Contemporary Mathematics
Volume 00, 0000

A Domain Decomposition Approach to Solving the Helmholtz Equation with a Radiation Boundary Condition

OLIVER ERNST AND GENE H. GOLUB

July 27, 1992

ABSTRACT. In the numerical solution of scattering problems, an important computational kernel problem is that of solving the Helmholtz equation on regular domains with so-called approximative radiation conditions imposed on the boundary. While very efficient techniques are available for solving Helmholtz equations on regions such as rectangles and circles, these can often not be applied due to the radiation condition. Our domain decomposition approach to solving this problem consists of separating the interior problem from the boundary problem, solving each problem separately and then iterating to obtain a solution of the complete problem. This can be described as preconditioning the suitably reordered system and applying one of the many iterative methods for non-Hermitian systems. The preconditioning step avoids the difficulty of the radiation condition, and can thus be performed using a fast solver.

1. Introduction

An often occuring problem in mathematical physics [**5**] is that of computing the scattered field of an object upon which monochromatic acoustic or electromagnetic waves are impinging. Assuming a unit index of refraction and harmonic time dependence, the wave field satisfies the Helmholtz or reduced wave equation

$$-\Delta u - k^2 u = f$$

1991 *Mathematics Subject Classification.* Primary 65F05, 65F10; Secondary 65N55, 65N06.

The reasearch of the second author was supported in part by the National Science Foundation under Grant NSF CCR-8821078.

This paper is in final form and no version of it will be submitted for publication elsewhere.

This paper was presented at the Sixth International Conference on Domain Decomposition Methods which was held in Como, Italy, June 1992.

outside of the domain occupied by the scattering body, on the boundary of which suitable boundary conditions depending on material properties are imposed. Here, k denotes the wave number and f is a sufficiently regular source function. In addition, to make the solution of this problem unique, an asymptotic condition for the behavior of the solution far away from the scatterer is needed, the well-known Sommerfeld radiation condition

$$\frac{\partial u}{\partial n} - iku = o\left(r^{\frac{1-d}{2}}\right) \text{ as } r \longrightarrow \infty,$$

in which r denotes the distance from a fixed point (usually near the center of the scatterer) and d denotes the dimension of the underlying space. We will only consider two dimensions here. Physically, this condition causes the solution to be an outgoing wave.

To render this problem numerically tractable, the exterior problem needs to be restriced to a finite computational domain. Thus, the scattering body is imbedded into some regularly shaped auxiliary domain such as a circle or a rectangle. Instead of the Sommerfeld condition at infinity, an approximation to it is imposed on the boundary of the auxiliary domain. One class of approximations consists of differential operators, which annihilate as many terms of a far-field expansion of the solution as possible [**4**]. The simplest (and crudest) of these is obtained simply by applying the Sommerfeld condition

$$\frac{\partial u}{\partial n} - iku = 0$$

on the artificial boundary rather than at infinity. The partial derivative with respect to n denotes the derivative in the direction of the outer normal of the boundary. The higher the order of the differential operator occuring in the boundary condition, the smaller the auxiliary domain can be made to achieve the same accuracy of approximation. We intend to look at higher approximations in future reports.

Finally, there are techniques by which the boundary condition on the scattering body can be eliminated, further simplifying the problem. These are Lagrange multiplier techniques (cf.[**6**]) and the capacitance matrix method (cf.[**1, 2, 9**]). The latter uses a discrete analog of potential theory to write the problem without the scatterer as a low-rank modification of the original problem. After these simplifications have been applied, the problem takes on the shape in which we will approach it, namely

$$-\Delta u - k^2 u = f \qquad \text{in } R$$

$$\frac{\partial u}{\partial n} - iku = 0 \quad \text{on } \partial R.$$

The domain $R = (0,1) \times (0,1)$ is the unit sqare, k is real and u and f are complex valued functions defined on the closure of R.

If the problem is discretized using finite differences on a regular mesh, the resulting linear system of equations has block tridiagonal structure. For problems of this type resulting from the discretization of separable elliptic boundary value problems on rectangles or circles, so-called fast solvers can solve the linear system in $O(N^2 \log N)$ arithmetic operations, N being the number of mesh points in one direction [**3, 10**]. However, due to the radiation boundary condition, these techniques cannot be applied to our problem in any obvious way. Thus, we attempt a domain decomposition approach which separates the interior unknowns from the boundary unknowns, solves these two problems separately in an efficient way, and uses an iterative method to put the solutions together. Since we can apply fast solvers to the interior problem, we can very efficiently precondition with the solution of the interior problem. If the number of iterations needed to solve this preconditioned problem can be bounded independent of the mesh-size, this approach can itself be regarded as a fast solver. Although we have not been able to prove the existence of such a bound, the numerical results presented in Section 4 seem to suggest this to be the case.

2. Discretization of the problem

In this section, the linear system arising through one simple finite-difference discretisation of the problem is derived. For now, we are not concerned with the exact order of the discretization error, noting only that standard error estimates show it to be at least $O(h)$ [**8**].

If we discretize the equation with the five-point discrete Laplacian and the normal derivatives in the boundary condition with one-sided differences using a uniform mesh-width $h = 1/(N+1)$ we arrive at the linear system

$$A\mathbf{x} = b$$

for the values of the solution on the mesh, where

$$A = \begin{bmatrix} \tilde{T} & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & \tilde{T} \end{bmatrix} \in \mathbb{C}^{(N+2)^2 \times (N+2)^2},$$

I denotes the identity in $\mathbb{C}^{N+2}$ and T is given by

$$T = \begin{bmatrix} 3 - k^2h^2 - ikh & -1 & & & \\ -1 & 4 - k^2h^2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 - k^2h^2 & -1 \\ & & & -1 & 3 - k^2h^2 - ikh \end{bmatrix}.$$

The matrix $\tilde{T}$ is a shift of T by the mesh dependent constant $\alpha = -1 - ikh$, i.e.

$$\tilde{T} = T + \alpha I.$$

In order to write the system in a more compact fashion, we introduce the two auxiliary matrices

$$J = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} 1 & 0 & & & \\ 0 & 0 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & 0 & 0 \\ & & & 0 & 1 \end{bmatrix}.$$

Now, using Kronecker sums and products, we can write A as

$$(I \otimes T) + (\alpha V - J) \otimes I = T \oplus (\alpha V - J).$$

3. Preconditioned Iterative Methods

In the absence of a fast direct solver, the size and sparsity of the discretization matrix suggests using an iterative method to solve the discretized Helmholtz equation. Since the system is non-Hermitian, one of the recently developed Krylov subspace methods such as BCG, GMRES or QMR suggest themselves. We chose the latter due to the short recurrences it uses to compute the iterates, making it a very efficient method. Of course, this alone does not yield a fast solver. However, if the number of iterations needed to achieve a given residual norm reduction can be bounded independently of the mesh size, a very efficient method results. To keep the number of iterations low, it is necessary to precondition the system, i.e. to modify it to either

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$$

or

$$AM^{-1}\mathbf{y} = \mathbf{b}, \qquad M\mathbf{x} = \mathbf{y},$$

where M^{-1} is an approximation of the inverse A^{-1} of A.

As in other iterative methods of this type, this results in having to solve an additional linear system involving the matrix M at each step of the iteration. A fast algorithm is obtained by choosing the preconditioner M such that standard fast solvers such as Fourier techniques or cyclic reduction can be used to perform the preconditioning step.

In the following, after a brief description of the QMR method, we present several different ideas for obtaining preconditioners for our problem.

3.1. QMR. The iterative procedure we chose for our computations is the recently proposed QMR method of Freund and Nachtigal [**7**], which is a Krylov subspace method that uses the nonsymmetric Lanczos process to generate the bases for the Krylov subspaces. This combines the short recurrences of the BCG method with a quasi-optimality property for minimizing the residual in the current Krylov space. Also, to remedy the exact and numerical breakdowns to which the Lanczos process is susceptible, so-called 'look-ahead' techniques are usually employed in QMR to skip over non-existing Lanczos vectors, making QMR a very robust algorithm. In our computations, no look-ahead was performed since the class of problems was a very restricted one and breakdowns were not observed.

For later reference, we give a short summary of the QMR algorithm. The iterates $\mathbf{x}_n, \quad n = 0, 1, \ldots$ are chosen such that

$$\mathbf{x}_n \in \mathbf{x}_0 + K_n(\mathbf{r}_0, A), \quad \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0,$$

where $K_n(\mathbf{r}_0, A)$ is the n-th Krylov space of A w.r.t. the initial vector $\mathbf{r}_0$. The nonsymmetric Lanczos process for generating a basis for K_n with which to construct the iterates, is summarized as follows:

Lanczos Algorithm.

set $\mathbf{v}_0 = \mathbf{w}_0 = 0$
$\mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$, $\mathbf{w}_1$ *arbitrary*, $\|\mathbf{w}_1\| = 1$
for $n = 1, 2, \ldots$
 $\delta_n = \tilde{\mathbf{w}}_n^T \tilde{\mathbf{v}}_n$, *if* $\delta_n = 0$ *set* $L = n-1$, *stop*
 $\mathbf{v}_n = \tilde{\mathbf{v}}_n/\gamma_n$
 $\mathbf{w}_n = \tilde{\mathbf{w}}_n/\beta_n$
 $\gamma_n\beta_n = \delta_n$
 $\alpha_n = \mathbf{w}_n^T A \mathbf{v}_n$
 $\tilde{\mathbf{v}}_{n+1} = A\mathbf{v}_n - \alpha_n \mathbf{v}_n - \beta_n \mathbf{v}_{n-1}$
 $\tilde{\mathbf{w}}_{n+1} = A^T\mathbf{w}_n - \alpha_n \mathbf{w}_n - \gamma_n \mathbf{w}_{n-1}$
 if $\tilde{\mathbf{v}}_{n+1} = 0$ *or* $\tilde{\mathbf{w}}_{n+1} = 0$ *then stop.*

This yields

$$\begin{aligned} K_n(A, \mathbf{r}_0) &= \text{span } (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) \\ K_n(A^T, \mathbf{w}_1) &= \text{span } (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n), \end{aligned}$$

i.e. a basis for the Krylov space of A^T is also generated. The bases satisfy the relations

$$\begin{aligned} AV_n &= V_n H_n + [\begin{array}{cccc} 0 & \ldots & 0 & \tilde{\mathbf{v}}_{n+1} \end{array}] \\ A^T W_n &= W_n H_n^T + [\begin{array}{cccc} 0 & \ldots & 0 & \tilde{\mathbf{w}}_{n+1} \end{array}] \\ W_n^T V_n &= I \end{aligned}$$

where the matrices V_n and W_n contain the columns $\mathbf{v}_j$ and $\mathbf{w}_j$, $j = 1, \ldots, n$, respectively. The matrix H_n is the tridiagonal matrix associated with the Lanczos process:

$$H_n = [\ I_n \quad 0\ ]\ H_n^{(e)}$$

where

$$H_n^{(e)} = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \gamma_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_n & \alpha_n \\ 0 & & \cdots & 0 & \gamma_{n+1} \end{bmatrix} \in \mathbb{C}^{(n+1)\times n}.$$

These quantities having been generated by the Lanczos process, the QMR iterates are obtained by writing the residual of the n-th iterate $\mathbf{x}_n = \mathbf{x}_0 + V_n \mathbf{z}_n$, where $\mathbf{z}_n \in \mathbb{C}^n$, as

$$\mathbf{r}_n = \mathbf{b} - A\mathbf{x}_n = \mathbf{r}_0 - AV_n\mathbf{z}_n = V_{n+1}\left(d_n - H_n^{(e)}\mathbf{z}_n\right)$$

and quasi-minimizing the residual by choosing $\mathbf{z}_n$ to satisfy

$$\|d_n - H_n^{(e)}\mathbf{z}_n\| = \min_{\mathbf{z}_n \in \mathbb{C}^n} \|d_n - H_n^{(e)}\mathbf{z}_n\|$$

We note here a property of QMR which will be important with regard to choosing preconditioners later. The QMR residuals satisfy the following error bound due to Freund and Nachtigal [**7**]:

THEOREM 3.1. *If the $n \times n$ matrix H_n generated by n steps of the Lanczos algorithm is diagonalizable, then the residual vectors of the QMR algorithm satisfy*

$$\|\mathbf{r}_n\| \le \|\mathbf{r}_0\|\ \kappa(H_n)\ \sqrt{n+1}\ \varepsilon^{(n)}$$

where

$$\varepsilon^{(n)} = \min_{p(0)=1}\ \max_{\lambda \in \lambda(A)} |p(\lambda)|$$

where p is a polynomial of degree n, $\lambda(A)$ denotes the spectrum of A and $\kappa(\cdot)$ denotes the condition number with respect to the 2-norm.

Thus, if the spectrum of A is clustered around a few points, then the constant $\varepsilon^{(n)}$, which is a measure of how well the function zero can be approximated on the spectrum of A, by polynomials of degree n normalized to one at zero, will be small yielding a small error bound. In particular, if A is a rank p perturbation of the identity, then A has at most $p+1$ distinct eigenvalues, so that QMR will converge in at most $p+1$ steps.

3.2. Preconditioners obtained by altering the radiation boundary condition. Since much of the difficulty in this problem stems from the radiation condition, a fast preconditioner can result from simply replacing this condition at parts of the boundary with a simpler one. This is due to the fact that standard fast Poisson solver software such as the CBLKTR routine from the FISHPAK library cannot handle the systems arising from the discretization of the problem containing the radiation condition. Several variants of this idea are discussed below.

3.2.1. *Applying a Neumann Condition on two boundaries.* Cyclic reduction can still be used if the radiation boundary condition is retained along the sides $x = 0$ and $x = 1$ but a Neumann condition replaces the radiation condition on the other two sides. Using one-sided differences at the boundary, this yields the preconditioner

$$M_1 = \begin{bmatrix} (T-I) & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & (T-I) \end{bmatrix}$$

where T is the discretization matrix using one-sided differencing for the radiation condition on the sides.

3.2.2. *Applying a Dirichlet Condition on two boundaries.* Another possibility is to retain the radiation boundary condition along the sides $x = 0$ and $x = 1$ but to impose a zero Dirichlet condition on the remaining two sides of the unit square. The preconditioning matrix is obtained here by replacing the upper and lower $\tilde{T}$ blocks by T, thus yielding a matrix to which the classical cyclic reduction technique using Chebyshev polynomials can be applied, i.e.

$$M_2 = \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{bmatrix}.$$

3.3. Domain decomposition approaches. In the following approaches, we separate the discrete domain into all or part of the boundary unknowns and interior unknowns. This results in coupled subproblems that can be solved separately in an efficient way using a fast solver for the interior and solving few tridiagonal systems for the boundary. The coupling is reintroduced by iterating for the solution of the complete problem.

3.3.1. *A line reordering.* If the solution vector is partitioned into unknowns corresponding to one line of the rectangular grid(i.e. points on the grid with the same y-value), we obtain a block vector of the form

$$(\mathbf{x}_0, \ldots, \mathbf{x}_{N+1})^T,$$

where the $\mathbf{x}_0$ and $\mathbf{x}_{N+1}$ blocks correspond to the first and last lines, respectively. To separate interior from boundary points, the vector is permuted to

$$(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}_0, \mathbf{x}_{N+1})^T$$

which transforms the coefficient matrix of the equation to

$$\tilde{A} = \left[\begin{array}{ccccc|cc} T & -I & & & & -I & \\ -I & T & -I & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & -I & T & -I & & \\ & & & -I & T & & -I \\ \hline -I & & & & & \tilde{T} & \\ & & & & -I & & \tilde{T} \end{array}\right] \in \mathbb{C}^{(N+2)^2 \times (N+2)^2}.$$

We write the reordered system as

$$\tilde{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} := \begin{bmatrix} B & C \\ C^T & D \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \tag{1}$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$, $\mathbf{y} = (\mathbf{x}_0, \mathbf{x}_{N+1})^T$ and $\mathbf{b}$ is partitioned accordingly among $\mathbf{b}_1$ and $\mathbf{b}_2$. The large $N \times N$ block B is the block tridiagonal matrix obtained from discretizing the Helmholtz equation with zero Dirichlet boundary conditions at $y = 0$ and $y = 1$. The lower right 2×2 block D is small and block diagonal, the diagonal blocks themselves being again tridiagonal, so it is easily solved directly. Thus, if the two blocks C and C^T containing the couplings are replaced by zero blocks, the resulting matrix

$$M_3 = \begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix}$$

can be expected to yield a good preconditioner to $\tilde{A}$, whose application can be performed with fast techniques. The rank difference between the preconditioner and the original system is $4(N + 2)$. Also, the preconditioned system turns out to be two-cyclic, a property which we will take advantage of in Section 3.5 to reduce the number of operations necessary in the iteration by half.

A third preconditioner is obtained with the help of the Schur complement of (1). To this end, we form the equivalent system

$$\mathbf{x} = B^{-1}(\mathbf{b}_1 - C\mathbf{y}) \tag{2}$$

$$(D - C^T B^{-1} C)\mathbf{y} = \mathbf{b}_2 - C^T B^{-1} \mathbf{b}_1 \tag{3}$$

by eliminating $\mathbf{x}$ from the second block equation in (1). This suggests the solution approach of obtaining the right hand side of (2) by one fast solve, then using iteration to solve (3) for $\mathbf{y}$ after which $\mathbf{x}$ is obtained from (2) with another fast solve. Furthermore, for the iteration, (3) can be easily preconditioned with D.

The coefficient matrix $(D - C^T B^{-1} C)$ in system (3) is known as the Schur complement of $\tilde{A}$ and has dimension $2(N+2)$ which is much smaller than the $(N+2)^2$ of the original system. However, an application of the fast solver is still necessary at each iteration step in order to perform the matrix-vector multiplication with the Schur complement. This can only be avoided by either computing the Schur complement in its entirety prior to the iteration or by finding a more efficient technique of applying the Schur complement to a vector. The very simple structure of the two matrices C and C^T multiplying B^{-1} in (3) gives some hope that this is possible. Indeed, if the matrix B^{-1} is partitioned into N^2 blocks of size $(N+2) \times (N+2)$ as

$$B^{-1} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,N} \\ \vdots & & \vdots \\ X_{N,1} & \cdots & X_{N,N} \end{bmatrix},$$

then (3) becomes

$$\left(\begin{bmatrix} \tilde{T} & 0 \\ 0 & \tilde{T} \end{bmatrix} - \begin{bmatrix} X_{1,1} & X_{1,N} \\ X_{N,1} & X_{N,N} \end{bmatrix} \right) \mathbf{y} = \mathbf{b}_2 - C^T B^{-1} \mathbf{b}_1 \tag{4}$$

in which only the four blocks $X_{1,1}$,$X_{1,N}$,$X_{N,1}$ and $X_{N,N}$ of B^{-1} appear. These can be written as the first and last blocks of the solutions of the two matrix equations

$$B \begin{bmatrix} X_{1,1} \\ X_{1,2} \\ \vdots \\ X_{1,N} \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ and } B \begin{bmatrix} X_{N,1} \\ \vdots \\ X_{N,N-1} \\ X_{N,N} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \end{bmatrix}. \tag{5}$$

Since these matrix equations contain so many zero blocks, we might hope to obtain the four matrices in question by some simple direct method. One approach by which we could accomplish this is by explicitly computing the block LU decomposition of B and using it to find an expression for the solution components that we need. If we write B as

$$\begin{bmatrix} T & -I & & \\ -I & T & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & T \end{bmatrix} = \begin{bmatrix} I & & & \\ L_2 & I & & \\ & \ddots & \ddots & \\ & & L_N & I \end{bmatrix} \begin{bmatrix} U_1 & -I & & \\ & \ddots & \ddots & \\ & & U_{N-1} & -I \\ & & & U_N \end{bmatrix}$$

and compare corresponding blocks, we arrive at the following recursions

$$
\begin{array}{rcllcl}
U_1 & = & T & & & \\
L_2 & = & -U_1^{-1}, & U_2 & = & T + L_2 \\
 & \vdots & & & & \\
L_N & = & -U_{N-1}^{-1}, & U_N & = & T + L_N.
\end{array} \tag{6}
$$

The matrices U_k and L_k can be written as

$$
\begin{array}{rcll}
L_k & = & -[p_{k-1}(T)]^{-1}\, p_{k-2}(T) & k = 2, \dots, N \\
U_k & = & [p_{k-1}(T)]^{-1}\, p_k(T) & k = 1, \dots, N
\end{array}
$$

where the p_k are the polynomials defined by the following initial conditions and recurrence:

$$
\begin{array}{rcl}
p_0(z) & = & 1 \\
p_1(z) & = & z \\
p_{k+1}(z) & = & z\, p_k(z) - p_{k-1}(z).
\end{array} \tag{7}
$$

Using this representation of U_k and L_k, we can represent the solution of the matrix equations (5) as follows:

$$
\begin{array}{rcl}
X_{1,k} & = & p_{k-1}(T) \sum_{j=k}^{N} p_{j-1}(T)^{-1} p_j(T)^{-1} \\
X_{N,k} & = & p_N(T)^{-1} p_{k-1}(T)
\end{array} \tag{8}
$$

for $k = 1, \dots, N$. This gives

$$
\begin{array}{rcl}
X_{1,N} & = & p_N(T)^{-1} \\
X_{N,N} & = & p_N(T)^{-1}\, p_{N-1}(T)
\end{array}
$$

and somewhat more complicated expressions for $X_{N,1}$ and $X_{1,1}$. Since T and thus B are complex symmetric, however, we also obtain

$$
\begin{array}{rcl}
X_{1,N} & = & X_{N,1} \\
X_{N,N} & = & X_{1,1}.
\end{array}
$$

The definition of the polynomials p_k in (7) determine these to be the second kind Chebyshev polynomials on the interval $[-1, 1]$, of which the zeros are known. In particular,

$$
p_k(T) = \prod_{j=1}^{k} \left(T - \lambda_j^{(k)} I \right)
$$

where

$$
\lambda_j^{(k)} = 2 \cos \frac{j\pi}{k+1}.
$$

Thus, using a technique described in [**11**], multiplication of a vector by one of the matrices $X_{1,1}$ or $X_{1,N}$ can be performed by solving a sequence of N tridiagonal systems. Thus, the Schur complement can be applied in $O(N^2)$ operations rather than the $O(N^2 \log N)$ operations necessary to perform the fast solve to apply B^{-1} in (3).

3.4. Another low-rank modification. Another preconditioner to the system (1) is obtained by writing $\tilde{A}$ as

$$\tilde{A} = \left[\begin{array}{cccccc|cc} (T-I) & -I & & & & & & \\ -I & T & -I & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & -I & T & -I & & & \\ & & & -I & (T-I) & & & \\ \hline & & & & & & \tilde{T}-I & \\ & & & & & & & \tilde{T}-I \end{array}\right] + UU^T$$

where

$$U^T = \left[\begin{array}{cccc|cc} -I & 0 & \cdots & 0 & I & 0 \\ 0 & \cdots & 0 & -I & 0 & I \end{array}\right] \in \mathbb{C}^{2(N+2)\times(N+2)^2}.$$

Denoting $\tilde{A} - UU^T$ as M_{3b}, this yields a preconditioner that can be applied using a fast solver with rank difference $4(N+2)$ to $\tilde{A}$. It can be interpreted as introducing normal derivatives similarly to Section 3.2.1.

3.5. Using the '2-cyclic' property. If we look again at the line-reordered system from Section 3.3.1 preconditioned with M_3, we note that it has a special structure, i.e.

$$M_3^{-1}\tilde{A} = \begin{bmatrix} I & B^{-1}C \\ D^{-1}C^T & I \end{bmatrix} =: \begin{bmatrix} I & B_1 \\ B_2 & I \end{bmatrix}.$$

Matrices of this type are known as 'weakly cyclic of index 2' or to 'possess property A' [**12**], which we will denote as 'being 2-cyclic' for short. For Krylov subspace methods such as CG or QMR, this property can be exploited to save roughly half the work necessary per iteration step. To see this, we look at the recurrence relation for the right Lanczos vectors

$$\tilde{\mathbf{v}}_{n+1} = A\mathbf{v}_n - \alpha_n \mathbf{v}_n - \beta_n \mathbf{v}_{n-1}.$$

Assuming now that two pairs of consecutive left and right Lanczos vectors possess a zero structure given by

$$\mathbf{v}_n = \begin{bmatrix} \mathbf{v}_n^{(1)} \\ 0 \end{bmatrix}, \mathbf{v}_{n-1} = \begin{bmatrix} 0 \\ \mathbf{v}_{n-1}^{(2)} \end{bmatrix}, \mathbf{w}_n = \begin{bmatrix} \mathbf{w}_n^{(1)} \\ 0 \end{bmatrix}, \mathbf{w}_{n-1} = \begin{bmatrix} 0 \\ \mathbf{w}_{n-1}^{(2)} \end{bmatrix},$$

this results in

$$\alpha_n = 1 \quad \text{and} \quad \mathbf{v}_{n+1} = \begin{bmatrix} 0 \\ \mathbf{v}_{n+1}^{(2)} \end{bmatrix},$$

a corresponding relation holding for the $n+1$-st left Lanczos vector $\mathbf{w}_{n+1}$. Thus, in the matrix-vector and inner products occuring in the QMR iteration, this zero structure of the Lanczos vectors can be exploited to reduce the number of arithmetic operations by one half. The only restriction is for the initial residual $\mathbf{r}_0$ to be of the form

$$\mathbf{r}_0 = \begin{bmatrix} * \\ 0 \end{bmatrix},$$

which is easily achieved by choosing the upper block of the initial guess $\mathbf{x}_0^{(1)}$ arbitrarily and setting the lower block to be

$$\mathbf{x}_0^{(2)} = \mathbf{b}^{(2)} - B_2 \mathbf{x}_0^{(1)},$$

where the right hand side $\mathbf{b}$ is again partitioned as

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \end{bmatrix}.$$

4. Numerical results

We have done numerical experiments to get an empirical impression of the effectiveness of the preconditioners described in the preceding sections, whose design was based purely on heuristic considerations. In order to obtain a smooth problem on which to test the algorithm, the forcing term in the Helmholtz equation was chosen to be $f(x, y) \equiv 1$. The wave number k in the Helmholtz equation was chosen depending on the mesh size so as to produce 2 waves in the unit square. In each case, the iteration was continued until the initial residual had been reduced by a factor of 10^{-6}. The iterative method used here was the QMR method due to Freund and Nachtigal. No look-ahead was performed, as no breakdowns were ever observed in these problems.

In the following tables, the iteration counts as well as the execution times in seconds for the preconditioned QMR iteration applied to our problem are tabulated. The calculations were performed on an SGI 4D/35 workstation using double precision FORTRAN 77. The fast solves were all performed using a variation of the CBLKTR routine from Swarztrauber and Sweet's FISHPAK collection available from netlib. In both tables, N denotes the number of mesh points in each direction including the boundary points. The linear system solved is thus of the size $N^2 \times N^2$. The preconditioners referred to in Table 1 are as follows: preconditioner M_1 replaces the radiation condition with a zero Neumann condition as in Section 3.2.1, preconditioner M_2 does the same with a zero Dirichlet condition and preconditioner M_3 is the line reordering described in Section (3.3.1).

In Table 2, preconditioner 3a is the Schur complement preconditioned with the matrix D from Section 3.3.1. Here, a fast solver is applied at every iteration step to perform the multiplication with B^{-1}. Preconditioner M_{3b} is the preconditioner described in 3.4 and M_3^{tc} denotes the the 2-cyclic iteration. Finally, M_3^{fs} denotes the iteration with the Schur complement using the fast application involving the Chebyshev polynomials to perform the matrix-vector multiplications

N	unpreconditioned		M_1		M_2		M_3	
10	15	(0)	6	(0.1)	6	(0.2)	11	(0)
20	40	(0.6)	7	(2)	8	(1.4)	13	(2)
30	63	(2.3)	7	(3)	8	(3.4)	15	(6)
40	84	(6)	7	(7)	9	(8.3)	16	(13)
50	106	(11)	7	(11)	10	(15)	18	(25)
60	127	(19)	7	(15)	11	(24)	19	(38)
70	148	(31)	7	(24)	10	(40)	20	(64)
80	170	(47)	7	(33)	10	(54)	22	(89)
90	190	(67)	7	(40)	11	(72)	23	(120)
100	212	(92)	7	(53)	13	(94)	24	(165)
110	256	(134)	7	(64)	13	(114)	24	(201)
120	281	(175)	7	(80)	14	(153)	25	(261)
130	303	(222)	7	(118)	14	(219	26	(407)
140	328	(286)	7	(138)	17	(317)	27	(496)
150	350	(341)	7	(163)	15	(348)	27	(592)
160	374	(427)	7	(199)	15	(414)	28	(710)
170	427	(549)	7	(235)	16	(493)	29	(873)
180	447	(645)	7	(272)	17	(602)	29	(1088)
190	473	(736)	7	(306)	16	(646)	30	(1181)
200	500	(864)	7	(368)	16	(787)	30	(1287)
210	491	(934)	8	(485)	17	(888)	31	(1557)
220	548	(1144)	7	(495)	18	(1067)	31	(1776)
230	574	(1310)	7	(523)	16	(1232)	35	(2388)
240	599	(1490)	7	(585)	18	(1374)	37	(1489)
250	621	(1676)	7	(626)	18	(1499)	37	(1895)
260	647	(2067)	7	(781)	18	(1881)	33	(3275)

TABLE 1. The unpreconditioned iteration and using preconditioners M_1, M_2 and M_3

as described in 3.3.1. This includes the fast solves for forming the right hand side of the Schur complement as well as another fast solve after the iteration to recover the values of the solution in the interior.

These results indicate that the most promising approach for obtaining mesh independent iteration counts appears to be the operator oriented preconditioner M_1. However, even though the iteration count in this case was indeed constant for all tested mesh sizes, the time for the whole calculation is reduced only to about one third compared with the unpreconditioned iteration for the largest problem. Preconditioner M_2 performs similarly to the first but not quite as well.

The preconditioners based on domain decomposition-type reorderings such as Preconditioner M_3 and M_{3b}, even when the 2-cyclic property is exploited, are not competitive with the former both with regard to overall execution time as well as to how the iteration count increases with the number of unknowns. Preconditioner M_{3b}, however, also appears to be mesh independent, having a somewhat higher iteration count than M_1, which suggests that normal derivatives seem to approximate this particular radiation boundary condition especially well.

The fast Schur complement iteration has the lowest overall execution time, while the iteration count still increases strongly with decreasing mesh size. This is also the case in which the domain decomposition idea is followed most consequently, as the global fast solve in every iteration step is avoided.

References

[1] B.L. Buzbee and F.W. Dorr *The discrete solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions* SIAM J. Num. Anal. **11** pp. 753–763 (1974)

[2] B.L. Buzbee, F.W. Dorr, J.A. George and G.H. Golub *The direct solution of the discrete Poisson equation on irregular regions* SIAM J. Num. Anal. **8** pp. 722–736 (1971)

[3] B. L. Buzbee G. Golub and C. W. Nielson *On direct methods for solving Poisson's equation*, SIAM J. Num. Anal. **7** pp.627–656 (1970).

[4] A. Bayliss, M. Gunzburger and E. Turkel *Boundary conditions for the numerical solution of elliptic equations in exterior regions* SIAM J. Appl. Math **42** pp. 430-451 (1982)

[5] R. Courant and D. Hilbert *Methods of Mathematical Physics* vol. II, Interscience, 1962

[6] Q.V. Dinh, R. Glowinski, J. He, V. Kwock, T.W. Pan and J. Periaux *Lagrange Multiplier Approach to fictitious domain methods:Application to Fluid Dynamics and Electro-Maghnetics* Proc. SIAM 5th Domain Decomposition Methods Conference, May 1991, Norfolk

[7] R.W. Freund and N.M. Nachtigal *QMR: a quasi-minimal residual method for non-Hermitian linear systems* Numer. Math. **60** pp. 315–339 (1991)

[8] R. Forsythe and W. Wasow *Finite Difference Methods for Partial Differential Equations* Wiley, 1960

[9] W. Proskurowski and O. Widlund *On the numerical solution of Helmholtz' equation by the capacitance matrix method* Math. Comp. **20** pp. 433-468 (1976)

[10] P. N. Swarztrauber *The method of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle* SIAM Rev. **19** pp.490-501 (1977)

[11] R. A. Sweet *A cyclic reduction algorithm for block tridiagonal systems of arbitrary dimension* SIAM J. Num. Anal. **14** pp. 706–720 (1977)

[12] R.S. Varga *Matrix Iterative Analysis*, Prentice Hall, New York, 1962

N	M_{3a}		M_{3b}		M_3^{tc}		M_3^{fs}	
10	5	(0)	6	(0)	11	(0)	5	(0)
20	8	(1)	8	(1)	13	(1)	8	(1)
30	8	(3)	8	(3)	15	(3)	8	(1)
40	10	(8)	9	(8)	16	(7)	10	(3)
50	11	(16)	9	(13)	18	(13)	11	(5)
60	13	(25)	10	(20)	19	(20)	13	(9)
70	14	(44)	10	(33)	21	(36)	14	(13)
80	14	(56)	10	(42)	24	(51)	14	(17)
90	16	(83)	11	(60)	23	(64)	16	(24)
100	17	(116)	11	(78)	24	(88)	17	(32)
110	16	(132)	11	(96)	25	(112)	16	(37)
120	18	(182)	11	(119)	25	(141)	18	(51)
130	19	(292)	12	(195)	26	(216)	19	(67)
140	20	(363)	12	(228)	27	(273)	20	(87)
150	20	(404)	12	(277)	27	(291)	20	(93)
160	21	(494)	11	(306)	28	(388)	21	(118)
170	22	(659)	13	(428)	29	(501)	22	(143)
180	22	(773)	13	(471)	29	(556)	22	(164)
190	23	(893)	12	(494)	30	(556)	23	(193)
200	24	(1031)	12	(601)	31	(682)	24	(220)
210	24	(1374)	12	(699)	31	(946)	24	(268)
220	25	(1418)	12	(821)	31	(1086)	25	(321)
230	25	(1721)	12	(960)	35	(1427)	25	(356)
240	26	(1381)	12	(1033)	37	(1445)	26	(407)
250	26	(2081)	12	(1124)	37	(1826)	26	(449)
260	27	(2576)	12	(1418)	33	(1675)	27	(521)

TABLE 2. Preconditioned Schur complement, M_3b, the 2-cyclic iteration and the fast Schur complement

Oliver Ernst, Scientific Computing & Computational Mathematics Program, Stanford University, Stanford, CA 94305, USA
E-mail address: ernst@sccm.stanford.edu

Gene H. Golub, Scientific Computing & Computational Mathematics Program, Stanford University, Stanford, CA 94305, USA
E-mail address: golub@sccm.stanford.edu