Research Article

A Modified SSOR Preconditioning Strategy for Helmholtz Equations

Shi-Liang Wu[1] and Cui-Xia Li[2]

[1] *College of Mathematics, Chengdu University of Information Technology, Chengdu*
   610225, China
[2] *School of Mathematics and Statistics, Anyang Normal University, Anyang*
   455002, China

Correspondence should be addressed to Shi-Liang Wu, wushiliang1999@126.com

The finite difference method discretization of Helmholtz equations usually leads to the large spare linear systems. Since the coefficient matrix is frequently indefinite, it is difficult to solve iteratively. In this paper, a modified symmetric successive overrelaxation (MSSOR) preconditioning strategy is constructed based on the coefficient matrix and employed to speed up the convergence rate of iterative methods. The idea is to increase the values of diagonal elements of the coefficient matrix to obtain better preconditioners for the original linear systems. Compared with SSOR preconditioner, MSSOR preconditioner has no additional computational cost to improve the convergence rate of iterative methods. Numerical results demonstrate that this method can reduce both the number of iterations and the computational time significantly with low cost for construction and implementation of preconditioners.

1. Introduction

The finite difference method is one of the most effective and popular techniques in computational electromagnetics and seismology, such as time-harmonic wave propagations, scattering phenomena arising in acoustic and optical problems, and electromagnetics scattering from a large cavity. More information about applications of this method in electromagnetics can be found in [1–5].

In this paper, we focus on the following form of the complex Helmholtz equation:

$$-\Delta u - pu + iqu = f \quad \text{in } \Omega,$$
$$u = g \quad \text{on } \partial\Omega.$$

(1.1)

Here Ω is a bounded region in $\mathbb{R}^2$. p and q are real continuous coefficient functions on $\overline{\Omega}$, while f and g are given continuous functions on $\overline{\Omega}$ and $\partial\Omega$, respectively.

To conveniently find numerical solutions of (1.1), the Laplace operator is approximated by using the second-order accurate 5-point difference stencil:

$$L_h \doteq \frac{1}{h^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}_h . \tag{1.2}$$

Making use of the above stencil, the following linear system is obtained:

$$\mathcal{A}x = \left(A - h^2 D + ih^2 E\right)x = b, \tag{1.3}$$

where $D(E)$ is a diagonal matrix whose diagonal elements are just the values of $p(q)$ at the mesh points and A is the symmetric positive definite M-matrix arising from the discrete Laplace operator and is of the block tridiagonal form

$$A = \begin{bmatrix} G_1 & F_2 & & \\ E_2 & G_2 & \ddots & \\ & \ddots & \ddots & F_m \\ & & E_m & G_m \end{bmatrix}, \tag{1.4}$$

with

$$G_k = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}_{n_k \times n_k} , \quad k = 1, 2, \ldots, m, \tag{1.5}$$

$$E_k = F_k^T = \begin{cases} \left(O_{n_k \times p_k} \ -I_{n_k} \ O_{n_k \times q_k}\right), & p_k, q_k \geq 0, \\ p_k + q_k = n_{k-1} - n_k & \text{if } n_k \leq n_{k-1}, \\ \left(O_{n_{k-1} \times s_k} \ -I_{n_{k-1}} \ O_{n_{k-1} \times t_k}\right)^T, & s_k, t_k \geq 0, \\ s_k + t_k = n_k - n_{k-1} & \text{if } n_k > n_{k-1}, \end{cases} \quad k = 2, 3, \ldots, m.$$

Obviously, from the linear systems (1.3), it is not difficult to find that the matrix $\mathcal{A}$ is a complex symmetric coefficient matrix. Matrix $\mathcal{A}$ becomes highly indefinite and ill-conditioned as p is a sufficiently large positive number. So, large amount of computation times and memory are needed in order to solve the linear systems (1.3) efficiently.

As is well known, direct methods and iterative methods can be employed to solve the linear systems (1.3). The former is widely employed when the order of the coefficient matrix $\mathcal{A}$ is not too large and is usually regarded as robust methods. The memory and the computational requirements for solving the large sparse linear systems may seriously challenge the most efficient direct solution method available today. Currently, the latter employed to solve the large sparse linear systems is popular. The reason is that iterative methods are easier to implement efficiently on high performance computers than direct methods. In practice, a natural choice is that we make use of iterative methods instead of direct methods to solve the large sparse linear systems.

At present, Krylov subspace methods are considered as one kind of the important and efficient iterative techniques for solving the large sparse linear systems because these methods are cheap to be implemented and are able to fully exploit the sparsity of the coefficient matrix. It is well known that the convergence speed of Krylov subspace methods depends on the distribution of the eigenvalues of the coefficient matrix [6]. When the coefficient matrix is typically extremely ill-conditioned and highly indefinite, the convergence of Krylov subspace methods can be unacceptably slow. In this case, Krylov subspace methods are not competitive without a good preconditioner. That is, preconditioning technique is a key ingredient for the success of Krylov subspace methods in applications. The idea of preconditioning technique is based on consideration of the linear system with the same solution as the original equation. The problem is that each preconditioning technique is suited for a different type of problem. Until current days, no robust preconditioning technique appears for all or at least much types of problems. Finding a good preconditioner to solve a given large sparse linear systems is often viewed as a combination of art and science.

In recent years, a great deal of effort has been invested in solving indefinite linear systems from the discrete Helmholtz equations. Most of the work has been aimed at developing effective preconditioning techniques. In general, there exist two classes of preconditioners for Helmholtz equations: the "operator-based" preconditioning technique and the "matrix-based" preconditioning technique.

The former is built based on an operator, such as the Laplace preconditioner [2, 3, 7–9], Analytic ILU [10], the Separation-of-variables [11]. The purpose of this class of preconditioners is that the spectrum of the corresponding preconditioned matrix is favorably clustered. Its advantage is that this operator does not have to be a representation of the inverse of the Helmholtz operator.

The latter is established based on an approximation of the inverse of the coefficient matrix. For this class, one of the natural and simplest ways of structuring a preconditioner is to employ a diagonal or block diagonal of the coefficient matrix as a preconditioner [12]. The above two diagonal preconditioners have no remarkable reduce with respect to the iterative number and CPU time. Another one of the simplest preconditioners is to perform an incomplete factorization (ILU) of the coefficient matrix [1]. The main idea of ILU factorizations depends on the implementation of Gaussian elimination which is used, see the survey [13] and the related references therein.

When the coefficient matrix of the linear systems (1.1) is complex symmetric and indefinite, it is difficult to solve iteratively. Using the symmetric successive overrelaxation (SSOR) as a preconditioner preserves the symmetry of the iterative matrix and also taking little initialization cost, which in some cases makes it preferable over other factorization methods such as ILU. So far, some variant SSOR preconditioning techniques have been proposed to improve the convergence rate of the corresponding iterative method for solving the linear systems. Mazzia and Alan [14] introduced a shift parameter to develop a shifted

SSOR preconditioner for solving the linear systems from an electromagnetics application. Bai in [15] used a (block) diagonal matrix instead of the diagonal matrix of the coefficient matrix to establish a modified (block) SSOR preconditioner for the second-order selfadjoint elliptic boundary value problems. Chen et al. [16] used a diagonal matrix with a relaxation parameter instead of the diagonal matrix of the coefficient matrix to establish a modified block SSOR preconditioner for the Biot's consolidation equations. Ran and Yuan in [17] also discussed a class of modified (block) SSOR preconditioners for linear systems from steady incompressible viscous flow problems. We refer the reader to [18, 19] for a general discussion.

Although the SSOR preconditioner with Krylov subspace methods can improve convergence improvement, the disadvantage of the SSOR preconditioner is that the convergence rate may still remain unsatisfactorily slow in some cases, especially in indefinite linear systems. In this paper, a modified symmetric successive overrelaxation (MSSOR) preconditioning strategy is presented, which can significantly improve the convergence speed and CPU time. Our motivation for this method arises from the solution of complex symmetric and indefinite linear systems from the discrete Helmholtz equations. The idea is to increase the values of diagonal elements of the coefficient matrix to obtain better preconditioners for the original linear systems, which is different from [14–17]. This modification does not require any significant computational cost as compared with the original SSOR preconditioner and also requires no additional storage cost.

The remainder of this paper is organized as follows. In Section 2, the MSSOR preconditioner for solving the resulting linear system is presented. In Section 3, numerical experiments are given to illustrate the efficiency of the presented preconditioner. Finally, in Section 4 some conclusions are drawn.

2. Modified SSOR Preconditioner

To improve the convergence rate of iterative methods, an appropriate preconditioner should be incorporated. That is, it is often preferable to solve the preconditioned linear system as follows:

$$P^{-1}\mathcal{A}x = P^{-1}b, \tag{2.1}$$

where P, called the preconditioner, is a nonsingular matrix. The choice of the preconditioner P plays an important role in actual implements. In general, the preconditioner P is chosen such that the condition number of the preconditioned matrix $P^{-1}\mathcal{A}$ is less than that of the original matrix $\mathcal{A}$. Based on the excellent survey of [13] by Benzi, a good preconditioner should meet the following requirements:

(1) the preconditioned system should be easy to solve;

(2) the preconditioner should be cheap to construct and apply.

Certainly, the best choice for P^{-1} is the inverse of $\mathcal{A}$. However, it is unpractical in actual implements because the cost of the computation of $\mathcal{A}^{-1}$ may be high. If $\mathcal{A}$ is a symmetric positive definite matrix, the approximation of $\mathcal{A}^{-1}$ can be replaced by SSOR or multigrid. However, in fact, the Helmholtz equation often results in an indefinite linear system, for which SSOR or multi-grid may be not guaranteed to converge.

To introduce the modified SSOR preconditioner, a brief review of the classical and well-known SSOR preconditioner is needed. The SSOR preconditioner is established by

the SSOR iterative method, which is a symmetric version of the well-known SOR iterative method. Based on matrix splitting, the coefficient matrix $\mathcal{A}$ is split as follows:

$$\mathcal{A} = \mathfrak{D} + \mathcal{L} + \mathcal{L}^T, \tag{2.2}$$

where $\mathfrak{D}$ and $\mathcal{L}$ are the diagonal parts and strictly lower triangular of $\mathcal{A}$. According to the foregoing matrix splitting (2.2), the standard SSOR preconditioner [6, 20] is defined by

$$P_{\text{SSOR}} = (\mathfrak{D} + \mathcal{L})\mathfrak{D}^{-1}\left(\mathfrak{D} + \mathcal{L}^T\right). \tag{2.3}$$

It is difficult to show theoretically the behavior of a preconditioner when the coefficient matrix $\mathcal{A}$ is a large, sparse, and symmetric indefinite. The SSOR iterative method is not convergent, but P_{SSOR} may be still used as a preconditioner. By a simple modification on the original indefinite linear systems (1.3), we establish the following coefficient matrix:

$$\overline{\mathcal{A}} = A + h^2 D + h^2 E i = \widetilde{\mathfrak{D}} + \mathcal{L} + \mathcal{L}^T, \tag{2.4}$$

where

$$\widetilde{\mathfrak{D}} = \text{diag}(A) + h^2 D + h^2 E i. \tag{2.5}$$

Obviously, $\overline{\mathcal{A}}$ is a symmetric and positive stable H-matrix. To increase the values of diagonal elements of the coefficient matrix to obtain better preconditioners for the original linear systems and reduce computation times and amount of memory, based on (2.4), the MSSOR preconditioner is defined by

$$P_{\text{MSSOR}} = \left(\overline{\mathfrak{D}} + \mathcal{L}\right)\overline{\mathfrak{D}}^{-1}\left(\overline{\mathfrak{D}} + \mathcal{L}^T\right), \tag{2.6}$$

with

$$\overline{\mathfrak{D}} = \text{diag}\left(\left|\widetilde{\mathfrak{D}}\right|\right), \qquad \left|\widetilde{\mathfrak{D}}\right| = \left|\tilde{d}_{ii}\right| \quad (i = 1, 2, \ldots, n). \tag{2.7}$$

This idea is based on an absolute diagonal scaling technique, which is cheap and easy to implement.

Since the coefficient matrix of the linear systems (1.3) is neither positive definite nor Hermitian with p being a sufficiently large positive number, the Conjugate Gradient (CG) method [21] may breakdown. To solve the complex symmetric linear systems, van der Vorst and Melissen [22] proposed the conjugate orthogonal conjugate gradient (COCG) method, which is regarded as an extension of CG method.

To solve the linear systems (1.3) efficiently, (1.3) is transformed into the following form with the preconditioner P_{MSSOR}, that is,

$$P_{\text{MSSOR}}\left(A - h^2 D + ih^2 E\right)x = P_{\text{MSSOR}}b. \tag{2.8}$$

Then the MSSOR preconditioned COCG (PCOCG) method can be employed to solve the preconditioned linear systems (2.8).

In the following, we give the MSSOR preconditioned COCG method for solving the linear systems (2.8). The MSSOR preconditioned COCG (PCOCG) algorithm is described as follows

Algorithm PCOCG [22]: given an initial guess x_0

(1) $v_0 = b - \mathcal{A}x_0$;

(2) $p_{-1} = 0$; $\beta_{-1} = 0$;

(3) $w_0 = P_{\text{MSSOR}}^{-1}v_0$; $\rho_0 = v_0^T w_0$;

(4) for $i = 0, 1, 2, \ldots$

(5) $p_i = w_i + \beta_{i-1}p_{i-1}$;

(6) $u_i = \mathcal{A}p_i$;

(7) $\mu_i = u_j^T p_j$; if $\mu_i = 0$ then quit (failure);

(8) $\alpha = \rho_i/\mu_i$;

(9) $x_{i+1} = x_i + \alpha p_i$; $v_{i+1} = v_i - \alpha u_i$

(10) if x_{i+1} is accurate enough, then quit (convergence);

(11) $w_{i+1} = P_{\text{MSSOR}}^{-1}v_{i+1}$;

(12) $\rho_{i+1} = v_{i+1}^T w_{i+1}$; if $|\rho_{i+1}|$ too small, then quit (failure);

(13) $\beta_i = \rho_{i+1}/\rho_i$;

(14) End for i.

It is not difficult to find that the main computation of algorithm PCOCG involves one matrix-vector multiplication and two triangular linear systems. These computations are very easy to implement. The main advantage is no extra computational cost in construction of MSSOR preconditioner.

Note that the *transpose* in all dot products in this algorithm is essential [23]. Meanwhile, note that two different breakdowns of this algorithm may occur: one is that if $v_{i+1}^T P_{\text{MSSOR}}^{-1}v_{i+1}$ is too small, but v_{i+1} exists (line 12), algorithm PCOCG breaks down and the other is that when the search direction $p_i \neq 0$, but $p_i^T \mathcal{A}p_i = 0$ (line 7), algorithm PCOCG breaks down. The breakdown can be fixed to some extent by restarting the process [22], such as the restarted process in GMRES [24]. However, breakdown scarcely happens in the actual computation of the Helmholtz equation.

3. Numerical Experiments

In this section, some numerical experiments are given to demonstrate the performance of both preconditioner P_{SSOR} and preconditioner P_{MSSOR} for solving the Helmholtz equation.

Example 3.1 (see [25]). Consider the following complex Helmholtz equation:

$$-\Delta u - pu + iqu = f \quad \text{in } \Omega,$$
$$u = g \quad \text{on } \partial\Omega, \tag{3.1}$$

Table 1: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/19$ meshes, using COCG with the preconditioner P_{SSOR} and P_{MSSOR} for solving the complex symmetric indefinite linear systems.

h	Preconditioner		(800, 10)	(800, 20)	(800, 30)	(800, 40)	(800, 60)
					(p, q)		
1/19	P_{SSOR}	IT	246	242	239	214	196
		CPU(s)	0.8438	0.8125	0.7813	0.7344	0.7031
	P_{MSSOR}	IT	138	133	126	117	96
		CPU(s)	0.25	0.2188	0.2031	0.1875	0.1719

Table 2: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/34$ meshes, using COCG with the preconditioner P_{SSOR} and P_{MSSOR} for solving the complex symmetric indefinite linear systems.

h	Preconditioner		(1400, 40)	(1500, 40)	(1600, 40)	(1700, 40)	(1800, 40)
					(p, q)		
1/34	P_{SSOR}	IT	230	274	310	336	399
		CPU(s)	3.2813	3.9375	4.8438	4.7188	5.6719
	P_{MSSOR}	IT	214	228	237	249	260
		CPU(s)	1.4375	1.5156	1.6563	1.6250	1.7188

Table 3: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/64$ and $p = 4100$, using COCG with the preconditioner P_{SSOR} and P_{MSSOR}.

h	Preconditioner		100	120	150	160	180
					q		
1/64	P_{SSOR}	IT	471	456	374	349	309
		CPU(s)	32.9344	30.9688	27.9844	26.0156	21.0938
	P_{MSSOR}	IT	373	366	347	326	293
		CPU(s)	12.4688	12.2500	11.6094	10.9063	9.4219

Table 4: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/119$ and $q = 2000$, using COCG with the preconditioner P_{SSOR} and P_{MSSOR}.

h	Preconditioner		15000	15500	16000	16500	17000
					p		
1/119	P_{SSOR}	IT	148	153	147	165	179
		CPU(s)	55.3594	57.4531	57.8594	52.4844	67.625
	P_{MSSOR}	IT	134	137	139	144	146
		CPU(s)	22.625	22.9063	23.2813	23.7344	24.625

where $\Omega = [0, 1] \times [0, 1]$ and $p \geq 0$ and $q \geq 0$ are real constants. Discretizing (3.1) with the approach above in introduction, we obtain the complex symmetric indefinite linear systems $\mathscr{A}x = (A - h^2 D + ih^2 E)\, x = b$, and f and g are adjusted such that $b = \mathscr{A}e$ ($e = (1, 1, \ldots, 1)^T$).

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

In Tables 1, 2, 3, and 4, we present some iteration results to illustrate the convergence behaviors of the COCG method preconditioned by P_{SSOR} and P_{MSSOR} to solve the complex

symmetric indefinite linear systems with the different values of p and q. In Tables 1–4, $(\cdot, \cdot)$ denotes the values of p and q. "CPU(s)" denotes the time (in seconds) required to solve a problem. "IT" denotes the number of iteration.

From Tables 1–4, it is not difficult to find that when the COCG method preconditioned by P_{SSOR} and P_{MSSOR} is used to solve the complex symmetric indefinite linear systems, the convergence rate of the preconditioner P_{MSSOR} is more efficient than that of the preconditioner P_{SSOR} by the iteration numbers and CPU time. That is, the preconditioner P_{MSSOR} outperforms the preconditioner P_{SSOR} under certain conditions. Compared with the preconditioner P_{SSOR}, the preconditioner P_{MSSOR} may be the "preferential" choice under certain conditions.

4. Conclusions

In this paper, MSSOR preconditioned COCG algorithm has been applied for solving the complex symmetric indefinite systems arising from Helmholtz equations. Due to the reduction of the iteration numbers and CPU time, the MSSOR preconditioner presented is feasible and effective. Without extra costs, MSSOR preconditioner is more efficient than SSOR preconditioner.
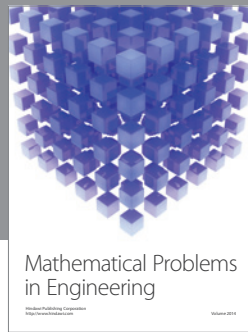
Acknowledgments

References

[1] C.-H. Guo, "Incomplete block factorization preconditioning for linear systems arising in the numerical solution of the Helmholtz equation," *Applied Numerical Mathematics*, vol. 19, no. 4, pp. 495–508, 1996.

[2] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, "On a class of preconditioners for solving the Helmholtz equation," *Applied Numerical Mathematics*, vol. 50, no. 3-4, pp. 409–425, 2004.

[3] A. Bayliss, C. I. Goldstein, and E. Turkel, "An iterative method for the Helmholtz equation," *Journal of Computational Physics*, vol. 49, no. 3, pp. 443–457, 1983.

[4] G. Bao and W. W. Sun, "A fast algorithm for the electromagnetic scattering from a large cavity," *SIAM Journal on Scientific Computing*, vol. 27, no. 2, pp. 553–574, 2005.

[5] Y. Wang, K. Du, and W. W. Sun, "Preconditioning iterative algorithm for the electromagnetic scattering from a large cavity," *Numerical Linear Algebra with Applications*, vol. 16, no. 5, pp. 345–363, 2009.

[6] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, Mass, USA, 1996.

[7] J. Gozani, A. Nachshon, and E. Turkel, "Conjugate gradient comipled with multi-grid for an indefinite problem," in *Advances in Computer Methods for Partial Differential Equations*, R. Vichnevestsky and R. S. Tepelman, Eds., vol. 5, pp. 425–427, IMACS, New Brunswick, NJ, USA, 1984.

[8] A. L. Laird, "Preconditioned iterative solution of the 2D Helmholtz equation," First Year's Report 02/12, Hugh's College, Oxford, UK, 2002.

[9] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, "Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation," *Applied Numerical Mathematics*, vol. 56, no. 5, pp. 648–666, 2006.

[10] M. Gander, "AILU for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization," *Journal of Computational Acoustics*, vol. 9, no. 4, pp. 1499–1506, 2001.

[11] R. E. Plessix and W. A. Mulder, "Separation-of-variables as a preconditioner for an iterative Helmholtz solver," *Applied Numerical Mathematics*, vol. 44, no. 3, pp. 385–400, 2003.

[12] M. M. M. Made, R. Beauwens, and G. Warzée, "Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix," *Communications in Numerical Methods in Engineering*, vol. 16, no. 11, pp. 801–817, 2000.

[13] M. Benzi, "Preconditioning techniques for large linear systems: a survey," *Journal of Computational Physics*, vol. 182, no. 2, pp. 418–477, 2002.

[14] F. Mazzia and R. Alan McCoy, "Numerical experimetns with a shifted SSOR preconditioner for symmetric matrices," type TR/PA/98/12, CERFACS, 1998.

[15] Z.-Z. Bai, "Mofidied block SSOR preconditioners for symmetric positive definite linear systems," *Annals of Operations Research*, vol. 103, pp. 263–282, 2001.

[16] X. Chen, K. C. Toh, and K. K. Phoon, "A modified SSOR preconditioner for sparse symmetric indefinite linear systems of equations," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 6, pp. 785–807, 2006.

[17] Y.-H. Ran and L. Yuan, "On modified block SSOR iteration methods for linear systems from steady incompressible viscous flow problems," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3050–3068, 2010.

[18] O. Axelsson, "A generalized SSOR method," *BIT*, vol. 18, pp. 443–467, 1972.

[19] A. Hadjidimos, "Successive overrelaxation (SOR) and related methods," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1-2, pp. 177–199, 2000.

[20] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, New York, NY, USA, 1995.

[21] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[22] H. A. van der Vorst and J. B. M. Melissen, "A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex," *IEEE Transactions on Magnetics*, vol. 26, pp. 706–708, 1990.

[23] R. W. Freund, "Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices," *SIAM Journal on Scientific Computing*, vol. 13, no. 1, pp. 425–448, 1992.

[24] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific Computing*, vol. 7, no. 3, pp. 856–869, 1986.

[25] S.-L. Wu, T.-Z. Huang, L. Li, and L.-L. Xiong, "Positive stable preconditioners for symmetric indefinite linear systems arising from Helmholtz equations," *Physics Letters A*, vol. 373, no. 29, pp. 2401–2407, 2009.