedding methods which can take advantage of fast direct solvers in the iterative solution of (13). Here, we consider an algebraic fictitious domain method [12] in which the matrix K is extended to have the same dimension as the matrix B. We use the absorbing extension [6, 10] for this. This leads to a new system of linear equations

$$\begin{pmatrix} \mathrm{K} & \mathrm{K}_{12} \\ 0 & \mathrm{K}_{22} \end{pmatrix} \begin{pmatrix} \mathrm{u} \\ \mathrm{u}_2 \end{pmatrix} = \begin{pmatrix} \mathrm{f} \\ 0 \end{pmatrix}. \tag{35}$$

The block K_{12} in (35) is the same as the corresponding block in the preconditioner B. Furthermore, K_{22} is the discrete counterpart of the Helmholtz operator in Ω with the first-order absorbing boundary condition on $\partial\Omega$. For details, see [6, 10].

We solve the linear system (35) iteratively with a right preconditioned GMRES method. Most of the rows of the extended matrix in (35) and the preconditioner B coincide. This can be used to reduce the GMRES iterations to a small subspace corresponding to the nodes on the boundary $\partial\Omega$ and the ones next to it. This reduces the memory and computational requirements vastly [6, 7, 9].

6 Numerical Experiments

Our first test problem is the scattering of a plane wave by a sphere. The domain Ω is chosen to be the unit sphere of $\mathbb{R}^3$. For this geometry, the analytical solution is given by the sum

$$u(x) = -\sum_{k=0}^{\infty} i^k (2k+1) \frac{j_k(\omega)}{h_k^{(1)}(\omega)} h_k^{(1)}(\omega r) P_k(\cos\theta), \tag{36}$$

where r is the distance from the origin, j_k are the spherical Bessel function, $h_k^{(1)}$ are the spherical Hankel functions of the first kind, P_k denote the Legendre polynomials, and θ is the angle between x and the propagation direction of the incidence plane wave [4]. We study the accuracy of the truncated and discretized problems by computing the radar cross section

$$RCS = 10\log_{10}\left(\frac{2\pi}{\lambda^2}\lim_{r\to\infty}|r\,u(x)|^2\right), \tag{37}$$

which describes the amplitude of a scattered wave in different directions. We approximate the RCS for the solution of the discrete problem as a sum of the Green functions for the exterior Helmholtz problem [7]. We compare the RCS in the x_1x_2-plane and, thus, x has the form $(r\cos\theta, r\sin\theta, 0)^T$, $\theta \in [0, 2\pi]$, in (37). The incident wave is chosen to be $e^{-i\omega x_2}$. The wave number ω is 2π and, thus, the wave length λ is one.

The rectangular domain Π is $[-6, 6]^3$ and the orthogonal meshes are uniform with the mesh steps sizes (denoted by h) 1/10, 1/20 and 1/40. We use the second-order absorbing boundary condition and a perfectly matched layers with 2, 4 and 8 elements on the layer (δ/h). For the PML, the parameter σ_0 in (8) is obtained by minimizing the modulus of the reflection coefficient R for a perpendicular plane wave [8]. This is done numerically by computing R for $\sigma_0 = \exp((k/1000)\log 1000)$, $k = 0, 1, \ldots, 1000$ and, then, by choosing the one corresponding to the smallest $|R|$. The L_2 errors in the RCS are given in Tab. 1 for nine different exponents p for the PML.

In the GMRES, the stopping criterion was that the norm of the residual is reduced by the factor 10^{-6}. The number of required iterations were the same and, also, the running times were essentially the same for all problems having the same number of unknowns. The results are given in Tab. 2. The experiments were performed on a HP 9000/J5600 workstation.

Table 1: The L_2 errors in the RCS.

	$\lambda/h = 10$			$\lambda/h = 20$			$\lambda/h = 40$		
ABC	0.762			0.236			0.070		
δ/h	2	4	8	2	4	8	2	4	8
$p = 0.0$	0.827	1.276	0.805	1.152	0.210	0.340	1.312	0.600	0.062
$p = 0.5$	3.188	1.050	0.787	2.036	1.377	0.608	1.047	0.954	0.474
$p = 1.0$	2.104	0.795	0.716	1.281	0.518	0.300	0.643	0.379	0.122
$p = 1.5$	1.276	0.796	0.774	0.652	0.211	0.230	0.319	0.103	0.058
$p = 2.0$	0.909	0.789	0.787	0.493	0.221	0.217	0.540	0.061	0.060
$p = 2.5$	0.808	0.784	0.787	0.565	0.215	0.218	0.496	0.062	0.073
$p = 3.0$	0.794	0.775	0.786	0.442	0.221	0.218	0.311	0.070	0.062
$p = 3.5$	0.829	0.753	0.786	0.235	0.210	0.219	0.126	0.060	0.062
$p = 4.0$	0.929	0.722	0.787	0.323	0.204	0.219	0.114	0.056	0.062

Table 2: The number of iterations and running times for experiments with the sphere.

λ/h	10	20	40
n_k	60	120	240
iter	23	31	49
time	21s	219s	2551s

The second test geometry Ω is a submarine-like object shown in Fig. 1. With this scattering problem we only used the second-order absorbing boundary condition. The artificial boundary was chosen to be at least two wave lengths away from $\partial\Omega$. In our high-frequency experiments, the submarine-like object was between 20 and 200 wave lengths long and there was always 20 nodes per one wave length. The largest

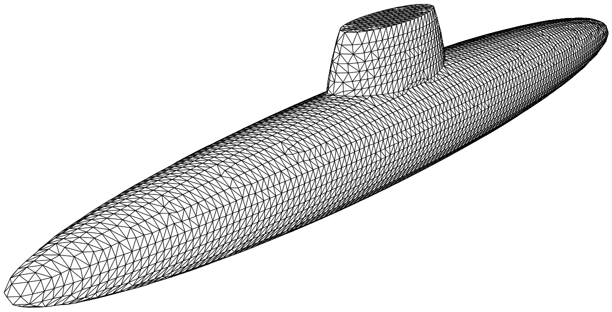

Figure 1: Submarine-like object for high frequency experiments.

problem has 1.3 billion unknowns in the extended system (35). We used a SGI Origin 2000 using up to 32 processors and 32 Gbytes of memory. Some of the results are reported in Tab. 3, where "D/λ" gives the length of Ω in terms of waves, "N" is approximately the number of unknowns in (35), "iter" is the number of GMRES iterations required to reduce the norm of the residual by the factor of 10^{-6}, "p" is the number of processors and "time" is the running time in hours and minutes. More detailed description of these experiments, together with the parallel implementation of the solver, is given in [7].

Table 3: Results with submarine-like object.

D/λ	N	iter	p	time
20	8.0×10^6	41	8	4m
40	2.8×10^7	53	16	11m
80	1.3×10^8	70	32	49m
160	7.3×10^8	97	32	8h11m
200	1.3×10^9	105	32	22h35m

Acknowledgements

This research was supported by the Academy of Finland, grants 49006, 53588 and 66407.

References

[1] 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), pp. 323–352.

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

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

[4] D. Colton, R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, Second Edition, Springer-Verlag, Berlin, (1998).

[5] I. Harari, M. Slavutin, E. Turkel, *Analytical and numerical studies of a finite element PML for the Helmholtz equation*, J. Comput. Acoust., 8, (2000), pp. 121–137.

[6] E. Heikkola, Y. A. Kuznetsov, K. N. Lipnikov, *Fictitious domain methods for the numerical solution of three-dimensional acoustic scattering problems*, J. Comput. Acoust., 7, (1999), pp. 161–183.

[7] E. Heikkola, T. Rossi, J. Toivanen, *A parallel fictitious domain method for the three-dimensional Helmholtz equation*, Report B 9/2002, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, (2000).

[8] E. Heikkola, T. Rossi, J. Toivanen, *Fast direct solution of the Helmholtz equation with a perfectly matched layer / an absorbing boundary condition*, Report B 1/2002, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä (2002).

[9] Y. A. Kuznetsov, *Computational methods in subspaces*, in *Computational Processes and Systems*, no. 2, Nauka, Moscow (1985), pp. 265–350.

[10] Y. A. Kuznetsov, K. N. Lipnikov, *3D Helmholtz wave equation by fictitious domain method*, Russian J. Numer. Anal. Math. Modelling, 13, (1998), pp. 371–387.

[11] Y. A. Kuznetsov, A. M. Matsokin, *Partial solution of systems of linear algebraic equations*, in *Numerical Methods in Linear Algebra. Proc. Third Sem. Methods Numer. Appl. Math., Novosibirsk, 1978*, Akad. Nauk SSSR Sibirsk. Otdel. Vychisl. Tsentr, (1978), pp. 62–89 (in Russian).

[12] G. I. Marchuk, Y. A. Kuznetsov, A. M. Matsokin, *Fictitious domain and domain decomposition methods*, Sov. J. Numer. Anal. Math. Modelling, 1, (1986), pp. 3–35.

[13] 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), pp. 1778–1796.

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