**Studies in Mathematical Sciences**
**Vol. 2, No. 1, 2011, pp. 107-114**
**www.cscanada.org**

**ISSN 1923-8444 [Print]**
**ISSN 1923-8452 [Online]**
**www.cscanada.net**

# Preconditioners for Indefinite Linear System from the Helmholtz Equation

**Shiliang WU[1,*]**
**Cuixia LI[2]**

**Abstract:** Using the finite difference method to discretize the Helmholtz equation usually leads to a large spare linear system. Since the coefficient matrix of the linear system is frequently indefinite, it is difficult to solve iteratively. The approach taken in this paper is to precondition this linear system with SSOR and ILU preconditioners and then to solve it iteratively by using Krylov subspace method such as LSQR method. Numerical experiments are given in order to demonstrate the efficiency of the presented preconditioners.

**Key Words:** Helmholtz equation; Krylov subspace method; Preconditioner

## 1. INTRODUCTION

In computational electromagnetics and seismology, the finite difference method is one of the most effective and popular techniques. The finite difference method has many important applications such as time-harmonic wave propagations, scattering phenomena arising in acoustic and optical problems. More information about applications of this method in electromagnetics can be found in [1–3].

In this paper, for the sake of simplicity our main interest lies in solving the following form of the Helmholtz equation:

$$\Delta u + k^2 u = f, \quad \Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}], \tag{1}$$

$$f = \delta(x - \frac{1}{2})\delta(y), \qquad x = [x_{\min}, x_{\max}],\ y = y_{\min}, \tag{2}$$

$$u = 0, \qquad\qquad y = y_{\min}, \tag{3}$$

$$\frac{\partial u}{\partial n} - iku = 0, \qquad x = x_{\min}, x_{\max},\ y = y_{\max}, \tag{4}$$

where $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ is Laplace operator. $\Omega$ is a bounded region in $\mathbb{R}^2$. $k \in \mathbb{R}$ is the wave-number and depends on the spatial position in the domain. (1.2) is a radiation boundary condition and is one kind of the first-order Sommerfeld condition. $n$ stands for an outward direction normal to the boundary. The above-mentioned Helmholtz equation is also considered by Erlangga [2], which is so-called "open problem": Outgoing waves penetrate at least one boundary without (spurious) reflections. The Helmholtz equation

[1]College of Mathematics, Chengdu University of Information Technology, Chengdu, 610255, China.
E-mail addresses: wushiliang1999@126.com; slwu@aynu.edu.cn.
[2]School of Mathematics and Statistics, Anyang Normal University, Anyang, 455002, China.
E-mail address: lixiatk@126.com.
[*]Corresponding author.
[†]Received 1 January 2011; accepted 6 February 2011.

(1) often governs wave propagations and scattering phenomena arising in acoustic problems in many areas, such as aeronautics, marine technology, geophysics and optical problems.

Using the finite difference scheme to discretize the Helmholtz equation usually leads to the large spare linear system. As is known, there exist two methods employed to solve the linear system: direct methods and iterative methods. Direct methods are widely employed when the order of the coefficient matrix is not too large, and are usually regarded as robust methods. The memory and the computational requirements for solving the large linear systems may seriously challenge the most efficient direct solution method available today.

The alternative is to use iterative methods established for solving large linear systems. Naturally, it is necessary that we make use of iterative methods instead of direct methods to solve the large sparse linear systems. Meanwhile, iterative methods are easier to implement efficiently on high performance computers than direct methods. Currently, Krylov subspace methods are considered as one kind of the important and efficient iterative techniques available for solving large linear systems because the methods are cheap to be implemented and are able to exploit the sparsity of the coefficient matrix. However, in fact, Krylov subspace methods are not competitive without a good preconditioner. In this paper, SSOR and ILU perconditioners are presented to improve the convergence of Krylov subspace methods for solving the Helmholtz equation.

A great deal of effort has been contributed to the development of the powerful preconditioners for the Helmholtz equation. Generally speaking, there are two classes of preconditiones for the Helmholzt equation by observing [1–9]: matrix-based and operator-based. The former is based on an approximation of the inverse of the coefficient matrix of the linear system (such as incomplete LU (ILU) factorizations), the latter is found on the operator for which the spectrum of preconditioned system is clustered strongly (such as shifted Laplace, Analytic ILU (AILU)). One can refer to [1–3, 10] for more details.

By observing [2], it is easy to find that the first-order derivative (1.2) is disretized with the first-order forward scheme. In this paper, the first-order derivative discretized is different from [2] and the approximation of (1) is improved (see Section 2). In the light of the preconditioning idea, this paper is devoted to giving SSOR and ILU preconditioners for the nonsymmetric indefinite linear system.

The remainder of this paper is organized as follows. In Section 2, the discretization approach of (1) is presented in detail. Dealing with the first-order derivative is different from [2]. In Section 3, the SSOR and ILU preconditioners are given when the LSQR method is employed to solve the resulting linear sytem. In Section 4, numerical experiments are presented to confirm the efficiency of the SSOR and ILU preconditioners. Finally, in Section 5 some conclusions are drawn.

## 2. DISCRETIZATION APPROACH

To conveniently find numerical solutions of (1), the equation is discretized with the second-order difference scheme, in $x$-direction:

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{h^2}(u_{i-1} - 2u_i + u_{i+1}) + O(h^2), \tag{5}$$

and similarly in $y$-direction with a constant mesh spacing $h$ in both directions. In [2], Erlangga made use of the first-order forward scheme to discretize the first-order derivative in (1.2), that is,

$$\frac{\partial u}{\partial n} = \frac{1}{\Delta n}(u_{i-1} - u_i). \tag{6}$$

However, it is known a significant point that making use of the first-order forward scheme instead of the first-order derivative leads to the result: truncation error of the boundary points is lower than that of the internal points. To remedy for the approach above and increase the approximation of (1), here we deal with (1.2) as follows:

(1) For $x = x_{\min}$, then

$$\frac{u(0,j) - u(1,j)}{h} - iku(1/2,j) = 0. \tag{7}$$

To improve the approximation of the radiation condition, one simply sets

$$u(1/2,j) = \frac{u(0,j) + u(1,j)}{2}. \tag{8}$$

Substituting (8) into (7) yields

$$\frac{u(0,j) - u(1,j)}{h} - ik\frac{u(0,j) + u(1,j)}{2} = 0. \tag{9}$$

By simple computations, we get

$$u(0,j) = \frac{2 + ikh}{2 - ikh}u(1,j). \tag{10}$$

(2) Be simialr to (1), we have

$$u(I,j) = \frac{2 + ikh}{2 - ikh}u(I-1,j), \text{ for } x = x_{\max}, \tag{11}$$

$$u(i,J) = \frac{2 + ikh}{2 - ikh}u(i,J-1), \text{ for } y = y_{\max}. \tag{12}$$

(3) For $x = x_{\min}$ and $y = y_{\max}$, we have

$$\frac{u(0,J) - u(1,J-1)}{\sqrt{2}h} - iku(1/2,J-1/2) = 0. \tag{13}$$

If one sets

$$u(1/2,J-1/2) = \frac{u(0,J) + u(1,J-1)}{2}, \tag{14}$$

then we get

$$u(0,J) = \frac{\sqrt{2} + ikh}{\sqrt{2} - ikh}u(1,J-1). \tag{15}$$

(4) Be simialr to (3), we have

$$u(I,J) = \frac{\sqrt{2} + ikh}{\sqrt{2} - ikh}u(I-1,J-1), \text{ for } x = x_{\max} \text{ and } y = y_{\max}. \tag{16}$$

The above-mentioned approach brings about the large sparse linear system

$$Ax = b. \tag{17}$$

It is not difficult to find that matrix $A$ is large spare, but not symmetric by surveying the above-mentioned approach. Obviously, as $k$ is a sufficient large positive number, the matrix $A$ becomes complex-valued, highly indefinite and ill-conditioned.

# 3. SSOR AND ILU PRECONDITIONERS

To improve the rate of convergence for iterative methods, in general, a suitable preconditioner has to be applied. That is, it is often preferable to solve the preconditioned linear system as follows:

$$P^{-1}Ax = P^{-1}b, \tag{18}$$

where $P$, called the preconditioner, is a non-singular matrix. The choice of the preconditioner $P$ is important in actual implements. According to the excellent survey of [11] by Benzi, a good preconditioner should meet the following requirements:

- The preconditioned system should be easy to solve.
- The preconditioner should be cheap to construct and apply.

Of course, the best choice for $P^{-1}$ is the inverse of $A$. However, it is useless in actual implements. To improve the convergence rate of the LSQR method [12, 13] for solving the nonsymmetric indefinite linear system arising from the Helmholtz equation, SSOR and ILU preconditioners are well-loved preconditioning techniques.

## 3.1 SSOR preconditioner

Based on the idea of the chosen preconditioner, the SSOR preconditioner is often considered in actual implements and can be derived from the coefficient matrix without any work. First, a brief review of the classical symmetric SOR iterative method is needed. This method is proposed based on the idea of the SOR iterative method and is taken as the symmetric version of the SOR iterative method. Second, to built the SSOR preconditioner, in general, a matrix $A$ is split as follows:

$$A = D + L + U, \tag{19}$$

in which $D$ is the diagonal part of $A$, $L(U)$ is the strict lower(upper) part of $A$. Then the standard SSOR preconditioner [14] is defined by

$$P = (D + L)D^{-1}(D + U). \tag{20}$$

Obviously, the SSOR preconditioner $P$ is given in factored form and can share many properties of other factorization-based methods. For example, its suitability for vector processors or parallel architectures depends strongly on the ordering of the variables.

## 3.2 ILU preconditioner

As is known, in the case of ILU preconditioner, which is one of the most popular preconditioning techniques, can be constructed by performing Gauss elimination and dropping some elements based on certain constraints. For the common situation $A$ is nonsymmetric and indefinite, standard ILU factorizations maybe fail. For instance, standard ILU is the fatal breakdown due to the encounter of a zero pivot. Ordinarily, in actual complements standard ILU factorizations is taken place of nonstandard ILU factorization. To a general ILU factorization procession, there exist a sparse lower triangular matrix $E$ and a parse upper triangular matrix $F$ such that the residual matrix $R$ satisfies

$$A = EF + R. \tag{21}$$

Multiplying the above equality from left hand by $E^{-1}$ and right hand by $F^{-1}$, we get

$$E^{-1}AF^{-1} = I + E^{-1}RF^{-1}. \tag{22}$$

From (22), the following two results are obtained by Saad in [14]:
(i). When the matrix $A$ is diagonally dominant, then $E$ and $F$ are well conditioned and the size of matrix $E^{-1}RF^{-1}$ remains confined within reasonable limits, typically with a nice clustering of its eigenvalues around the origin in the Cartesian Coordinates.
(ii). When the matrix $A$ is not diagonally dominant, $E^{-1}$ or $F^{-1}$ may have very large norms, causing the error $E^{-1}RF^{-1}$ to be very large and thus adding large perturbations to the identity matrix.

Obviously, ILU factorization is easy to be computed, the work in one iteration is cheap. Although ILU preconditioner may require extra storage due to fill-in and this requirement may exceed that for storing $A$, in our numerical experiments ILU preconditioner with drop tolerance 0.01 is superior to the above-mentioned SSOR preconditioner. Meanwhile, instability scarcely happens in the actual computation of our Helmholtz problems (see Section 4).

# 4. NUMERICAL EXPERIMENTS

In this section, we give numerical experiments to demonstrate the performance of our preconditioning approach on the Helmholtz equation. For convenience, the sign '$M$' denotes the ILU preconditioner.

To solve the linear system (17), iterative methods based on Krylov subspace method are cheap to be implemented and are able to make better use of the sparsity of the coefficient matrix. Since the coefficient matrix of (17) is neither positive definite nor symmetric with $k$ be a sufficient large positive number, the Conjugate Gradient (CG) method [15, 16] may breakdown. In our numerical experiments LSQR with preconditioner in [12, 13] is adopted. The incomplete LU factorization of $A$ is with drop tolerance 0.01.

All tests are started from the zero vector, preformed in MATLAB with machine precision $10^{-16}$, and terminated when the LSQR iteration terminates if the relative residual error satisfies $\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} < 10^{-6}$ or the iteration number is more than 500.

**Table 1:** Iterations and time(s) for LSQR method with 32×32

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 32×32 | | $k$ | 100 | 110 | 120 | 130 | 140 | 150 |
| | P | Iter | 7 | 5 | 4 | 4 | 3 | 3 |
| | | Time(s) | 0.2188 | 0.1563 | 0.1094 | 0.125 | 0.0938 | 0.0938 |
| | M | Iter | 4 | 3 | 4 | 4 | 3 | 3 |
| | | Time(s) | 0.1563 | 0.1406 | 0.1025 | 0.125 | 0.0625 | 0.0638 |

**Example 1**. The following two dimension open homogeneous problem is considered

$$\Delta u + k^2 u = f, \qquad \Omega = [0,1] \times [0,1], \tag{23}$$

$$f = \delta(x - \frac{1}{2})\delta(y), \; x = [0,1], \; y = 0, \tag{24}$$

$$u = 0, \qquad y = 0, \tag{25}$$

$$\frac{\partial u}{\partial n} - iku = 0, \qquad x = 0,1, \; y = 1, \tag{26}$$

with $k$ be constant in $\Omega$. The above-mentioned Helmholtz equation describes an open problem allowing waves to penetrate the boundaries. It is not difficult to find that waves created at the upper surface propagate.

By the above discussion, we solve the resulting linear system with LSQR method and compare the preconditioner $P$ and the preconditioner $M$. Table 1 shows the computational performance in terms of number of iterations (denoted by Iter) and computational time (denoted by Time(s)) to reach the specified convergence with the different mesh.

**Table 2:** Iterations and time(s) for LSQR method with 64×64

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 64×64 | | $k$ | 185 | 190 | 195 | 200 | 205 | 210 |
| | P | Iter | 21 | 12 | 9 | 7 | 6 | 6 |
| | | Time(s) | 2.9219 | 1.7188 | 1.2969 | 1.0156 | 0.9063 | 0.9063 |
| | M | Iter | 6 | 4 | 5 | 4 | 4 | 3 |
| | | Time(s) | 1.5156 | 1.1719 | 0.9531 | 0.7813 | 0.7656 | 0.6094 |

**Table 3:** Iterations and time(s) for LSQR method with 128×128

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 128×128 | | $k$ | 370 | 390 | 410 | 430 | 450 | 470 |
| | P | Iter | 21 | 9 | 6 | 5 | 5 | 4 |
| | | Time(s) | 17.6719 | 8.1406 | 5.6719 | 4.8438 | 4.7969 | 3.9688 |
| | M | Iter | 6 | 5 | 4 | 3 | 3 | 3 |
| | | Time(s) | 10.3125 | 5.4531 | 4.5781 | 3.5313 | 3.4219 | 3.6719 |

From Tables 1-3, it is easy to find that the effectiveness of the preconditioner $M$ outperforms the perconditioner $P$ when the choice of drop tolerance is appropriate.

**Table 4:** Iterations and time(s) for LSQR method with 32×32

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 32×32 | | $k_{ref}$ | 100 | 110 | 120 | 130 | 140 | 150 |
| | P | Iter | 6 | 5 | 4 | 4 | 4 | 4 |
| | | Time(s) | 0.1719 | 0.1406 | 0.125 | 0.125 | 0.125 | 0.1094 |
| | M | Iter | 3 | 3 | 4 | 3 | 3 | 3 |
| | | Time(s) | 0.0938 | 0.1094 | 0.1094 | 0.1094 | 0.0938 | 0.1094 |

**Example 2.** In this example we repeat the computation of Example 1 but now in an in-homogeneous medium. The wave-number varies inside the domain according to

$$k = \begin{cases} k_{ref}, & 0 \le y \le 1/3, \\ 1.5k_{ref}, & 1/3 \le y \le 2/3, \\ 2k_{ref}, & 2/3 \le y \le 1. \end{cases} \tag{27}$$

with $k_{ref}$ be constant. Numerical results are presented in Table 2. LSQR method with the preconditioner P and M is employed to solve the resulting linear system.

In Tables 4-6, we list the computational performance in terms of number of iterations (denoted by Iter) and computational time (denoted by Time(s)) to reach the specified convergence with the different mesh. From Tables 4-6, it is easy to find that the effectiveness of the preconditioner $M$ outperforms the perconditioner $P$ when the choice of drop tolerance is appropriate.

# 5. CONCLUSIONS

In this paper, we make use of the finite difference method to discretize the Helmholtz equation. There is a new approach to deal with the radiation boundary condition. LSQR method with SSOR and ILU preconditioner has been employed to solve the resulting linear system. SSOR and ILU preconditioners are compared in Examples 1 and 2. From our numerical experiments, ILU preconditioner is superior to SSOR

**Table 5:** Iterations and time(s) for LSQR method with 64×64

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 64×64 | P | $k_{ref}$ | 190 | 200 | 210 | 220 | 230 | 240 |
| | | Iter | 8 | 6 | 5 | 5 | 4 | 4 |
| | | Time(s) | 1.2656 | 0.9688 | 0.7813 | 0.7813 | 0.6563 | 0.6875 |
| | M | Iter | 3 | 3 | 3 | 3 | 3 | 3 |
| | | Time(s) | 0.6563 | 0.5781 | 0.5625 | 0.5625 | 0.5938 | 0.5469 |

**Table 6:** Iterations and time(s) for LSQR method with 128×128

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 128×128 | P | $k_{ref}$ | 370 | 390 | 410 | 430 | 450 | 470 |
| | | Iter | 10 | 8 | 5 | 5 | 5 | 4 |
| | | Time(s) | 9.3125 | 7.5625 | 5.8281 | 5.1094 | 4.9688 | 4.2813 |
| | M | Iter | 3 | 3 | 3 | 3 | 3 | 3 |
| | | Time(s) | 4.0156 | 3.75 | 3.7656 | 3.6719 | 3.6875 | 3.7969 |

preconditioner when the choice of drop tolerance is appropriate. At present, the hard work is to discuss the spectrum of the preconditioned matrix and is need to study further in the future, since the coefficient matrix of linear system is large, sparse, nonsymmetric and indefinite.

Of course, multigrid method can be also employed to solve the resulting linear system. Optimal or quasi-optimal multigrid applied to highly indefinite systems runs into serious difficulties. For instance, if the coarse mesh size is not 'small enough', disappointing performances have been observed [17], to say nothing of the hard tasks of defining efficient restriction and/or prolongation operators, especially in the case of unstructured meshes.

# ACKNOWLEDGEMENTS

This research was supported by NSFC Tianyuan Mathematics Youth Fund (11026040, 11026083).

# REFERENCES

[1] Guo, C. H. (1996). Imcomplete block factorization preconditioner for linear systems arising in the numerical solution of the Helmholtz equation. *Appl. Numer. Math., 19*(4), 495-508.

[2] Erlangga, Y. A., Vuik, C., & Oosterlee, C. W. (2004). On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math., 50*(3-4), 409-425.

[3] Bayliss, A., Goldstein, C. I., & Turkel, E. (1983). An iterative method for Helmholtz equation. *J. Comput. Phys., 49*(3), 443-457.

[4] Laird, A. L. (2001). *Preconditioned iterative solution of the 2D Helmholtz equation*. First Year's Report, St. Hugh's College, Oxford.

[5] Made, M. M. M. (2001). Imcomplete factorization-based preconditionings for solving the Helmholtz equation. *Int. J. Numer. Meth. Eng., 50*(5), 1077-1101.

[6] Van Gijzen, M. B., Erlangga, Y. A., & Vuik, C. (2007). Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted laplican. *SIAM J. Sci. Comput., 29*(4), 1942-1958.

[7] Elman, H. C., Ernst, O. G., & O'Leary, D. P. (2001). A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. *SIAM J. Sci. Comput., 23*(4), 1291-1315.

[8] Bollhoefer, M., Grote, M. J., & Schenk, O. (2009). Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media. *SIAM J. Sci. Comput., 31*(5), 3781-3805.

[9] Osei-Kuffuor, D., & Saad, Y. (2010). Preconditioning Helmholtz linear systems. *Appl. Numer. Math., 60*(4), 420-431.

[10] Gander, M. J., & Nataf, F. (2001). AILU for Helmholzt problems: a new preconditioner based on the analytical parabolic factorization. *J. Comput. Acoust., 9*(4), 1499-1506.

[11] Benzi, M. (2002). Preconditioning techniques for large linear systems: A Survey. *J. Comput. Phys., 182*(2), 418-477.

[12] Barrett, R., Berry, M., & Chan, T. F. (1994). *Templates for the solution of linear systems: Building blocks for iterative methods*. Philadelphia: SIAM Press.

[13] Paige, C. C., & Saunders, M. A. (1982). LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Soft., 8*(1), 43-71.

[14] Saad, Y. (2003). *Iterative methods for sparse linear systems* (2nd ed.). Philadelphia: SIAM Press.

[15] Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards., 49*(6), 409-436.

[16] Freund, R. W. (1992). Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Statist. Comput., 13*(1), 425-448.

[17] Bramble, J. H., Pasciak, J. E., & Xu, J. (1988). The analysis of multigrid algorithms for nonsymmetric and indefinite problems. *Math. Comput., 51*(184), 389-414.