A unified boundary element method for the analysis of sound and shell-like structure interactions. II. Efficient solution techniques

Shaohai Chen and Yijun Liu[a)]
Department of Mechanical Engineering, P.O. Box 210072, University of Cincinnati, Cincinnati, Ohio 45221-0072

Xinyu Dou
Acoustics Technology Center, Motorola, Inc., Schaumburg, Illinois 60196

(Received 23 March 2000; revised 15 September 2000; accepted 16 September 2000)

Efficient solution methods are investigated in this paper for solving the linear system of equations resulting from the recently developed boundary element method (BEM) for the coupled structural acoustic analysis [S. H. Chen and Y. J. Liu, J. Acoust. Soc. Am. **106**, Pt. 1, 1247–1254 (1999)]. An iterative solver, namely, the quasiminimal residual method (QMR), is selected among others and found to be very favorable over the direct solver for solving the linear systems of equations with complex coefficients generated by the structural acoustic BEM. Four problem-dependent preconditioning schemes are developed to facilitate or accelerate the convergence of the iterative solver. A new effective preconditioner specially designed for frequency-sweep analysis is also presented in this paper. With this preconditioner, the iterative solver has been found to be stable in a frequency-sweep analysis and can converge much faster than the direct solver. The double-precision arithmetic is also found very useful in improving the convergence rate of the iterative solver for structural acoustic problems. © *2000 Acoustical Society of America.* [S0001-4966(00)04512-4]

PACS numbers: 43.20.Fn, 43.20.Rz, 43.40.Rj [CBB]

I. INTRODUCTION

Recently, a unified boundary element method (BEM) was developed for the coupled analysis of acoustic waves interacting with thin, elastic, shell-like structures.[1] Numerical examples demonstrated that the unified BEM developed is very effective and accurate for the analysis of sound and shell-like structure interactions in both scattering and radiation problems. The method is valid for shell-like structures with arbitrarily small or nonuniform thickness, and does not suffer from the thin-shape breakdown and fictitious eigenfrequency difficulty in the exterior acoustic domain. It is the objective of this paper to improve the efficiency of the unified BEM developed by reducing the solution time of solving the linear system of equations generated. An iterative solver of the Krylov subspace type, that is, the quasiminimal residual (QMR) iterative method, is investigated, among others, and found to be very efficient in this type of application. Preconditioning techniques to improve the convergence are developed which include reordering of the mesh, scaling of the submatrices, and other special treatments designed for improving the characteristics of the matrix of the system. Besides these problem-dependent preconditioning techniques, a preconditioner specially designed for frequency-sweep analysis is also presented in this paper.

The iterative methods currently available for solving a linear system of equations $[A]\{x\}=\{b\}$ can be characterized into two groups, namely, stationary and nonstationary iterative methods. The stationary iterative methods refer to iterative methods such as Jacobi, Gauss–Seidel, successive over-relaxation (SOR), and symmetric successive over-relaxation (SSOR). In general, the stationary iterative methods, if they converge, converge much slower than nonstationary methods.[2] The nonstationary methods include conjugate gradient (CG), conjugate gradient on the normal equations (CGNE), generalized minimal residual (GMRES), biconjugate gradient (BiCG), quasiminimal residual (QMR), conjugate gradient squared (CGS), biconjugate gradient stabilized (Bi-CGSTAB), Chebyshev iteration, and so on. All nonstationary iterative methods listed above except Chebyshev iteration are of the Krylov subspace type. The differences among them depend on how the basis for spanning the Krylov space is found and how the linear system is enforced in this space. A variety of convergence behaviors for these iterative solvers is therefore observed, although they are all closely related to the characteristics of the left-hand-side matrix $[A]$ and the right-hand-side vector $\{b\}$. CG is derived for symmetric positive definite linear systems, with its convergence rate depending on the condition number of $[A]$. BiCG is applicable to nonsymmetric systems and requires two matrix–vector multiplications (involving $[A]$ or its transpose $[A^T]$). The convergence behavior of BiCG is quite irregular and may suffer breakdowns. CGS is similar to BiCG and has no matrix–vector multiplication with $[A^T]$. It can converge faster than BiCG, although the convergence is still irregular and may be subject to the breakdown problem. BiCGSTAB

[a)]Author to whom correspondence should be addressed. Electronic mail: Yijun.Liu@uc.edu

is an improved version of CGS designed to avoid the breakdown that often occurred in CGS while preserving the fast convergence rate. GMRES works on nonsymmetric systems directly and generates orthogonal vectors that form the basis spanning the Krylov subspace. Only one matrix–vector multiplication is required at each iteration. It minimizes the residual norm with exact arithmetic in each iteration and guarantees the convergence in less than n steps without restart (where n is the number of equations). Because the orthogonal vectors at each iteration have to be generated by using all the previously computed vectors, the storage requirement and computational effort tend to increase proportionally. In order to control the storage requirement, restarts after a certain number of iterations are often needed. QMR is applicable to both symmetric and unsymmetric matrices with real or complex elements. It requires two matrix-vector multiplications per iteration, both with $[A]$ and $[A^T]$. Instead of forming the exact orthogonal vectors as in GMRES, it generates a biorghogonal basis for the Krylov subspace by using the Lanczos process with short recurrences. Two recurrence schemes, three-term and coupled two-term, have been developed. A look-ahead Lanczos algorithm has been employed to extend QMR to general non-Hermitian matrices and avoid the possible breakdowns in some cases. Smooth convergence behavior can be observed for QMR in general. A FORTRAN package, QMRPACK,[3] has been developed, which contains all the QMR algorithms. It should be noted that a transpose-free version of the QMR, TFQMR, has also been developed and added to the QMRPACK. For a complete review on solving linear system with iterative solvers, please refer to Refs. 2 and 4 and the references therein.

The convergence behavior of iterative solvers varies for different types of applications. For a particular type of problem, an iterative solver may or may not converge, or converge more slowly than direct solvers. A suitable preconditioning scheme can greatly improve the situation. Intensive research efforts have been directed to address the suitable iterative solvers for all the existing applications and the corresponding preconditioning schemes. As long as a competitive iterative solver and a suitable preconditioning scheme can be identified for a particular problem, high efficiencies in solving the linear system of equations can be expected over direct solvers.

Generic preconditioning involves finding an economically invertible matrix (often referred to in the literature as a *preconditioner*), and applying the inverse of that matrix through multiplication to the original linear system for a new linear system which has a coefficient matrix with more favorable characteristics. Clearly, it is desirable that the preconditioner resembles $[A]$, as the new coefficient matrix would be close to an identity matrix. The matrix–matrix multiplication is never computed explicitly but integrated into the iterative process, where the factorization of the preconditioning matrix is formed once, and only forward and backward substitution processes are needed for all iterations. The existing generic preconditioning schemes include diagonal preconditioning, block diagonal preconditioning, SSOR preconditioning, incomplete LU decomposition (ILUD), and so on. Diagonal preconditioning employs only the diagonal terms of $[A]$ to form the preconditioner. It is the simplest, but often quite effective for some applications. The block preconditioner is similar to the diagonal preconditioner but formed by the small blocks on the diagonal direction of $[A]$. The SSOR preconditioner is formulated from the diagonal, lower, and upper triangular parts of $[A]$. A special case of the SSOR preconditioner is the diagonal preconditioner. Although applying the SSOR preconditioner is relatively inexpensive, it is unlikely to obtain a preconditioner closely resembling $[A]$, which is the key feature of a good preconditioner. The ILUD preconditioner is formed by dropping off the nonzero elements in the factorization of $[A]$ in positions where $[A]$ has zeros. This preconditioning scheme is often very effective. The drawback, besides the memory consumption, is the long computing time needed even when $[A]$ is sparse. As a BEM formulation usually generates a fully populated matrix, this preconditioning scheme is not feasible. Preconditioning can also be performed in the process of forming the linear system, which is problem dependent. This is often more effective and economical than generic preconditioning. Some preconditioning schemes of this kind are presented in this paper and shown to be effective for the coupled sound thin-shell interaction problems considered, although with all the preconditioning schemes ever developed, the iterative solution methods in general are considered to be less stable than direct solution methods. This situation may have been changed with a preconditioning scheme developed in this paper for problems requiring frequency-sweep analysis.

To the authors' best knowledge, no applications of iterative solvers for frequency-dependent problems, using the BEM have been reported in the literature. For static problems, such as elastostatic and potential problems, the performance of the iterative solvers in solving linear system of equations generated by the BEM have been reported.[5–8] In Ref. 5, CGN and GMRES were applied to 2D elastostatic problems with the use of diagonal, block diagonal, and ILUD preconditioners. Example problems with degrees of freedom up to 488 were tested using double-precision arithmetic with residual norm of 10^{-6} as the stopping criterion. The preconditioned GMRES was found to be faster than the direct solver in general, while CGN was found not as fast with or without preconditioners. In Ref. 6, GMRES, CGS, BiCGSTAB, and CGN with diagonal and block diagonal preconditioners were tested on small thermal and elastic problems with 2D and 3D geometries. The largest 3D model used for the elastostatic analysis contains 541 nodes. Stopping criterion in the form of residual norm was set as 10^{-6} (10^{-7} in some cases) and 10^{-4} for elastic and thermal problems, respectively. GMRES with diagonal preconditioning was shown to be significantly faster than the direct solver and was the most effective solver among other iterative solvers tested. CGS and BiCGSTAB were also found faster than the direct solver when used with diagonal preconditioning. In Ref. 7, the comparison of a number of iterative solvers was performed using two linear systems (with 250 and 1000 equations) generated by the BEM for a 2D potential problem. Four different types of matrices, two from the conventional boundary integral equation (BIE) and the other two

from the hypersingular BIE, were considered. GMRES with no restart, BiCG, and QMR outperformed other iterative schemes such as CG, CGS, CGN, and BiCGSTAB, when no preconditioning schemes were considered for all the algorithms. All the iterative algorithms were in general faster than direct methods, even with very stringent stopping criterion (10^{-10}). In Ref. 8, a reorthogonalization scheme with double-precision arithmetic was adopted to generate more accurate basis vectors for GMRES, which was then used to solve a relatively large linear system (up to 4902 equations) resulting from the BEM for 3D elastostatic problems. The solution time in the case of 4902 equations was more than five times faster than that of a direct solver, even with a stringent stopping criterion (10^{-7}). The non-restart version of GMRES was used in Ref. 8.

For dynamic analyses formulated in the BEM, QMR is a very good candidate among all the iterative solvers as it is applicable to unsymmetrical matrices with complex elements and less prone to numerical breakdowns. In this paper, the feasibility and efficiency of the iterative solver QMR in solving the linear system resulting from a BEM formulation of a frequency-dependent coupled structural acoustic problem are demonstrated. In the following, the stopping criterion used in QMR for BEM applications is justified first. The performance of QMR is then demonstrated by a pure acoustic application formulated in the BEM. The preconditioning schemes developed in this paper are then described in detail and tested. Finally, the comparison of the direct solver (LAPACK) and the iterative solver QMR for solving the linear system resulting from the BEM formulation for a coupled sound–structure interaction problem is presented using the testing cases.

II. THE ITERATIVE SOLVER FOR THE STRUCTURAL ACOUSTIC BEM

The detailed formulation of the unified boundary element method for analyzing the coupled sound–structure interaction problem has been presented in Ref. 1. The two sets of ordinary differential equations corresponding to the acoustic field and the elastic field in the frequency domain are recast into two sets of boundary integral equations (BIEs), which are coupled by the interface conditions defined on the wet surface of the elastic structure. After the discretization of the two BIEs on the surfaces of the structure, a linear system of equations $[A]\{x\}=\{b\}$ is obtained, which has the following structure:

$$\begin{bmatrix} H & D & 0 \\ E_a & T_{aa} & T_{ab} \\ E_b & T_{ba} & T_{bb} \end{bmatrix} \begin{Bmatrix} \Phi \\ u_a \\ u_b \end{Bmatrix} = \begin{Bmatrix} \Phi' \\ -U_{ab}t_b \\ -U_{bb}t_b \end{Bmatrix}, \tag{1}$$

where $\{\Phi\}$ and $\{u\}$ are vectors that account for the total disturbed acoustic pressure and displacement at the nodes, respectively; $[H]$ and $[T]$ are square submatrices resulting from the singular kernel of the BIE for acoustic field and elastodynamic field, respectively; $[D]$ and $[E]$ are rectangular submatrices obtained after applying the interface conditions; the subscripts a and b denote the outer (wet) surface and the inner (dry) surface, respectively; $\{\Phi'\}$ and $\{t_b\}$ are known vectors resulting from the incident wave and traction on the inner surface, respectively. Please refer to Ref. 1 for the derivation of (1).

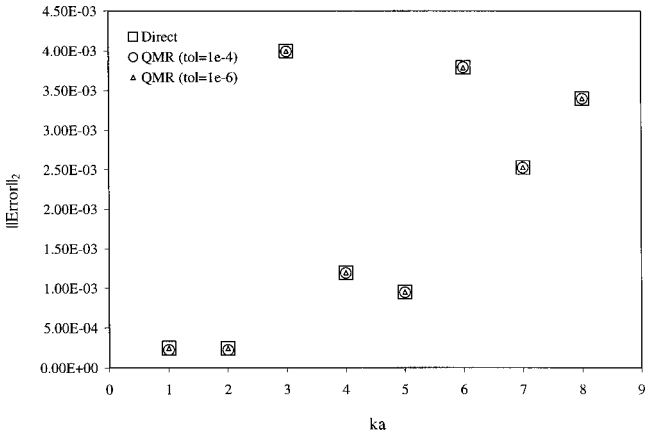

FIG. 1. Error level for the radiation analysis of a pulsating sphere (DOF = 290).

In this paper, the performance of the iterative solver QMR in solving linear systems of complex equations resulting from the BEM, as shown in Eq. (1), is investigated. The QMR algorithm based on the coupled two-term variant of the look-ahead Lanczos process[3] is used. The iterative solver is considered converged when the relative residual norm is less than a preset value (often referred to as tolerance). The relative residual norm is defined as

$$\frac{\|r_n\|_2}{\|r_0\|_2}=\frac{\|Ax_n-b\|_2}{\|Ax_0-b\|_2}, \tag{2}$$

where $\{x_n\}$ is the solution vector at the end of the nth iteration, $\{x_0\}$ the initial guess (usually set as a zero vector), $\{r_n\}$ the residual vector and $\|\cdot\|_2$ the Euclidean norm. The preset tolerance has to be small enough so that reliable results can be obtained, but not so small that computation efforts are wasted.

The proper value of the tolerance was studied numerically using the radiation problem of a pulsating sphere (no coupling with elastic structure), for which the analytical solution is available. A mesh consisting of quadratic elements with 290 nodes was generated over the surface of a unit sphere for this purpose. The corresponding linear systems formed after the discretization of the BIEs were solved using both direct solver and iterative solver. The error with respect to the analytical solution was calculated in the Euclidean norm

$$\|\text{Error}\|_2=\frac{\|x-X\|_2}{\|X\|_2}, \tag{3}$$

where $\{X\}$ represents the analytical solution, $\{x\}$ the solution resulting from the BEM using direct solver or iterative solver. The error levels at eight frequencies from using direct solver and QMR with two stopping tolerances (10^{-4} and 10^{-6}) are shown in Fig. 1. It can be seen that the direct solver and QMR achieved virtually the same level of accuracy over all eight frequencies. The error level is in general increasing toward higher frequencies, due to the fact that there are fewer elements within one wavelength. Higher er-

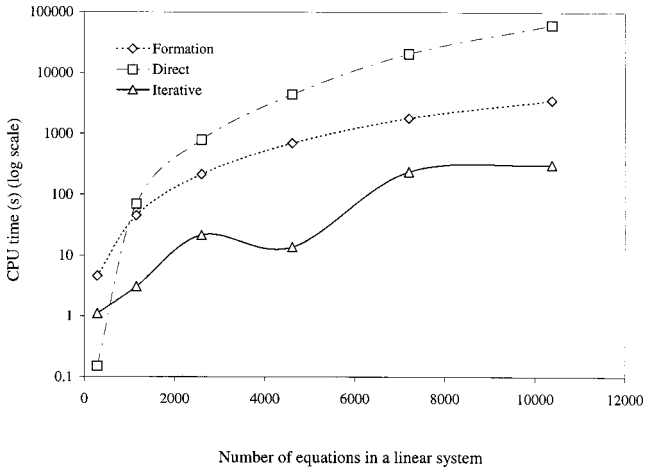

FIG. 2. CPU time consumption for the radiation analysis of a pulsating sphere.

ror levels are observed at $ka=3$ and 6, which are in the vicinity of the fictitious eigenfrequencies π and 2π, due to the adverse matrix characteristics although unique solutions have been guaranteed by Burton and Miller's formulation (which employs the hypersingular BIE). Since the round-off error (on the order of 10^{-4}) is small relative to the discretization error (on the order of 10^{-3}), solving the linear system more exactly (reducing round-off error) cannot render a more accurate result as compared to the analytical solution (reducing the discretization error). This fact is also supported by the results from the iterative solver. With the stopping criterion set as 10^{-4}, the iterative solver achieved the same level of accuracy as the direct solver (compared to the analytical solution). The solution resulting from using 10^{-6} as the tolerance gave trivial improvement in the accuracy with respect to the analytical solution (Fig. 1), while consuming three times more solution time. The stopping criterion is therefore set as 10^{-4} for all the following test cases, since errors in the 10^{-4} level are acceptable for most engineering purposes. Higher levels of accuracy can only be achieved by using finer meshes to reduce the discretization errors. It was observed that QMR is not free of breakdowns. In fact, when the tolerance was set as 10^{-6}, the iterative solver suffered abnormal termination before it reached the stopping tolerance at $ka=3$. This breakdown was not encountered when double precision was used (Fig. 1), as the direction vectors can be further refined in double-precision arithmetic.

III. PURE ACOUSTICS ANALYSIS—TEST RESULT

The iterative solver was first tested with pure acoustic problems, i.e., without the coupling with structures. The conventional BIE for the same pulsating sphere problem described in the previous section was discretized using six different meshes with the total number of nodes as: 290, 1154, 2594, 4610, 7202, and 10 370. The highest frequency feasible for each mesh ($ka=8$, 16, 24, 32, 40, and 48, respectively) was employed in obtaining the corresponding CPU time consumption. Single-precision arithmetic was used for all six cases. The saving in CPU time by using the iterative solver over direct solver is illustrated in Fig. 2. All data were

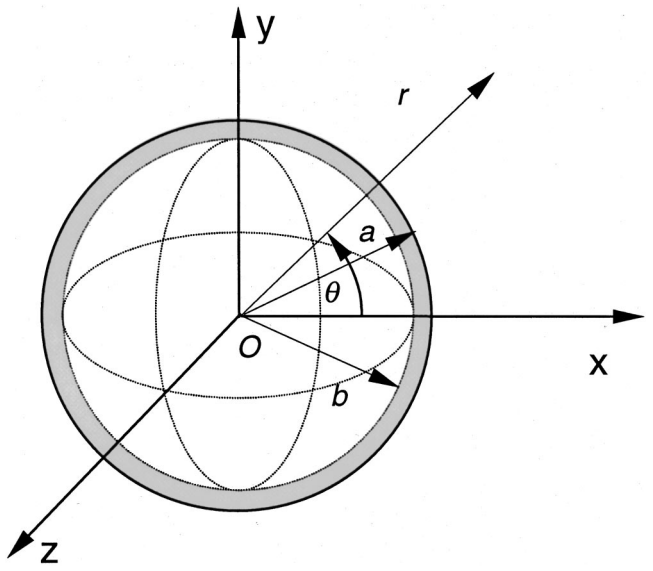

FIG. 3. A spherical shell with uniform thickness (outer radius$=a$, inner radius$=b$, thickness$=h=a-b$).

obtained on a Pentium II PC (400 MHz, 256 MB RAM) with WINDOWS NT operating system. An estimated value of the CPU time consumption of the direct solver for the case with the largest mesh was used since the CPU time was too long (estimated over 10 days of clock time) for the direct solver. The ratio between the time consumed by direct solver and that by iterative solver increases as the problem size increases. For the case with 7202 nodes, the iterative solver was 86 times faster than the direct solver. It should be noted that the CPU time consumption of the iterative solver is less for 4610 degrees of freedom than for 2594 degrees of freedom, as the iterative solver is sensitive to the conditioning of the system. The iterative solver was applied without using any preconditioning schemes in this pure acoustic case. From this test, the CPU time savings in solving the acoustic BEM equations using the iterative solver QMR are evident.

IV. COUPLED STRUCTURAL ACOUSTICS ANALYSIS—FIVE PRECONDITIONING SCHEMES

For the coupled problem, the linear system of equations (1) has very high condition numbers in general due to the mismatch of the materials (structure and fluid). A special partitioning scheme has to be used for the direct solver to obtain reliable solutions. The solution time can also be dramatically reduced by using the iterative solver, but not without the help of preconditioning. The three existing preconditioners available in the literature (the diagonal, the block-diagonal, and the SSOR preconditioners) were tested and found not working for the applications considered. Therefore, five new preconditioning schemes are developed in this study. They will be referred to as scheme 1, 2, 3, 4, and 5 in the following sections. A steel spherical shell (Fig. 3) immersed in water will be used as the test case for the first four preconditioning schemes. The dimension of the shell is $a=1\ m$, $h/a=0.01$, where a is the outer radius and h is the thickness.

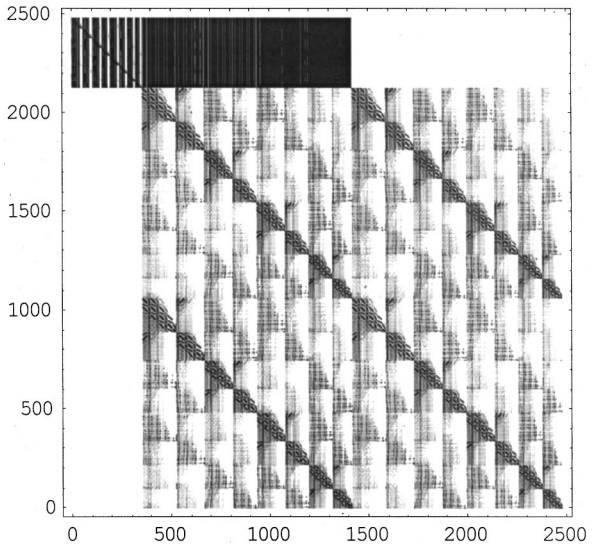

FIG. 4. Image of matrix $[A]$ before applying any of the four preconditioning schemes.

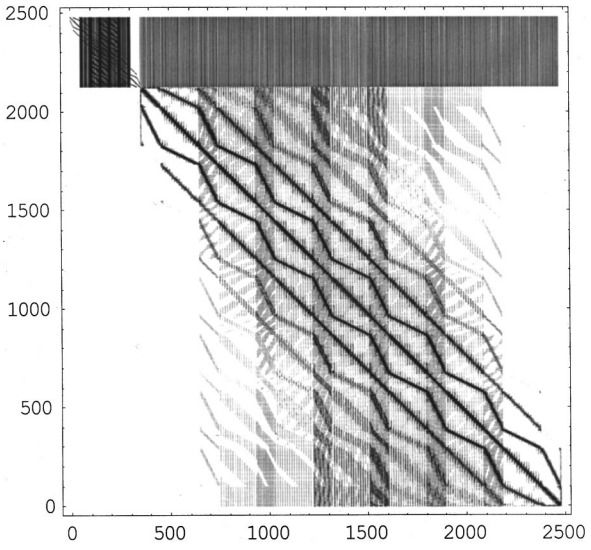

FIG. 6. Image of matrix $[A]$ after applying preconditioning schemes 1 and 2.

Scheme 1 reorders the nodes on the outer (wet) surface of the structure, which are used for the discretization of the acoustic BIE. This scheme is aimed at providing a suitable numbering of the nodes in the mesh to place all the dominant entries (i.e., with larger absolute values) of the matrix as close as possible to the main diagonal. Specifically, the nodes in the vicinity of every single node in the mesh will be assigned node numbers close to the node number of that node by an algorithm calculating and comparing the distance between nodes. Figures 4 and 5 show the image of the coefficient matrix $[A]$ resulting from the spherical shell model before and after applying scheme 1, respectively. The larger the absolute value of an entry, the darker the dot shown in the image. It can be seen in Fig. 4 that the $[H]$ and $[D]$ submatrices dominate matrix $[A]$ with very large entries distributed all over these submatrices. The $[E_a]$ and $[E_b]$ submatrices consist of very small entries (shown as a nearly white area in the image). The $[T_{aa}]$, $[T_{ab}]$, $[T_{ba}]$, and $[T_{bb}]$ submatrices in $[A]$ result from the singular integral operator for the elastic domain. The diagonal dominant phenomenon is evident in each of the four submatrices, although $[T_{ab}]$ and $[T_{ba}]$ can't contribute to the overall diagonal dominance of $[A]$. Figure 5 shows the image of the same coefficient matrix after applying scheme 1. Significant changes can be observed in submatrices $[H]$, $[T_{aa}]$, $[T_{ab}]$, $[T_{ba}]$, and $[T_{bb}]$.

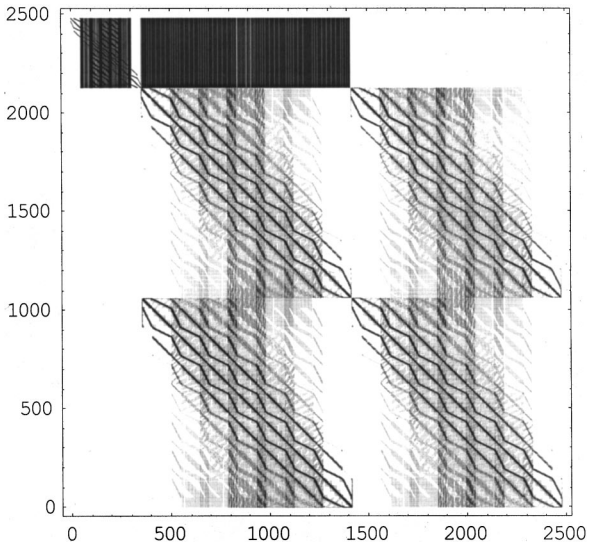

FIG. 5. Image of matrix $[A]$ after applying preconditioning scheme 1.

Scheme 2 involves reordering of all the nodes on the two structure surfaces for the discretization of the elastodynamic BIE. This reordering scheme is motivated by the fact that when the thickness of the shell gets smaller, the matrix entries with contributions from the nearly singular integrals become larger. It is therefore desirable to place these entries closer to the main diagonal by reordering the nodes. Figure 6 shows the coefficient matrix after applying schemes 1 and 2. The entries of the zero block of the primitive matrix are now mixed with the entries of the $[D]$ submatrix. It is evident that the $[T_{aa}]$, $[T_{ab}]$, $[T_{ba}]$, and $[T_{bb}]$ submatrices are now contributing toward the overall diagonal dominance of $[A]$. Because of the coupling process of the two domains, the matrix is again extremely unbalanced with $[D]$ featuring very large entries, while $[E_a]$ and $[E_b]$ consist of very small entries.

Scheme 3 is designed to render a better scaling of the coefficient matrix $[A]$. Because of the mismatch of the two domains with quite different material properties, the entities in $[D]$ are much larger than those in $[E]$. This kind of unbalance among entities in a system matrix results in a very high condition number. A scaling factor related to material properties is used to provide a better scaling. Figure 7 shows the resulting matrix after applying schemes 1, 2, and 3. The matrix is now evidently diagonally dominant with a very clear pattern except for the $[H]$ submatrix.

Scheme 4 is designed to further improve the characteristics of $[A]$ by utilizing the inverse of $[H]$, which is obtained by the direct method. This preconditioning scheme is more expensive than the previous three. However, the resulting coefficient matrix $[A]$ (Fig. 8) has better characteristics that often reduce the total solution time (including the time used for obtaining the inverse of $[H]$). This is due to the further

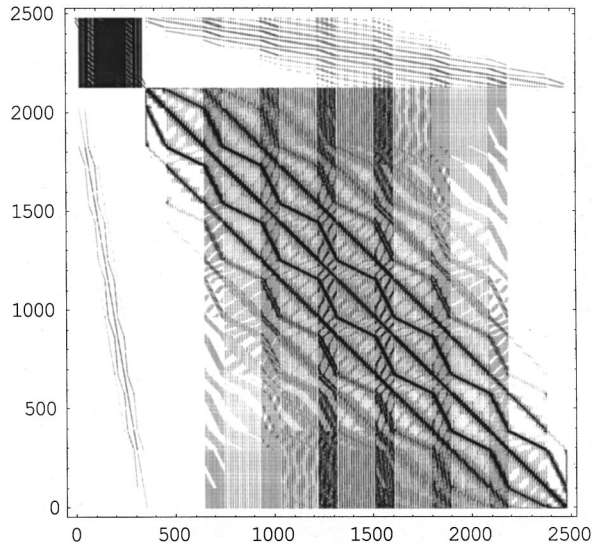

FIG. 7. Image of matrix [A] after applying preconditioning schemes 1, 2, and 3.

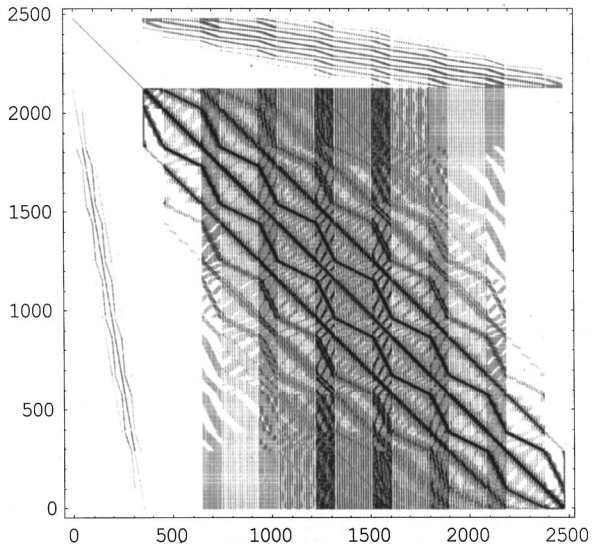

FIG. 8. Image of matrix [A] after applying preconditioning schemes 1, 2, 3, and 4.

reduction in the condition number of [A], as demonstrated in the following test cases.

In practice, a dynamic problem often needs to be solved for many frequencies. Scheme 5 is specially designed for this kind of application. It is found that the coefficient matrix [A] at the first frequency in a frequency sweep can serve as a perfect preconditioner for all the subsequent frequencies. As all the system matrices for different frequency cases result from the same geometry and boundary conditions, they resemble each other. Any one of them can be used as a preconditioner for all the other cases. With the help of this preconditioner, a frequency sweep for a dynamic problem can be performed very efficiently using the iterative solver. The inverse of this preconditioner is never computed explicitly. Instead, a more economical process, the LU factorization, is performed and the result is stored. The matrix–vector multiplication involving the preconditioner in each iteration step is then obtained by one forward and one backward substitution, which consume the same amount of computation effort as that of a direct matrix–vector multiplication. The matrix–vector multiplication involving the transpose of the preconditioner presents no extra computing effort. As the time consumed by each of the consecutive frequency cases can be dramatically reduced by using the iterative solver, the overall time consumption for the frequency sweep can be much less than that by using the direct solver for every frequency. The more frequencies are involved in a frequency sweep, the more saving in CPU time can be expected. The significance of this scheme is that the iterative solver can converge much faster than the direct solver and provide the efficiency in frequency-sweep analyses.

V. COUPLED STRUCTURAL ACOUSTICS ANALYSIS—TEST RESULTS

The effectiveness of the first four preconditioning schemes in accelerating the convergence of the iterative solver was tested first. Three different meshes for the spherical shell (Fig. 3) with 64, 256, and 576 quadratic elements, which yield 574, 2478, and 5726 equations in the final linear system, respectively, were used. Ten test cases representing all the interesting scenarios were performed using the mesh with 256 elements on the Pentium II PC. Results from five of the ten cases (case 1 with no preconditioning; case 2 with preconditioning scheme 1; case 3 with preconditioning schemes 1 and 2; case 4 with preconditioning schemes 1, 2, and 3; and case 5 with preconditioning schemes 1, 2, 3, and 4) are shown in Table I. It was found that with the application of all the four preconditioning schemes (case 5), the iterative solver converged at the fastest rate among all the test cases. The condition number of the original matrix dropped from the order of 10^8 to the order of 10^4. The corresponding CPU time consumption of the iterative solver is

TABLE I. The effectiveness of the preconditioning schemes. Note: o—not applied; √—applied; SP—single precision; DP—double precision.

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Number of iterations SP	Number of iterations DP	CPU time (s) SP	CPU time (s) DP
Case 1	o	o	o	o	>1000	>1000	···	···
Case 2	√	o	o	o	>1000	932	···	···
Case 3	√	√	o	o	>1000	933	···	···
Case 4	√	√	√	o	355	58	···	···
Case 5	√	√	√	√	190	47	425.83	196.15

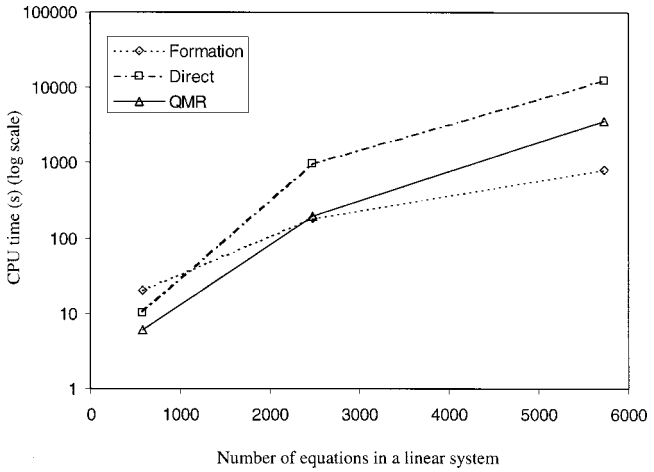

FIG. 9. CPU time consumption for the coupled radiation analysis of a steel spherical shell ($h/a=0.01$).

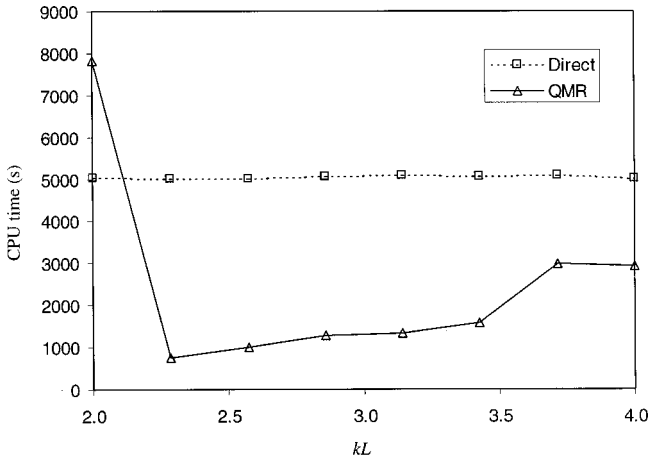

FIG. 11. Comparison of CPU time consumption at each frequency for a scattering problem on the submarine-like model using QMR and direct solver.

425.83 s and 196.15 s, compared to 581.38 s and 972.62 s consumed by the direct solver, for single-precision and double-precision arithmetic, respectively. The saving in solution effort rendered by the iterative solver over the direct solver is clearly demonstrated in this coupled analysis, especially in the case of double precision (about five times faster than the direct solver). To show the consistency of this approach, two additional tests were performed on the same spherical shell with the other two meshes using the four preconditioning schemes with double-precision arithmetic. The comparison of CPU time consumption of direct solver and QMR for all the three meshes are shown in Fig. 9. The formation time is also plotted as a reference. The effectiveness of the first four preconditioning schemes is evident.

To demonstrate the effectiveness of scheme 5 for the frequency-sweep analysis, a submarine-like model (Fig. 10) was studied next. The length of the submarine-like model is 7 m, main radius 0.5 m, and the thickness of the shell 0.01 m. The result from the iterative solver with the first four preconditioning schemes could not render fast convergence for this slender submarine-like model in the coupled analysis. With scheme 5, however, a stable result was obtained for the frequency-sweep analysis. The BEM model used in this study consists of 416 quadratic elements and 1188 nodes. The structure is immersed in seawater and impinged upon by a plane incident wave in the positive x direction. With the use of all the first four preconditioning schemes, the iterative solver could not even come close to convergence in 600 iterations, twice as much time as the direct solver would consume. This extremely slow convergence rate was dramatically changed by using the preconditioner in scheme 5. A frequency sweep over 8 frequencies from $kL=2$ to $kL=4$ was performed for demonstration purpose (L is the total length of the model).

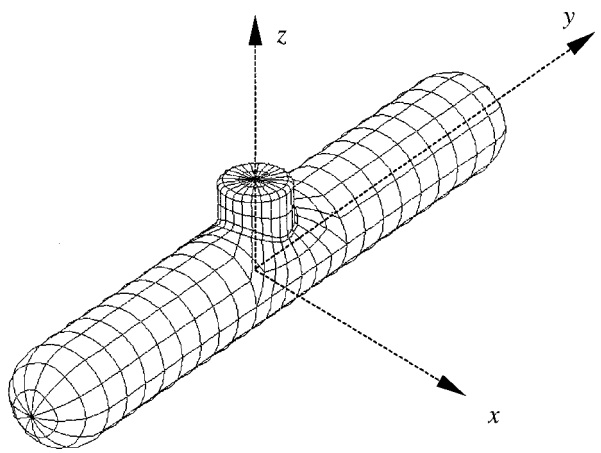

FIG. 10. A submarine-like model (main radius = 0.5 m, total length = 7 m, and thickness = 0.01 m).

Figure 11 shows the CPU time consumption by using QMR for each frequency case in the frequency sweep, as compared to the direct solver. The preconditioning schemes 1, 2, 3, and 5 were used. Besides the first frequency case, where the LU factorization of the preconditioner was performed, a great deal of savings in CPU time was achieved for all the subsequent cases. It can be seen in Fig. 11 that the preconditioner performed better when the frequency at which the calculation was conducted was closer to the frequency at which the preconditioner was generated. The reason is quite obvious, as the system matrix obtained after preconditioning would be closer to the identity matrix, when the two frequencies are closer to each other. In light of this fact, better performance can be expected when the preconditioner is generated at the middle frequency of a frequency span. The condition number of the resulting linear system (with 4158 unknowns) after applying schemes 1, 2, and 3 is on the order of 10^6. Scheme 4, which is the most costly one among the first four schemes, is found unnecessary when the preconditioner provided by scheme 5 is used. Since fast convergence is ensured by the preconditioning schemes, the CPU time consumption at each frequency can be much less than that of a direct solver. The more frequency steps in the frequency-sweep analysis, the more savings in CPU time can be achieved in the solution process. The forward-scattering and back-scattering results of the coupled analysis using QMR and the direct solver are compared in Figs. 12 and 13, respectively. The results from pure acoustic analysis (considering the structure as rigid and stationary) are also plotted as

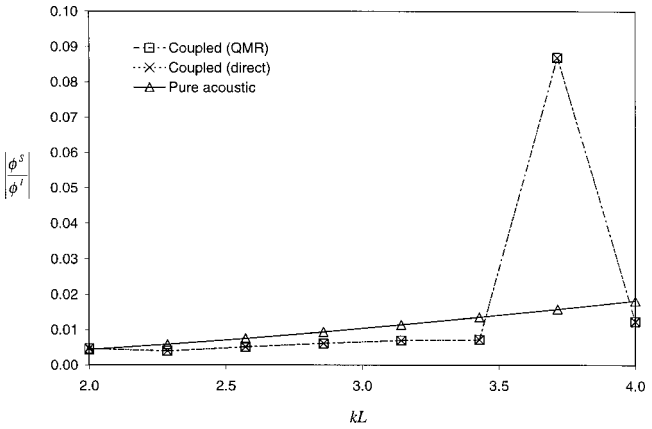

FIG. 12. Forward-scattering result at point (35, 0, 0).

a reference. The consistency between the results from QMR and the direct solver again demonstrates the efficiency and reliability of the iterative solver, and the sufficiency of using 10^{-4} as the stopping tolerance for the QMR solver.

More sophisticated numerical tests, for example those involving nonuniformly applied loads, at higher frequencies or larger models, need to be studied to further fine-tune the iterative solver for the analysis of coupled structural acoustics problems using the BEM.

VI. CONCLUSION

Effective solution schemes for the applications of the unified BEM to coupled sound and thin-shell structure interaction problems have been studied. An iterative solver, namely, the quasiminimal residual method (QMR), was selected among others and found to be much more efficient compared to the direct solver in solving the linear systems of equations with complex coefficients generated by the structural acoustic BEM. Four problem-dependent preconditioning schemes are developed to accelerate the convergence of the iterative solver. Double-precision arithmetic is also very useful in improving the convergence rate.

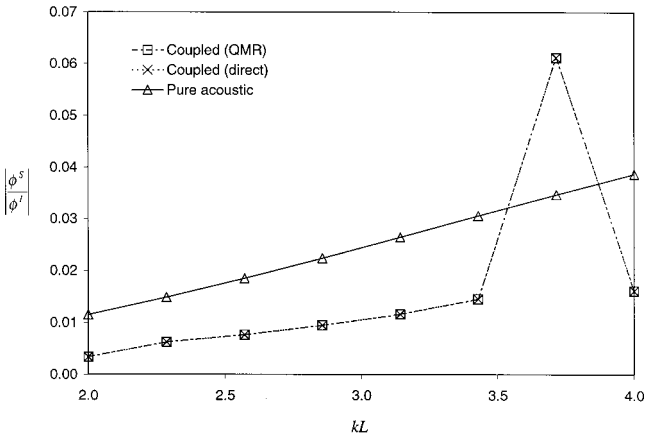

FIG. 13. Backward-scattering result at point (−35, 0, 0).

In addition, an effective preconditioner (scheme 5) specially designed for frequency-sweep analysis is presented. With this preconditioner, the iterative solver has been found to be stable in a frequency-sweep analysis. The scheme ensures convergence, and the CPU time consumption is much less than that of the direct solver in the case studied in this paper.

To further improve the efficiency of the developed BEM in analyzing even larger structural acoustic problems, methods to reduce the CPU time in the formation of the coefficient matrices should be explored. The formation time has been shown to become dominant in the whole BEM process with the use of the iterative solver (see, e.g., Fig. 2). This reduction can be achieved by using, for example, the multipole expansion method (see, e.g., Ref. 9) emerging recently in the BIE/BEM.

ACKNOWLEDGMENTS

The first two authors (S.H.C. and Y.J.L.) would like to acknowledge the partial support to the University of Cincinnati from Motorola, Inc. for this project. The authors thank the reviewers of this paper for their helpful comments.

[1] S. H. Chen and Y. J. Liu, ''A unified boundary element method for the analysis of sound and shell-like structure interactions: I. Formulation and verification,'' J. Acoust. Soc. Am. **106**, 1247–1254 (1999).
[2] R. Barrett, M. Berry, T. F. Chan, J. Demmel *et al.*, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed. (SIAM, Philadelphia, 1994).
[3] R. W. Freund and N. M. Nachtigal, ''QMRPACK: a package of QMR algorithms,'' ACM Trans. Math. Softw. **22**, 46–77 (1996).
[4] G. H. Golub and C. F. V. Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 1983).
[5] J. H. Kane, D. E. Keyes, and K. G. Prasad, ''Iterative solution techniques in boundary element analysis,'' Int. J. Numer. Methods Eng. **31**, 1511–1536 (1991).
[6] K. G. Prasad, J. H. Kane, D. E. Keyes, and C. Balakrishna, ''Preconditioned Krylov solvers for BEA,'' Int. J. Numer. Methods Eng. **37**, 1651–1672 (1994).
[7] J. E. Romate, ''On the use of conjugate gradient-type methods for boundary integral equations,'' Comput. Mech. **12**, 214–232 (1993).
[8] C. Y. Leung and S. P. Walker, ''Iterative solution of large three-dimensional BEM elastostatic analyses using the GMRES technique,'' Int. J. Numer. Methods Eng. **40**, 2227–2236 (1997).
[9] V. Rokhlin, ''Rapid solution of integral equations of scattering theory in two dimensions,'' J. Comput. Phys. **86**, 414–539 (1990).