Contemporary Mathematics
Volume 180, 1994

Numerical Treatments for the Helmholtz Problem by Domain Decomposition Techniques

Seongjai Kim

Abstract. A parallelizable iterative procedure based on nonoverlapping domain decomposition techniques for numerical solution of the Helmholtz problem in a bounded domain is discussed. An automatic efficient strategy for choosing the algorithm parameter is demonstrated. Numerical results are reported.

1. Introduction

Consider the (complex–valued) scalar Helmholtz problem

$$\begin{aligned} -\frac{\omega^2}{c(x)^2}u - \Delta u &= f(x), \quad x \in \Omega, \\ i\frac{\omega}{c(x)}u + \frac{\partial u}{\partial \nu} &= 0, \qquad x \in \Gamma, \end{aligned} \tag{1.1}$$

where $\Omega \subset \mathbb{R}^d$, $d \leq 3$, is a bounded domain with a Lipschitz boundary $\Gamma = \partial\Omega$, the coefficient $c(x)$ denotes the wave speed and is bounded below and above by positive constants c_0 and c_1, respectively, ν is the outer unit normal to Γ, and the angular frequency $\omega > 0$. The second equation of (1.1) represents first–order absorbing boundary condition that allows normally incident waves to pass out of Ω transparently.

The Helmholtz problem appears difficult to solve. In addition to having a complex–valued solution, the problem (1.1) is neither Hermitian symmetric nor coercive; as a consequence, most standard iterative methods either fail to converge or converge so slowly as to be impractical. The question to be treated in this paper is that of finding the numerical solution of (1.1) in an effective and computationally efficient fashion. We shall define a parallelizable domain decomposition iterative procedure and indicate an efficient strategy of choosing iteration parameters.

1991 *Mathematics Subject Classification.* Primary 65N55, 65F10; Secondary 65N06.
This paper is in final form and no version of it will be submitted for publication elsewhere.

0271-4132/94 $1.00 + $.25 per page

Concerning the iterative numerical solvers for the Helmholtz problem, we refer to Bayliss, Goldstein and Turkel [1] for the preconditioned conjugate–gradient algorithms applied to the normal equations, and Douglas, Hensley and Roberts [2] for an ADI algorithm. The convergency of the strip–type domain decomposition algorithms has been tested by the author [3].

An outline of the paper is as follows. In §2 the domain decomposition algorithm is defined and the iterative procedure using the Robin–type interface condition is illustrated for finite difference approximate solution of the problem (1.1). In §3, an automatic efficient strategy for finding the algorithm parameter is presented. Some experimental results are reported in §4. The last section indicates the conclusions and possible applications.

2. Domain decomposition iterative procedure

Let $\{\Omega_j,\ j=1,\ldots,M\}$ be a partition of Ω:

$$\overline{\Omega} = \cup_{j=1}^{M} \overline{\Omega}_j; \quad \Omega_j \cap \Omega_k = \emptyset, \quad j \neq k.$$

Assume that Ω_j, $j = 1, 2, \cdots, M$, is convex. In practice, each Ω_j would be a rectangular or cubic region. Let

$$\Gamma_j = \Gamma \cap \partial\Omega_j, \quad \Gamma_{jk} = \Gamma_{kj} = \partial\Omega_j \cap \partial\Omega_k, \quad \Sigma = \cup_{j,k=1}^{M} \Gamma_{jk}.$$

Let us consider the decomposition of the problem (1.1) over $\{\Omega_j\}$. The problem (1.1) is equivalent to the following: Find u_j, $j = 1, \ldots, M$, such that

$$\begin{aligned} -\frac{\omega^2}{c^2} u_j - \Delta u_j &= f(x), \quad x \in \Omega_j, \\ i\frac{\omega}{c} u_j + \frac{\partial u_j}{\partial \nu} &= 0, \quad x \in \Gamma_j, \\ \frac{\partial u_j}{\partial \nu_j} + i\beta u_j &= -\frac{\partial u_k}{\partial \nu_k} + i\beta u_k, \quad x \in \Gamma_{jk}, \end{aligned} \tag{2.1}$$

where ν_j is the outwards normal to Ω_j, the consistency conditions are replaced by the Robin interface boundary condition, see [6, 3]. The algorithm parameter β is a complex function on Σ with $\mathrm{Re}(\beta) > 0$, for which the subproblems in (2.1) are well–posed. The problem (2.1) is very interesting from a computational point of view; we do not know of any convergence analysis for the problem.

Let $\Omega \subset \mathbb{R}^2$ be composed of M nonoverlapping rectangular regions with the interface edges parallel to either coordinate axes. Let δ_x^2 denote the centered second order difference with respect to x, and ∂_ν, $\partial_{\mathfrak{f}}$ and $\partial_{\mathfrak{b}}$ be the centered, forward and backward first differences, respectively, in the direction of the outer normal (here, an exterior bordering of the domain is assumed). Let $\Delta_h = \delta_x^2 + \delta_y^2$. Then, one proper finite difference approximation to (2.1) for

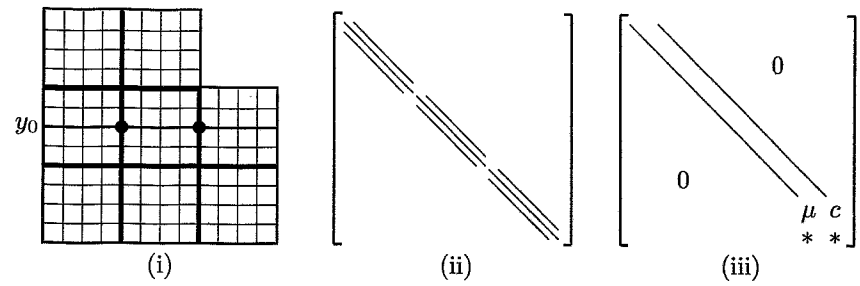

Fig. 3.1: (i). *A decomposition of the domain Ω with mesh lines.* (ii). *The tridiagonal system for the restricted one–dimensional problem on Ω^{y_0}.* (iii). *The matrix U of the LU–factorization performed up to the $(m-1)$–th row.*

two–dimensional problems can be defined by

$$
\begin{aligned}
-\omega^2 u_j^n - \Delta_h u_j^n &= f, \quad x \in \Omega_j, \\
i\omega u_j^n + \partial_\nu u_j^n &= 0, \quad x \in \Gamma_j, \\
\partial_{\mathbf{f}}\, u_j^n + i\beta\, u_j^n &= \partial_{\mathbf{b}}\, u_k^{n-1} + i\beta\, u_k^{n-1}, \quad x \in \Gamma_{jk}.
\end{aligned} \tag{2.2}
$$

Note that the Robin boundary condition is approximated by a combination of forward–backward differences. This combination is very necessary for both convergence and efficience, and the second–order approximation of the (centered) five point finite difference scheme would not be destroyed. For each subdomain Ω_j, only the subdomains sharing an edge as an interior interface boundary are considered as the adjacent subdomains Ω_k.

3. ADOP: *Alternating Direction Optimal Procedure*

In this section we present a heuristic, automatic method of finding efficient algorithm parameter β for general coefficient problems. Consider an L–shape domain and its domain decomposition depicted in Fig. 3.1 (i). There the bold lines denote the interfaces. We shall determine β, line by line, by using horizontal or vertical mesh lines.

Let us find the values of β on the dotted points along the line $y = y_0$. Ignoring the term u_{yy}, we restrict our problem to the one–dimensional subspace $\Omega^{y_0} := \{(x,y) : y = y_0\}$ decomposed into three subdomains with two dotted points being the interfaces. If the points in Ω^{y_0} are ordered from left to right, one iteration for the restricted problem can be performed by inverting a block diagonal matrix (three blocks), where each block is a tridiagonal matrix of dimension, say, m, see Fig. 3.1 (ii). First, consider the tridiagonal block corresponding to the left–end subdomain. When the LU–factorization in which the diagonal elements of L are 1 is performed up to the $(m-1)$–th row, the matrix U can be expressed like in Fig. 3.1 (iii). This factorization is possible

even if the block has not been completely assembled on the interface point. We choose β satisfying

$$-\frac{c}{\mu} = 1 - i\beta h, \tag{3.1}$$

on the first interface point. By using this β, one can complete not only the last row of the first block but also the first row of the next block corresponding to the mid–subdomain. Now, we consider the second tridiagonal block. After performing LU–factorization up to the $(m-1)$–th row, choose β for the second interface point as in the first case. For each mesh line having interface points, including both the horizontal and the vertical mesh lines, this searching can be continued. This procedure is readily applicable to the multiple decompositions of more general domains, clearly. This procedure is developed in [4] and referred to ADOP (alternating direction optimal procedure):

LEMMA 3.1 ([4]). *Let G be the iteration matrix of the one–dimensional problem of* (2.2) *restricted on Ω^{y_0}, with the parameter β found by ADOP. Then the spectral radius of G is zero, i.e., $\rho(G) = 0$.*

The above lemma implies ADOP seeks the parameter β in such a way that the spectral radii of the iteration matrices of the one–dimensional alternating direction problems are zero. It should be noticed that ADOP is automatic and non–expensive. In the next section, efficiency of ADOP will be numerically checked.

4. Numerical results

This section reports some experimental data for the algorithm (2.2) with the parameter β founded by ADOP. In [4], it is numerically checked that the ADOP parameters introduce a faster convergence than any other constant parameters. The computation is performed in complex double precision on an IBM RS/6000, a serial machine. Let $\Omega = (0,1)^2$. For the results reported in this section, the source function f is selected such that the true solution $u(x,y) = \dfrac{\phi(x)\cdot\phi(y)}{\omega^2}$, where $\phi(x) = e^{i\omega(x-1)} + e^{-i\omega x} - 2$. Zero initial values are assumed. Each subproblem in the algorithm (2.2) is solved directly. The errors are estimated on the relative L^∞-error $r_\infty^n = \dfrac{\|U^n - u\|_{L^\infty(\Omega)}}{\|u\|_{L^\infty(\Omega)}}$, where U^n is the approximate solution of the n–th iteration. For the stopping criterion, $\dfrac{\|U^n - U^{n-1}\|_{L^\infty(\Omega)}}{\|U^n\|_{L^\infty(\Omega)}} \le 10^{-4}$ is used. We choose three different typical functions for $c(x,y)$:

$$\begin{aligned} c_1(x,y) &= 1, \\ c_2(x,y) &= 1 + 2x^3 + y, \\ c_3(x,y) &= e^{xy}(2 - \sin(2\pi x))(2 + \sin(4\pi y)). \end{aligned} \tag{4.1}$$

In Table 4.1, iteration counts n and the error r_∞^n are presented for various

$M_x \times M_y$	1×1	2×1	4×1	8×1	16×1	32×1	64×1
n	–	10	10	21	68	147	200
r_∞^n	.048	.048	.048	.049	.048	.049	.048

Table 4.1: *Strip decompositions, when* $\omega = 25$, $c = c_1$ *and* $h = 1/64$.

$M_x \times M_y$	1×1	2×1	5×1	10×1	15×1	20×1	30×1
n	–	21	34	80	75	95	123
r_∞^n	.026	.026	.026	.026	.026	.026	.026
time(sec)	88.0	46.0	26.6	40.3	35.2	42.5	54.1

Table 4.2: *Strip decompositions, when* $\omega = 40$, $c = c_2$ *and* $h = 1/120$.

$M_x \times M_y$	4×1	4×4	8×1	8×8	16×1	16×4	16×16
N	343	419	544	742	864	861	1167
$r_{\infty\rho\infty}$	.0265	.0265	.0265	.0264	.0264	.0263	.0264

Table 4.3: *Total iteration numbers* N *for solving 100 time steps of the problem* (4.2) *and the error* $r_{\infty\rho\infty}$, *when* $\omega = 25$, $c = c_3$, $h = 1/64$ *and* $\Delta t = 1/200$.

strip domain decompositions $M_x \times M_y$, when $\omega = 25$, $c = c_1$, and $h = 1/64$.

Table 4.2 shows iteration counts n, the error r_∞^n and the CPU–time(second), when $\omega = 40$, $c = c_2$ and $h = 1/120$. For these two examples, constant parameters cannot be used, and the standard iterative methods (relaxations and extrapolations) do not converge.

Next, we consider the following time–discretized Schrödinger problem:

$$
\begin{aligned}
i\frac{u^m - u^{m-1}}{\Delta t} - \frac{\omega^2}{c(\mathbf{x})^2}u^m - \Delta u^m &= f(\mathbf{x}, t^m), \quad \mathbf{x} \in \Omega, \\
i\frac{\omega}{c(\mathbf{x})}u^m + \frac{\partial u^m}{\partial \nu} &= 0, \qquad\quad \mathbf{x} \in \Gamma, \\
u^0(\mathbf{x}) &= u_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega,
\end{aligned} \tag{4.2}
$$

where $\mathbf{x} = (x, y)$ and $t^m = m\,\Delta t$ for some $\Delta t > 0$. To check the error propagation, we choose the true solution $u(x, y, t) = (1.5 + \sin(\pi t))\,\phi(x)\,\phi(y)/\omega^2$.

Table 4.3 indicates the total iteration counts N to solve the first 100 time steps of the problem (4.2) by using the domain decomposition method presented in §§2–3 and and the errors

$$
r_{\infty\rho\infty} = \max_{1 \le m \le 100} \frac{\|U^m - u^m\|_\infty}{\|u^m\|_\infty},
$$

where U^m is the approximate solution at $t = t^m$, when $\omega = 25$, $c = c_3$, $h = 1/64$ and $\Delta t = 1/200$. The average iteration counts for solving the problem of one time step can be obtained by dividing by 100. When on each time step an one–domain direct solver is used, we have the error $r_{\infty\rho\infty} = .0265$.

5. Conclusions

We have defined a nonoverlapping domain decomposition iterative procedure for the (complex–valued) Helmholtz problem in the finite differences framework. By a combination of forward–backward finite differences, the Robin interface boundary condition is approximated. An effective strategy for finding the algorithm parameters ADOP is introduced and the effectiveness is numerically tested. In addition to being effective, ADOP is automatic as a preprocessor and its cost is never expensive.

When an iterative (domain decomposition) algorithm is designed, iteration parameters are often introduced to accelerate the convergence speed of the iteration. For certain model problems, the parameters can be selected easily and effectively. However, the problem of choosing iteration parameters for a realistic problem may not be so simple. ADOP is proposed as an answer to the problem of choosing iteration parameters.

For the problem (1.1), it is numerically checked that if $\max(\frac{\omega}{c})h \leq \frac{1}{4}$ and $M_x \leq \frac{1}{4h}$, the procedure ADOP leads to convergence for strip domain decompositions. A numerical example for Schrödinger equation is added. From the example, one can expect ADOP will be more useful for singularly perturbed problems such as, e.g., second order time–dependent partial differential equations. When the wave speed c in (1.1) is complex–valued with $\mathrm{Re}(c) > 0$ and $\mathrm{Im}(c) > 0$, the convergence of the iterative algorithm (2.2) can be analyzed [5].

References

[1] A. Bayliss, C. Goldstein and E. Turkel, *An iterative method for the Helmholtz equation*, *J. Comput. Phys.* 49 (1983), 443–457.

[2] J. Douglas, Jr., J. L. Hensley, and J. E. Roberts, *An alternating–direction iteration method for Helmholtz problems* (Technical Report #214, Mathematics Department, Purdue University, W. Lafayette, IN 47907, 1993).

[3] S. Kim, *A parallelizable iterative procedure for the Helmholtz problem*, to appear in Appl. Numer. Math. (1994).

[4] S. Kim, *ADOP: an automatic strategy for finding efficient iteration parameters in domain decomposition algorithms* , submitted for publication.

[5] S. Kim, *Domain decomposition method and parallel computing: finite elements with mass lumping for scalar waves*, submitted for publication.

[6] P.L. Lions, *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, in: T.F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, Ed., *Third International Symposium on Domain Decomposition Method for Partial Differential Equations*, SIAM, Philadelphia (1990), 202–223.

DEPARTMENT OF MATHEMATICS
PURDUE UNIVERSITY, W. LAFAYETTE, IN 47907
E–mail address: skim@math.purdue.edu