ELSEVIER

Engineering Analysis with Boundary Elements 31 (2007) 1013–1023

ENGINEERING ANALYSIS with BOUNDARY ELEMENTS

www.elsevier.com/locate/enganabound

Numerical experiments of preconditioned Krylov subspace methods solving the dense non-symmetric systems arising from BEM

Hong Xiao*, Zejun Chen

College of Mechanical Engineering, Yanshan University, Qinhuangdao 066004, China

Received 20 September 2006; accepted 2 April 2007
Available online 18 June 2007

Abstract

Discretization of boundary integral equations leads, in general, to fully populated non-symmetric linear systems of equations. An inherent drawback of boundary element method (BEM) is that, the non-symmetric dense linear systems must be solved. For large-scale problems, the direct methods require expensive computational cost and therefore the iterative methods are perhaps more preferable. This paper studies the comparative performances of preconditioned Krylov subspace solvers as bi-conjugate gradient (Bi-CG), generalized minimal residual (GMRES), conjugate gradient squared (CGS), quasi-minimal residual (QMR) and bi-conjugate gradient stabilized (Bi-CGStab) for the solution of dense non-symmetric systems. Several general preconditioners are also considered and assessed. The results of numerical experiments suggest that the preconditioned Krylov subspace methods are effective approaches solving the large-scale dense non-symmetric linear systems arising from BEM.

Keywords: Krylov subspace method; Preconditioner; Dense non-symmetric matrix; Boundary element method; Eigenvalue

1. Introduction

The boundary element method (BEM) is an important technique in the numerical solution of engineering and scientific problems. The boundary integral equation (BIE) technique is now well established as a complementary method to finite element method (FEM) for the analysis of many engineering problems. The basic idea of the BEM is to transform the original partial differential equations (PDEs) into an equivalent BIEs by means of corresponding Green's representation formula then to solve the corresponding BIE numerically. In applying BEM, only a mesh of the boundaries is required, making it easier to use and often more efficient than FEM, and reducing the number of unknowns to be solved. The computational bottlenecks of BEM are the computation of the coefficient matrix and the solution of the corresponding dense non-symmetric linear system. As the matrix of system arising from BEM is fully populated and non-symmetric, so the solution of the system is usually obtained using direct solvers as Gauss elimination, $O(N^3)$ operations and $O(N^2)$ memory are required which limit the BEM only apply to medium-size problems, where N is the number of degrees of freedom (DOF) of the numerical algorithm. Of course, various techniques have been exploited to restructure the matrix, e.g. adaptive cross approximation (ACA), fast multipole method (FMM), etc. The need to solve large-scale complex problems brings other solution techniques into focus, namely, the iterative methods. A fast convergent iterative method, which would require the number of iterations much less than N, will decrease the computational cost to $O(N^2)$. On the other hand, the problem remained in the solution of the dense non-symmetric linear systems obtained from the discretization step are usually ill-conditioned, especially when mixed boundary conditions exist or large-scale problems [1]. The classical iterative methods like Jacobi or Gauss-seidel, even with relaxation, either do not convergent or present very low convergence rates. The conjugate gradient (CG) method was first described by Hestenes and Stiefel for symmetric positive definite matrices and later generalized to non-symmetric

*Corresponding author. Tel.: +86 335 805 7032; fax: +86 335 807 4783.
E-mail address: xhh@ysu.edu.cn (H. Xiao).

0955-7997/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enganabound.2007.04.005

matrices leading to several methods: bi-conjugate gradient (Bi-CG) [2], generalized minimal residual (GMRES) [3], conjugate gradient squared (CGS) [4], quasi-minimal residual (QMR) [5] and bi-conjugate gradient stabilized (Bi-CGStab) [6]. However, for the solutions of large-scale problems, the convergence rates of iterative methods are poor even stagnate or divergent, thus, preconditioning is an essential technique for the successful use of these iterative methods.

In recent years, many researches have been carried out concerning the development of efficient iterative methods for BEM systems; see, for example, Refs. [7–15]. Among these researches, there is an interest on the analysis of the performances of Krylov subspace iterative methods solving linear systems arising from BEM formulation [13–15]. Otherwise, in order to accelerate the convergence rates of iterative methods, various preconditioning techniques have been proposed in the open literatures [16–20]. A number of specialized schemes such as wavelet-based approximation or the fast multipole algorithm have been proposed too which lead to efficient methods with linear computational complexity.

The aim of this paper is to study the convergence histories of Krylov subspace iterative methods, as applied to boundary element equations. We demonstrate the performance of several Krylov subspace iterative methods in conjunct with several general preconditioners, to solve dense non-symmetric matrices arising from BEM elasticity problems.

2. Krylov subspace methods

The final systems of equations of BEM are obtained by assuming that the discretized equations are satisfied exactly at a defined set of collocation points. The system can be written as

$$Ax = b, \tag{1}$$

where x is a vector which is composed of unknown displacements and tractions, A is the fully populated non-symmetric influence matrix and b is a vector obtained from the integrals in terms of the prescribed displacements and tractions.

The Krylov subspace methods are considered the most effective iterative methods currently available for dense non-symmetric linear systems. We review these methods briefly in this section.

Let x_0 be an initial approximation to the solution of Eq. (1), $r_0 = b - Ax_0$ be the initial residual and let $K_m(A, r_0) = \mathrm{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\}$ be the Krylov subspace of dimension m defined by A and r_0. Note that these subspaces are nested, i.e. $K_m \subseteq K_{m+1}$.

The Krylov subspace methods are iterative methods in which at the mth step an approximation to the solution of Eq. (1), x_m is found in $x_0 + K_m$, i.e., this approximation is of the form $x_m = x_0 + q_{m-1}(A)r_0$, where q_{m-1} is a polynomial of degree at most $m-1$. The residual $r_m = b - Ax_m$ is associated with the so-called residual polynomial p_m of degree at most m with $p_m(0) = 1$, since

$$r_m = b - Ax_m = r_0 - Aq_{m-1}(A)r_0 = p_m(A)r_0. \tag{2}$$

Analogously, the error satisfies $x_m - x^* = p_m(A)(x_0 - x^*)$, where x^* is the solution of Eq. (1). Let us denote by P_m the set of all polynomials p of degree at most m such that $p(0) = 1$. The approximate solution $x_m \in x_0 + K_m$ is often found by requiring x_m to be the minimal values of some functions. Different methods depend on the choices of this function, on the characteristics of the matrix, and on some implementation details [22].

For detailed comments concerning several general Krylov subspace methods (Bi-CG, GMRES, CGS, QMR and Bi-CGStab), see Refs. [2–6]. For pseudocodes of preconditioned Krylov subspace methods, see Ref. [21].

3. Preconditioners and stopping criterion

It is well known that the convergence rate of an iterative method for solving Eq. (1) depends greatly on the spectral properties of coefficient matrix A. For ill-conditioned problems, the convergence may be very slow, the iterative processes are likely to stagnate or even diverge. If a matrix M approximates the coefficient matrix A in some ways, the transformed system $M^{-1}Ax = M^{-1}b$ has the same solution as the original system $Ax = b$, but the spectral properties of its coefficient matrix $M^{-1}A$ may be more favorable. The M is so-called the preconditioner to accomplish such a transformation.

In practice, the preconditioner should meet two requirements as follows. First, the preconditioned matrix $M^{-1}A$ must be (much) better conditioned than A so that the equation $M^{-1}Ax = M^{-1}b$ can be solved by iterative technique efficiently. Second, the cost of constructing and applying the preconditioner M must be cheap. The two requirements are contradictive commonly, so there is a trade-off between constructing and applying the preconditioning matrix. The goal of preconditioning is that the total time solving the preconditioned system is less than the original system [21].

Most existing preconditioners can be broadly classified as being either of the implicit or of the explicit kind. A preconditioner is implicit if its application, within each step of the chosen iterative method, requires the solution of a linear system. Perhaps, the most important example is provided by preconditioners based on incomplete LU (ILU) decomposition. Here, M is implicitly defined by $M = \bar{L}\bar{U}$, where $\bar{L}$ and $\bar{U}$ are triangular matrices which approximate the exact L and U factors of A. Applying the preconditioner requires the solution of two triangular systems. Implicit preconditioners are quite robust and often give fast convergence of the preconditioned iteration, but are difficult to implement in parallel. In contrast, with explicit preconditioning, a matrix $M \approx A$ is explicitly computed and the preconditioning operation reduces to forming a matrix–vector product or a matrix–matrix

product if multiple right-hand sides are present. These computations are easier to parallelize than the sparse triangular solves. Furthermore, the construction of some types of approximate inverse preconditioners can be performed in parallel.

3.1. Jacobi preconditioning

The simplest preconditioner consists of just the diagonal of the matrix:

$$m_{i,j} = \begin{cases} a_{i,j} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This is known as the Jacobi preconditioner. It is possible to use this preconditioner without using any extra storage beyond that of the matrix itself [21].

3.2. Block Jacobi preconditioning

Block versions of the Jacobi preconditioner can be derived by a partitioning of the variables. If the index set $S = \{1, 2, \ldots, n\}$ is partitioned as $S = \cup_i S_i$ with the sets S_i mutually disjoint, then

$$m_{i,j} = \begin{cases} a_{i,j} & \text{if } i \text{ and } j \text{ are in same index subset,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The preconditioner is now a block-diagonal matrix. Often, natural choices for the partitioning suggest themselves: in problems with multiple physical variables per node, blocks can be formed by grouping the equations per node. The partitioning can be based on the physical domain. Examples are a partitioning along lines in the 2D case, or planes in the 3D case [21].

3.3. Incomplete factorization preconditioners

A broad class of preconditioners is based on incomplete factorizations of the coefficient matrix. Such a preconditioner is then given in factored form $M = LU$ with L lower and U upper triangular. The efficiency of the preconditioner depends on how well M^{-1} approximates A^{-1}.

The most common type of incomplete factorizations is based on taking a set S of matrix positions, and keeping all positions outside this set equal to zero during the factorization. The resulting factorization is incomplete in the sense that fill is suppressed. Often, S is chosen to coincide with the set of non-zero positions in A, discarding all fill. This factorization type is called the Incomplete LU factorization of level zero. The pseudocode of ILU(0) is given in Ref. [21].

If not only we prohibit fill-in elements, but we also alter only the diagonal elements, i.e. any alterations of off-diagonal elements are ignored, we obtain a simplified version of ILU(0), called D-ILU. Splitting the coefficient matrix into its diagonal, lower triangular, and upper triangular parts as $A = D_A + L_A + U_A$, the preconditioner can be written as $M = (D + L_A)D^{-1}(D + U_A)$, where D is the diagonal matrix containing the pivots generated. The preconditioner is described in Ref. [21].

By definition, the L and U matrices in ILU(0) have together the same number of non-zero elements as the original matrix A. The accuracy of this incomplete factorization may be insufficient to yield an adequate rate of convergence. There are same people present the concept of level of fill-in ILUP(p).

There are a number of drawbacks to the above algorithm. First, the amount of fill-in and computational work for obtaining the ILUP(p) factorization is not predictable for $p > 0$. Secondly, the cost of updating the levels can be quite high.

The modification of above algorithm that leads to an incomplete factorization require account for sparse pattern include a dropping strategy for get rid of some fill-ins. A dropping rule can be applied to a whole row by apply the same rule to all elements of the row.

3.4. Stopping criterion

Since an iterative method computes successive approximations to the solution of a linear system, a practical test is needed to determine when to stop the iteration [21]. A good stopping criterion should

1. identify when the error $e^{(i)} = x^{(i)} - x$ is small enough to stop;
2. stop if the error is no longer decreasing or decreasing too slowly;
3. limit the maximum amount of time spent iterating.

The general stopping criterions are given as follows:

Criterion 1: $i > maxit$ or $||r^{(i)}|| \leqslant stop\_tol \cdot (||A|| \cdot ||x^{(i)}|| + ||b||)$.
Criterion 2: $i > maxit$ or $||r^{(i)}|| \leqslant stop\_tol \cdot ||b||$.
Criterion 3: $i > maxit$ or $||r^{(i)}|| \leqslant stop\_tol \cdot ||x^{(i)}||/||A||$.

The relative error $||e^{(i)}||/||x^{(i)}||$ in the computed solution is bounded by *stop_tol*, where *maxit* is the maximum number of iteration, $||b||$ is the norm of b, $||A||$ is the norm of A, *stop_tol* is the stopping toleration.

4. Numerical examples

In this section, we report on the results of numerical experiments aimed at assessing the performance and efficiency of general Krylov subspace methods in conjunction with different preconditioning techniques considered for the solution of large-scale dense non-symmetric linear systems arising from BEM elasticity problems. We investigate the residual histories of these Krylov subspace methods.

The experiments with different Krylov subspace methods are performed using the iterative solvers written in

FORTRAN90 and compiled Compaq Visual Fortran 6.5. All experiments are performed on a PC with 1 GB RAM memory under Windows XP Professional. The computer is equipped with a 2.8 GHz Intel Pentium 4 processor and no special software optimization or specialized hardware. The convenient choice of stopping criteria for different iterative methods is used in the stopping criteria 2 above Section 3.4.

4.1. Example 1

There are different performances and efficiencies of an iterative method using different precision numbers. The iterative behaviors of using single precision and double precision numbers to solve a problem are studied. The numerical example considered in this section consists of a $150\,\text{mm} \times 150\,\text{mm} \times 150\,\text{mm}$ on which uniaxial stress $\sigma_{zz} = 100\,\text{kPa}$ is applied. The properties of the cube are: Young's modulus $E = 210\,\text{GPa}$ and Poisson's ratio $v = 0.3$. The discretization employed 600 quadrilateral elements with a total 602 nodes and the number of DOF is 1806. The convergence histories of GMRES and Bi-CGStab are depicted in Figs. 1 and 2. The initial guess is the zero vector.

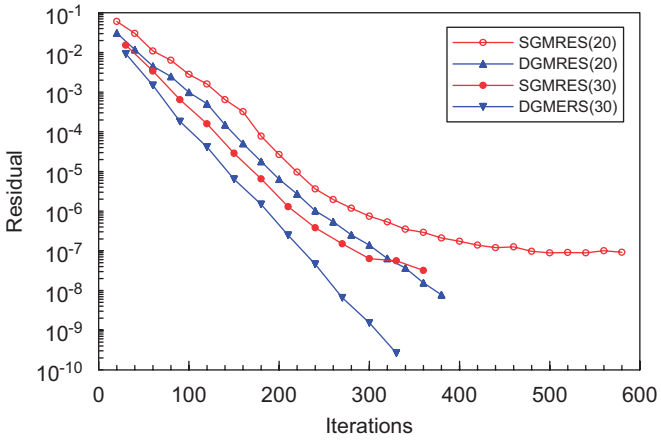

Fig. 1. Convergence histories of GEMRES.

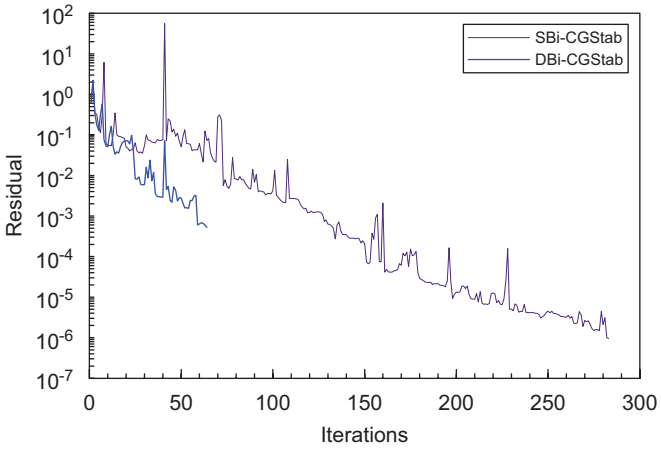

Fig. 2. Convergence histories of Bi-CGStab.

From Figs. 1 and 2, we compare the convergence histories of restarted GMRES and Bi-CGStab without using preconditioning techniques. The first letter "S" means single precision; "D" means double precision. From these figures, it can be seen that the convergence rates using double precision number is faster than single precision in

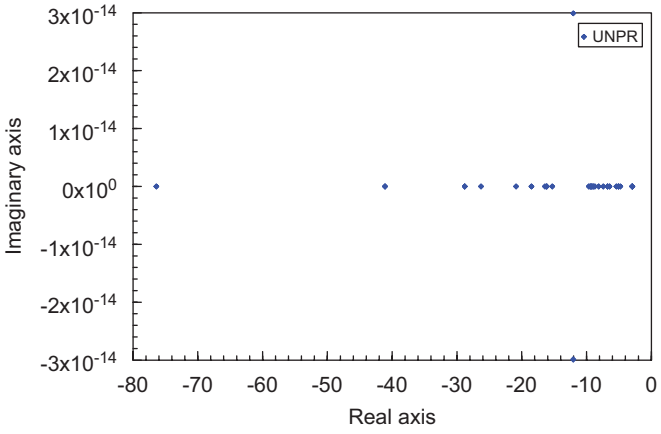

Fig. 3. Eigenvalues distribution of unpreconditioning.

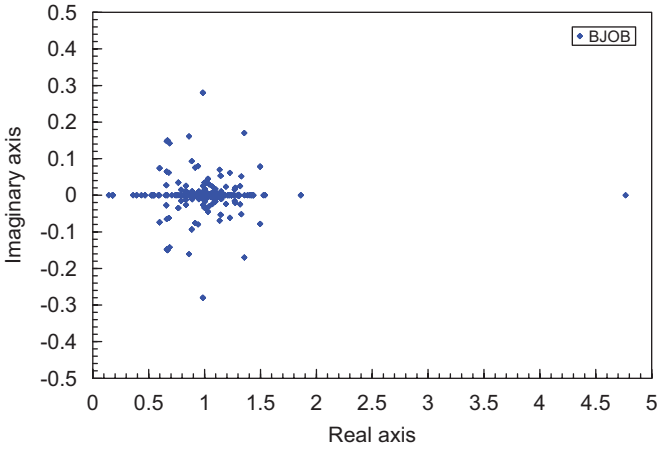

Fig. 4. Eigenvalues distribution of BJOB preconditioning.

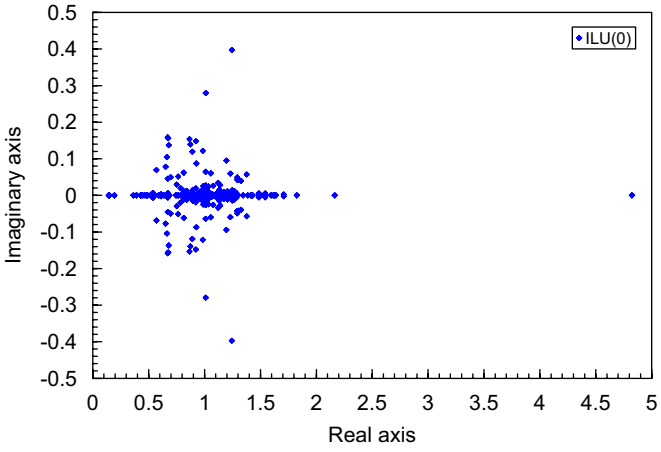

Fig. 5. Eigenvalues distribution of ILU(0) preconditioning.

the same conditions. Therefore, we employ the double precision number to compute the follow numerical examples so that we can research the properties of Krylov subspace methods more effectively.

4.2. Example 2

The convergence of Krylov subspace methods depends to a large extent on the eigenvalues distribution of the coefficient matrix. Distributions of the eigenvalues of unpreconditioned matrix A and preconditioned matrix $[MA]$ are shown in Figs. 3–6. In these figures, "UNPR" means without using preconditioning, "BJOB" means Block Jacobi preconditioning, "ILU0" means ILU(0) preconditioning, "DILU" means a simplified version of ILU(0) preconditioning.

From Fig. 3, it can be seen that the eigenvalues of the system are very scattered, all of them have a large negative real part and no clustering appears. Such a distribution is not at all favorable to the rapid convergence of Krylov subspace methods. From Figs. 4–6, it can be seen that the eigenvalues of preconditioned matrix $[MA]$ are well clustered around 1. One goal of preconditioning is to improve this distribution by grouping the eigenvalues into a few small clusters and around 1 as approximate as possible.

4.3. Example 3

In this section, we solve the same numerical problem using different preconditioned Krylov subspace methods. The model in this example is same as the model used in example 1. The convergence histories of Krylov subspace methods in conjunct with different preconditioning techniques are depicted in Figs. 7–11 which give us some clues

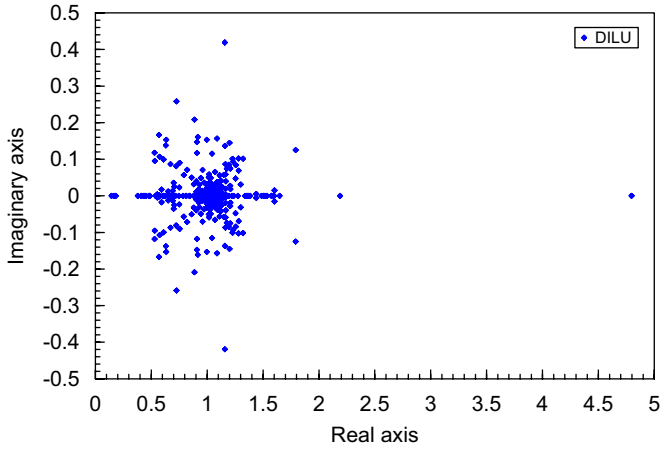

Fig. 6. Eigenvalues distribution of DILU preconditioning.

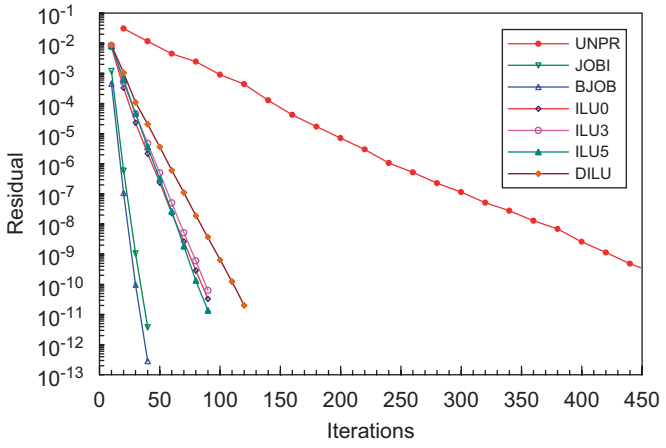

Fig. 8. Convergence histories of GMRES for different preconditioning techniques.

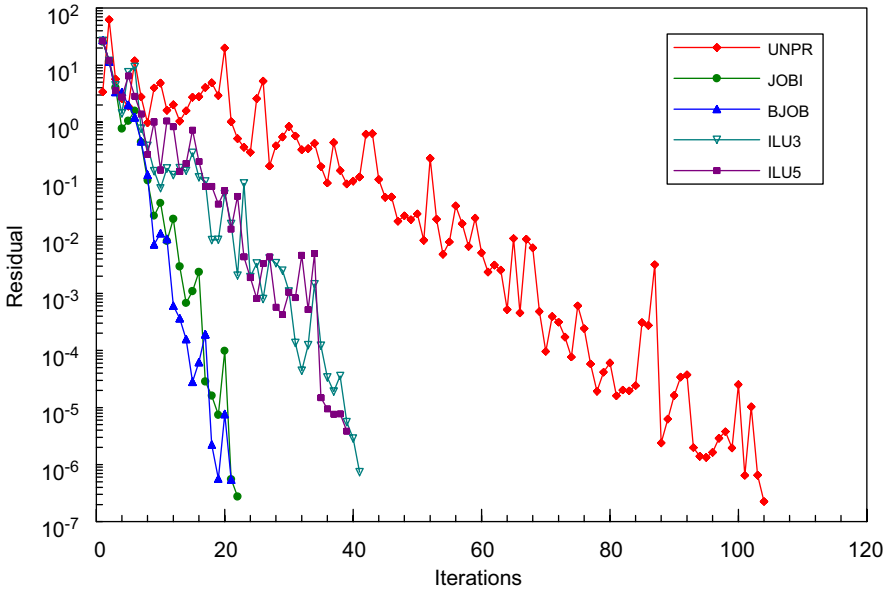

Fig. 7. Convergence histories of Bi-CG for different preconditioning techniques.

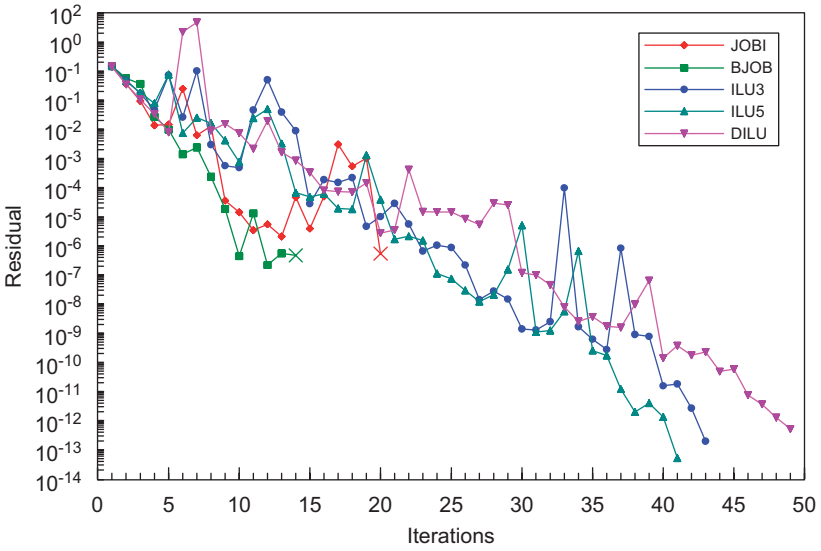

Fig. 9. Convergence histories of CGS for different preconditioning techniques.

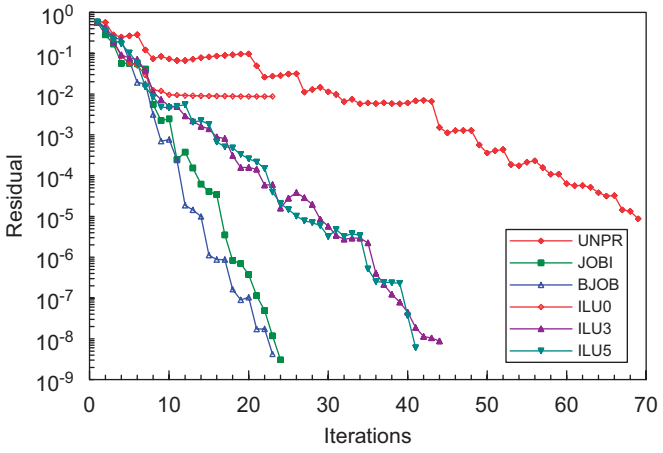

Fig. 10. Convergence histories of QMR for different preconditioning techniques.

concerning the numerical behaviors of the proposed schemes. In these figures, "JOBI" means Jacobi preconditioning, "ILU3" and "ILU5" mean Incomplete LU factorization preconditioning with Tri-diagonal and Five-diagonal non-zero structure, respectively. " × " means that the iterative process is break down or divergent.

From these figures, it can be seen that the convergence velocities of Krylov subspace methods can be significantly improved by using the preconditioning techniques. The residual values of Jacobi preconditioned Krylov subspace methods tend to decrease very rapidly compared with the tri-diagonal and five-diagonal non-zero structure preconditioning techniques. Among these Krylov subspace methods, the residual histories of GMRES decrease monotonically and rapidly with the increase of iterative number. Nevertheless, the convergence behaviors of other methods are quite irregular even break down. From Figs. 7 and 9, it can be seem that the convergence velocities of CGS are about twice as fast as for Bi-CG; however, the convergence behaviors of CGS iterative method are highly irregular and even break down in conjunct with some of preconditioning techniques.

4.4. Example 4

In this section, the performances of Krylov subspace iterative methods in conjunct with different preconditioning techniques for solving the linear equations arising from the BEM elasticity problems are investigated. The discretization is employed 1536 quadrilateral elements, and the number of DOF is 4614. Here, the convergence criterion is $\varepsilon = 1.0 \times 10^{-6}$. The computational time of solving the linear equation system using preconditioned Krylov subspace iterative methods along with the number of iterations are shown in Table 1. The "–" means the iterative process generate break down or divergence, the "min" means minute, "s" means second.

From Table 1, it can be seen that the solution time of preconditioned Krylov subspace methods outperforms the direct method, and the preconditioning techniques improve the convergence rates. With the increase of unknowns, the advantages of Krylov subspace methods will evident gradually. Thereby, the preconditioning techniques are indispensable to the numerical solution of large-scale problems.

4.5. Example 5

In this section, the dependencies of iterative performances of GMRES on restarted parameters are investigated. The model in this example is same as the model used in Example 1. The discretization employed 1536 quadrilateral elements, and the number of DOF is 4614. Here,

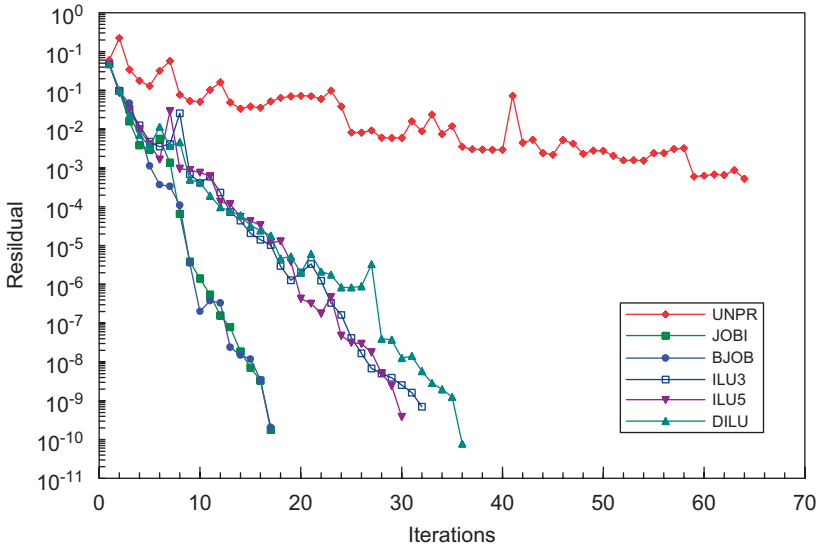

Fig. 11. Convergence histories of Bi-CGStab for different preconditioning techniques.

Table 1
Performances of preconditioned Krylov subspace iterative methods for DOF = 4614

Methods	Preconditioning techniques	Iterations	Residual	Time of each iterative step (s)	Time of solution
Direct methods	NO	–	–	–	10 min 2 s
Bi-CG	NO	174	7.29E−07	3.90	12 min 26 s
	JOBI	26	5.30E−07	3.91	2 min 45 s
	BJOB	22	1.42E−06	3.91	2 min 29 s
	ILU3	47	1.74E−06	3.89	4 min 6 s
	ILU5	48	1.46E−06	3.94	4 min 13 s
	DILU	–	–	5.50	–
GMRES(20)	NO	360	7.04E−07	2.01	12 min 56 s
	JOBI	20	4.50E−07	2.00	1 min 36 s
	BJOB	20	2.38E−08	2.00	1 min 36 s
	ILU3	40	7.95E−07	2.00	2 min 16 s
	ILU5	60	1.36E−08	2.00	2 min 56 s
	DILU	60	1.67E−07	3.12	4 min 4 s
GMRES(40)	NO	280	2.44E−08	1.95	7 min 26 s
	JOBI	40	5.07E−13	1.96	2 min 14 s
	BJOB	40	2.13E−11	1.96	2 min 14 s
	ILU3	40	4.78E−08	1.96	2 min 14 s
	ILU5	40	3.21E−08	2.65	2 min 14 s
	DILU	40	6.27E−08	3.12	2 min 59 s
CGS	NO	–	–	3.83	–
	JOBI	12	1.05E−05	3.81	1 min 45 s
	BJOB	12	2.30E−06	3.81	1 min 46 s
	ILU3	26	3.71E−06	3.80	2 min 38 s
	ILU5	22	4.13E−06	3.80	2 min 23 s
	DILU	28	5.27E−06	5.98	3 min 48 s
QMR	NO	135	7.32E−07	3.85	9 min 54 s
	JOBI	22	4.95E−07	3.80	2 min 43 s
	BJOB	18	9.35E−07	5.23	3 min 8 s
	ILU3	42	2.49E−07	3.86	3 min 58 s
	ILU5	36	6.49E−07	3.83	3 min 28 s
	DILU	–	–	5.05	–
Bi-CGStab	NO	129	1.38E−04	3.80	9 min 26 s
	JOBI	16	8.72E−07	3.81	2 min 8 s

Table 1 (*continued*)

Methods	Preconditioning techniques	Iterations	Residual	Time of each iterative step (s)	Time of solution
	BJOB	14	1.77E−07	3.82	2 min 0 s
	ILU3	24	9.37E−07	3.92	2 min 36 s
	ILU5	31	1.99E−07	3.95	3 min 3 s
	DILU	27	8.19E−07	6.00	3 min 41 s

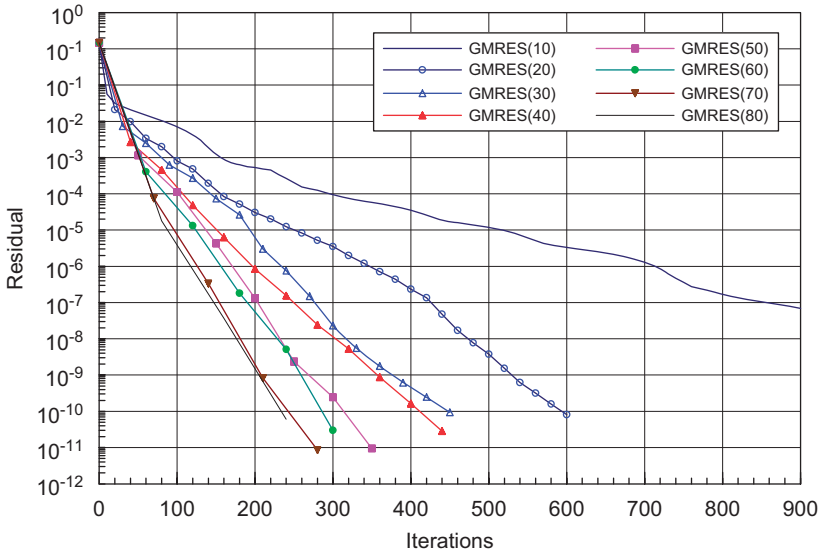

Fig. 12. Convergence histories of GMRES(m) for different restart parameter m.

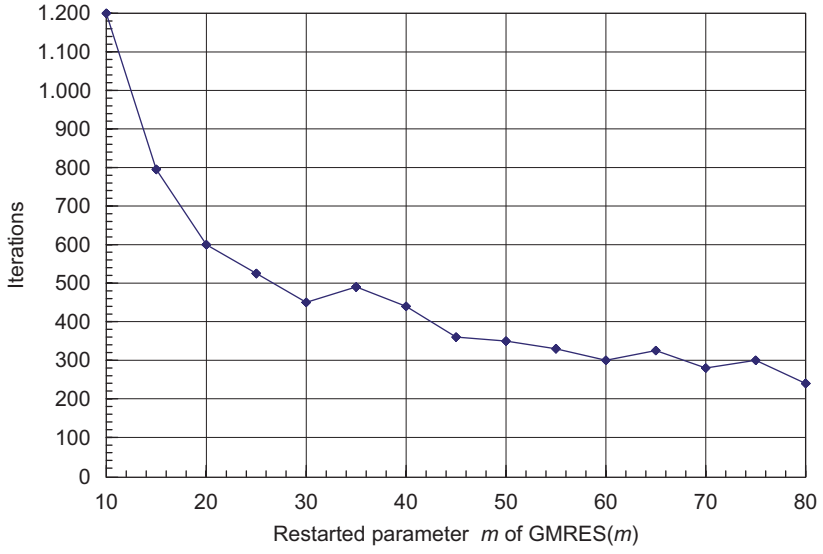

Fig. 13. The relationship of iterations to the restart parameter m.

the convergence criterion $\varepsilon = 1.0 \times 10^{-10}$ was chosen for all subsequent calculations.

We use the GMRES with different restarted parameter m to compute the example. According to the results of numerical tests, we plot the residual errors against the number of iterations in Fig. 12. From the figure, it can be seen that the performances of GMRES with the large restarted parameter outperform the small restarted parameter.

The relationship of the number of iterations to the restarted parameters is shown in Fig. 13. From the figure, it can be seen that the number of iterations decreases with the

increase of restarted parameter. Therefore, the smaller restarted parameter is unsuitable for the engineering computation. However, with the increase of restarted parameter, the requirement of memory is proportional increase, too. There is a trade-off between CPU time and the requirement of memory.

4.6. Example 6

From the results of above numerical experiments, it can be seen that the GMRES, QMR and Bi-CGStab are more effective in comparison with other Krylov subspace methods. In this section, we emphatically investigate the performances of the three methods solving different DOFs of BEM problems. The Block Jacobi preconditioning technique is employed. The results are shown in Table 2.

From Table 2, it can be seen that the performance of GMRES(30) is superior to other iterative methods. The choice of restarted parameter can affect the efficiency of GMRES. On the other hand, we can find that the number of iterations increases slowly with the increase of DOFs, and the number of iterations is fairly small relative to the number of unknowns.

Figs. 14 and 15 show that the performances of QMR and Bi-CGStab in conjunct with Block Jacobi preconditioning technique for different numbers of DOFs, respectively.

Table 2
The CPU time (in s) and the number of iterations for Krylov subspace methods

Solvers		DOF				
		1158	1806	4056	4614	7206
GMRES(15)	Residual	8.43E−13	9.25E−12	3.29E−10	3.41E−10	1.12E−11
	CPU time (s)	4.72	10.54	91.03	117.05	379.35
	Iterations	30	30	30	30	45
GMRES(30)	Residual	6.81E−12	5.38E−12	4.13E−11	4.17E−11	7.57E−11
	CPU time (s)	4.78	10.63	89.14	114.25	293.29
	Iterations	30	30	30	30	30
GMRES(60)	Residual	2.98E−12	1.90E−12	1.28E−11	1.36E−11	1.22E−11
	CPU time (s)	5.18	11.50	132.48	171.09	445.51
	Iterations	60	60	60	60	60
QMR	Residual	7.20E−11	3.35E−11	2.72E−11	3.97E−11	8.33E−11
	CPU time (s)	4.85	11.41	138.17	182.40	514.72
	Iterations	26	29	31	32	36
Bi-CGStab	Residual	5.85E−11	6.52E−12	8.82E−11	4.62E−10	8.79E−11
	CPU time (s)	4.57	10.64	101.51	126.69	354.17
	Iterations	17	19	19	18	21

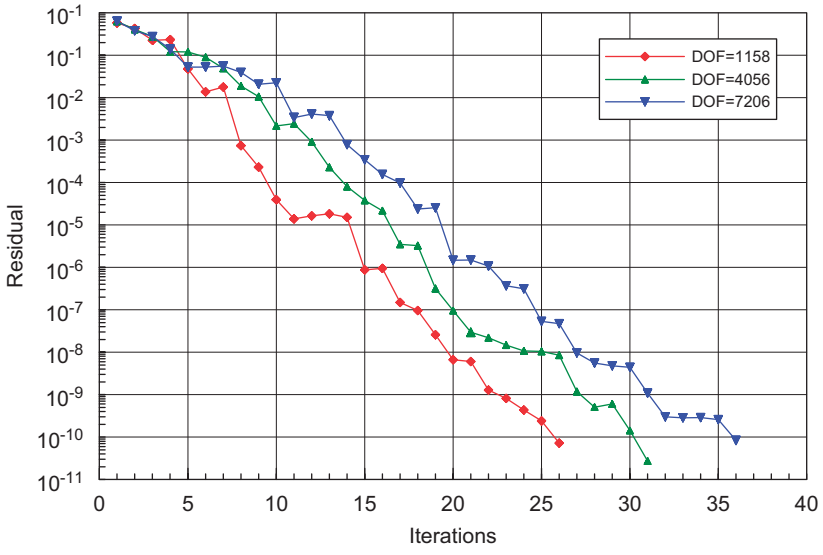

Fig. 14. Convergence histories of QMR for different DOFs.

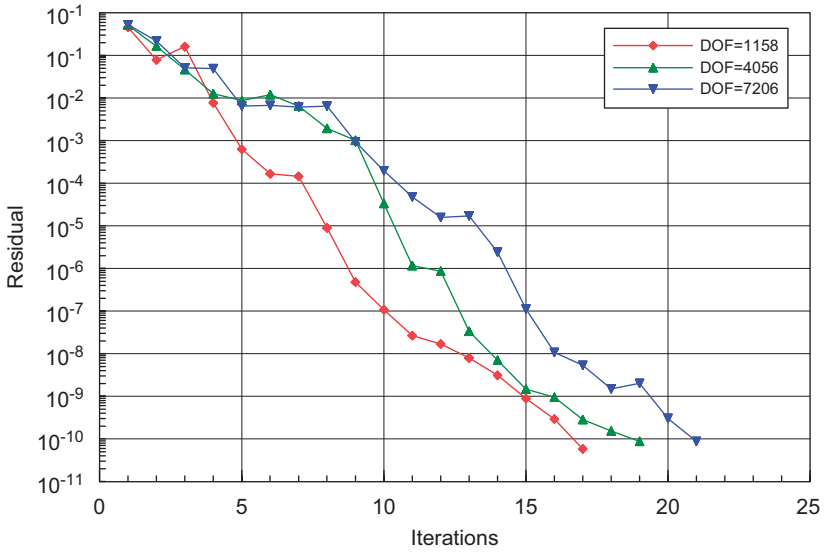

Fig. 15. Convergence histories of Bi-CGStab for different DOFs.

These curves are related to cases of DOF = 1158, 4056 and 7206, respectively. The figures indicate that the residues decrease rapidly with the increase of iteration number. Analogously, the iterative numbers of QMR and Bi-CGStab increase slowly with the increase of DOFs, too.

5. Conclusions

In this paper, we have shown how the dense non-symmetric linear systems arising from BEM elasticity problems can be efficiently solved with preconditioned Krylov subspace iterative methods. The computational cost to achieve the solution using the Krylov subspace methods are proportional to $it\,N^2$ and $it \ll N$ (it is the number of iterations), the improvement of computational efficiency is very substantial in comparison with the direct solution methods.

The preconditioned Krylov subspace solvers are found faster than the unpreconditioned solvers for large-scale problems. The reason is that the dense non-symmetric matrices arising from BEM are strongly ill-conditioned, especially when mix and complex boundary condition exist or large-scale problems. Thereby, the preconditioning techniques are indispensable to the solution of large-scale problems.

Based on the results of numerical experiments, we enrich the experiences concerning the performances of Krylov subspace methods in conjunct with preconditioning techniques for the solution of large-scale dense non-symmetric linear systems arising from BEM elasticity problems. Preconditioned Krylov subspace methods are well effective iterative solvers in large-scale boundary element engineering analyses.

Acknowledgment

This work is supported by the Natural Science Foundation of China (No. 50475081).

References

[1] Yao Z, Wang H, Wang P, et al. On the preconditioners for fast multipole boundary element methods for 2D multi-domain elastostatics. Eng Anal Bound Elem 2005;29:673–88.
[2] Flecher R. Conjugate gradient methods for indefinite systems, Lecture notes in mathematics, vol. 506. Berlin: Springer; 1976. p. 73–89.
[3] Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 1986;7:856–69.
[4] Sonneveld P. CGS: a fast Lanczos-type solver for nonsymmetric linear systems. SIAM J Sci Stat Comput 1989;10(1):36–52.
[5] Freund RW, Nachtigal NM. QMR: a quasi-minimal residual method for non-Hermitian linear systems. Numer Math 1991;60:315–39.
[6] Van Der Vorst HA. Bi-CGStab: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comput 1992;13(2):631–44.
[7] Guru Prasad K, Kane JH, Keyes DE, Balakrishna C. Preconditioned Krylov solvers for BEA. Int J Numer Meth Eng 1994;37:1651–72.
[8] Leung CY, Walker SP. Iterative solution of large three-dimensional BEM analyses using GMRES technique. Int J Numer Meth Eng 1997;40(12):2227–36.
[9] Urekew TJ, Rencis J. An iterative solution strategy for boundary element equations from mixed boundary value problems. Comput Methods Appl Mech Eng 1994;118:13–28.
[10] Amini S, Maines ND. Preconditioned Krylov subspace methods for boundary element solution of the helmholtz equation. Int J Numer Meth Eng 1998;41:875–98.
[11] Mittal RC, Al-Kurdi AH. An efficient method for constructing an ILU preconditioner for solving large sparse nonsymmetric linear systems by the GMRES method. Comput Math Appl 2003;45:1757–72.
[12] Zhang S-L. A class of product-type Krylov-subspace methods for solving nonsymmetric linear systems. J Comput Appl Math 2002;149:297–305.
[13] Valente FP, Pina HLG. Conjugate gradient methods for three-dimensional BEM systems of equations. Eng Anal Bound Elem 2006;30:441–9.
[14] Fang Z, Ingber MS, Martinez MJ. The solution of magnetostatic BEM systems of equations using iterative methods. Eng Anal Bound Elem 2002;26:789–94.

[15] Valente FP, Pina HLG. Iterative techniques for 3-D boundary element method systems of equations. Eng Anal Bound Elem 2001;25:423–9.
[16] Chen K, Harris PJ. Efficient preconditioners for iterative solution of the boundary element equations for the three-dimensional Helmholtz equation. J Appl Numer Math 2001;36(4):475–89.
[17] Harris PJ, Chen K. On efficient preconditioners for iterative solution of a Galerkin boundary element equation for the three-dimensional exterior Helmholtz problem. J Comput Appl Math 2003;156:303–18.
[18] Davey K, Bounds S. A preconditioning strategy for the solution of linear boundary element systems using the GMRES method. Appl Numer Math 1997;23:443–56.
[19] Schneider S, Marburg S. Performance of iterative solvers for acoustic problems. Part II. Acceleration by ILU-type preconditioner. Eng Anal Bound Elem 2003;27:751–7.
[20] Guillaume A, Michele B, Luc G. A preconditioned GMRES for complex dense linear systems from electromagnetic wave scattering problems. Linear Algebra Appl 2006;416:135–47.
[21] Barrett R, et al. Templates for the solution of linear systems: building blocks for iterative methods. Philadelphia, PA: SIAM; 1994.
[22] Valeria S, Daniel BS. Recent computational developments in Krylov subspace methods for linear systems. Report 05-9-25. This report is available in 〈http://www.math.temple.edu/~szyld〉.