Drop tolerance ILU preconditioners for iterative solution techniques in boundary element analysis

N.S. Jovanovic[a], D.E. Keyes[a], K. Guru Prasad[a], J.H. Kane[b]

[a]*Department of Mechanical Engineering, Yale University, New Haven, CT 06520, USA*

[b]*Department of Mechanical and Aeronautical Engineering, Clarkson University, Potsdam, NY 13699, USA*

Abstract

A drop tolerance preconditioning strategy for accelerating the convergence of iterative methods is tested on algebraic equations resulting from boundary element analysis of continuum structural and thermal response problems. The preconditioners are analogs of the drop tolerance incomplete LU (ILU) preconditioners commonly employed with Krylov methods in the solution of sparse linear systems. In all of the problems considered, preconditioned Krylov iteration is more efficient than a dense direct method, with an advantage that improves with problem size. In some of the problems, the optimal drop tolerance ILU (DTILU) preconditioning keeps more nonzeros than simple diagonal preconditioning, but in others it reduces to the diagonal. *A priori* determination of the optimal drop tolerance remains an open problem.

Introduction

Computational implementations of boundary element methods ultimately require the solution of a system of linear equations

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

as do implicit domain-based methods such as finite elements [1]. Whereas domain methods typically spawn large systems of locally coupled equations, resulting in sparse, banded matrices, A, solution of the same problem to

comparable discretization accuracy with boundary element methods generally yields smaller systems of globally coupled equations represented by dense A matrices. Solution of these algebraic systems can proceed via direct or iterative methods, the latter, until recently, more often associated with sparse systems [2, 3].

Iterative methods are also attractive, however, for solving dense systems of equations such as those arising from boundary element methods. Krylov methods such as the Generalized Minimal Residual (GMRES) method [4] have computational complexities that grow as $\mathcal{O}(N^2)$ for typical boundary element operators, whereas the complexity of a direct method grows as $\mathcal{O}(N^3)$, where N is the number of degrees of freedom. Krylov iterative methods typically require one or two matrix-vector multiplications at each iteration, and are guaranteed to generate the actual solution, in the absence of roundoff error, in at most N steps. The $\mathcal{O}(N^2)$ complexity estimate is based on the assumption that the number of iterations required to generate an acceptably accurate solution is independent of N, since each matrix-vector multiplication is already $\mathcal{O}(N^2)$. Previous studies by Kane, Keyes, and Prasad [5], Vavasis [6], Mansur *et al.* [7], and Barra *et al.* [8] demonstrate that this assumption is valid for many applications of the boundary element method when *preconditioning* is used in conjunction with an iterative method such as GMRES, a result which this study also confirms.

Equation (1) may be preconditioned by the formal transformation

$$\underbrace{B_L^{-1} A B_R^{-1}}_{\tilde{A}} \underbrace{B_R x}_{\tilde{x}} = \underbrace{B_L^{-1} b}_{\tilde{b}}. \tag{2}$$

That is,

$$\tilde{A}\tilde{x} = \tilde{b} \tag{3}$$

is solved for $\tilde{x}$, then

$$B_R x = \tilde{x} \tag{4}$$

for x. For right preconditioning $B_L = I$, and for left preconditioning $B_R = I$. For the studies of this paper, we prefer right preconditioning because it allows direct monitoring of the unpreconditioned residual for convergence testing. It should be noted that efficient implementations of equation (2) avoid matrix-matrix multiplications.

To accelerate the convergence of an iterative method at acceptable cost, a preconditioner B should possess two properties [10]:

- B approximates the system matrix A in some way, and

- solution of $Bz = r$, for unknown z, is subdominant, or at worst co-dominant, in the overall per-iteration computational complexity.

Choosing or developing a preconditioner that reduces the number of iterations necessary for convergence to suitable accuracy, without making the cost per iteration too great, involves trade-offs. The most effective method from the point of overall CPU time is often *not* the one requiring the fewest iterations.

Kane, Keyes and Prasad [5] investigated both GMRES and the Conjugate Gradient Normal (CGN) iterative methods, combined with diagonal, block diagonal, and truncated band ILU preconditioning, and demonstrated the competitiveness of iterative techniques with direct methods on a variety of boundary element applications in elastostatics. Mansur *et al.* [7] evaluated CGN, Biconjugate Gradient (Bi-CG), and Lanczos iterative methods, along with Richardson, Jacobi, and Gauss-Seidel preconditioning, using boundary element examples from elastostatics, fracture mechanics, and electric potential theory to do the same. Barra *et al.* [8] compared Bi-CG with GMRES, using diagonal preconditioning to solve boundary element problems in elastostatics and fracture mechanics. Urekew and Rencis [11] evaluated CGN, Jacobi, and Gauss-Seidel iterative methods for their study of boundary element formulations of mixed boundary value problems, using no preconditioning. Beyond traditional algebraic preconditioners, Vavasis [6] tested a variety of physically-motivated imposed-sparsity preconditioners.

This study adds to a growing body of evidence which supports the effectiveness of iterative methods in boundary element applications, by introducing an additional preconditioner family. The DTILU preconditioner is a simple modification to ILU. At one extreme, it is equivalent to a direct method, and, at the other, to diagonal preconditioning. Varying the user-specified drop tolerance allows continuous spanning of the intermediate fill regime. (See [9] for a discussion and bibliography in context of sparse matrices.) GMRES is chosen for the iterative solver because of its successes in the studies cited above, and because of its applicability to unsymmetric systems of equations, a necessary property for boundary element methods of collocation type.

Equation Solvers

Direct Method

The baseline direct method for the comparisons reported herein is the dense solver from the public domain LINPACK subroutine library [12]. The LIN-

PACK routines use Gaussian elimination with partial pivoting to compute the LU factorization of the system matrix. All computations were performed in double precision on a Sun Sparcstation IPC workstation.

Iterative Method

The commercial subroutine library PCGPAK2 [13] originally designed for the iterative solution of large, sparse systems of linear equations was modified to obtain the results for the preconditioned GMRES method. PCGPAK2 employs a modification of the compressed row storage format used in the Yale Sparse Matrix Package [14]. This format stores only the nonzero coefficients of a matrix, improving storage efficiency for sufficiently sparse matrices; it is used to store the preconditioner. The dense system matrix, however, is stored in coordinate format to avoid the inefficiency of storing and manipulating it in the sparse format.

GMRES This algorithm is started with an initial guess and calculates an improved solution once every k iterations, or whenever the convergence criteria have been met, whichever comes first. The restart parameter, k, is the predetermined maximum dimension of the Krylov subspace, K_k, which is used to build the solution vector. The Krylov subspace is defined by

$$K_k = span\{r_0, Ar_0, (A)^2 r_0, \ldots, (A)^{k-1} r_0\}, \tag{5}$$

where r_0 is the initial residual, and A is the system matrix. A is replaced by $\tilde{A}$, and r_0 by $B_L^{-1} r_0$ in the above definition when preconditioning is added.

In each set of k iterations, vectors $v_1, v_2, \ldots, v_k$, forming an orthonormal basis in K_k, are successively generated using a modified Gram-Schmidt procedure. The new approximation, $x_k = x_0 + y$, is obtained by choosing $y \in K_k$ such that the norm of the residual is minimized. Even though the new solution vector is calculated every k iterations, the left preconditioned residual can be inexpensively monitored at *every* step [4]. Previous experience with boundary element matrices in [5] suggests that k should be chosen larger than the expected number of iterations (effectively avoiding restarting GMRES), and that using a relative residual norm of 10^{-6} as a stopping criterion provides accuracy comparable to that of direct methods for our problem suite.

Preconditioning Strategies

If the preconditioner, B, is chosen to be the identity matrix, then the unpreconditioned GMRES method is obtained. If B is chosen to be A exactly, then GMRES becomes equivalent to a direct method, converging

in one iteration. Only in special cases is the preconditioner inverted explicitly; instead, the action of B^{-1} is accomplished, for example, via LU or ILU decomposition. The decomposition need be performed only once. Subsequent iterations then apply the inverse by performing a forward and backward substitution process.

Diagonal Preconditioning

This strategy sets

$$B = diag[A]. \tag{6}$$

Therefore,

$$(B^{-1})_{ii} = \frac{1}{b_{ii}} \tag{7}$$

Diagonal preconditioning has been one of the most successful preconditioners for solving the systems generated by boundary element methods. In this study, the term *scaling*, as it appears in data tables, will be reserved for left diagonal preconditioning and right diagonal preconditioning will be understood otherwise.

ILU Preconditioning

Incomplete LU decomposition preconditioning differs from a direct method in that the LU factors of B may be only approximately computed. This is accomplished by letting

$$B = \tilde{L}\tilde{U}, \tag{8}$$

where $\tilde{L} + \tilde{U}$ has predetermined sparsity, such as that of A. Since A is dense in boundary element methods, ordinary no-fill ILU preconditioning is equivalent to a direct method and it is natural to explore criteria for element selection not directly linked to the sparsity of A.

Row DTILU Preconditioning

In this strategy, a drop tolerance parameter, ϵ, chosen by the user, acts as a filter to introduce sparsity into the preconditioner. Thus, a coefficient of the system matrix, A, is passed into the preconditioner only if its magnitude is greater than a preset fraction of the magnitude of the coefficient on the main diagonal in its row. The drop tolerance filtering process,

$$a'_{ij} = \begin{cases} a_{ij} & \text{if } |a_{ij}| > \epsilon\,|a_{ii}|\,,\ j \neq i \\ a_{ii} & \text{if } j = i \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

forms an intermediate matrix approximation, A', containing only the larger coefficients of A. Because the kernels of our boundary integral operators arise from elliptic Green's functions, large magnitudes tend to correspond

to physically proximate couplings, but may not all be diagonally clustered because of column interchanges related to the choice of Dirichlet or Neumann boundary conditions. The introduced sparsity is then exploited by performing an ILU(0) decomposition of A' which allows no fill-in to occur. Thus,

$$B = \tilde{L}_{A'}\tilde{U}_{A'}, \tag{10}$$

where $\tilde{L}_{A'} + \tilde{U}_{A'}$ has the same sparsity structure as A'.

A tradeoff between iteration count and CPU time per iteration results, in some cases, in a region of optimal ϵ which minimizes total CPU time required to solve the linear systems. Prediction of this optimum drop tolerance appears difficult and problem dependent. In some cases, the optimum drop tolerance filters all of the coefficients except the diagonal; the minimizing ϵ lies on the boundary. In other cases, it lies in the interior.

Row-Column DTILU Preconditioning

The difference between this and the previous strategy is only in the method employed to filter the coefficients of A. Filtering is accomplished by letting

$$a'_{ij} = \begin{cases} a_{ij} & \text{if } |a_{ij}a_{ji}| > \epsilon^2 |a_{ii}a_{jj}|, \ j \neq i \\ a_{ii} & \text{if } j = i \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

We mention this particular method for completeness, but it was found to offer no advantages over Row DTILU preconditioning throughout the problem suite. That is, the optimal element selection produced by Row-Column DTILU over the full range of ϵ considered was never better than the optimal element selection produced by Row DTILU in the sense of CPU time.

Numerical Experiments

Numerical experiments were run on six examples. The 3–D problems were discretized into continuous, quadratic, isoparametric, triangular (6-noded) or rectangular (8-noded) elements, and the 2–D problems were discretized into continuous, quadratic, isoparametric (3-noded) elements. The matrices were created with boundary collocation. The maximum Krylov subspace dimension was set at 200, and the stopping criterion was a relative residual norm of 10^{-6}. Problem dimensions ranged from 126 (15,876 nonzeros) to 554 (306,916 nonzeros).

Cantilevered Bar Under Tension

The first problem is a 3–D cantilevered bar under uniform unit tension, with 20 triangular elements, 42 nodes, and 126 degrees of freedom. The bar has a square cross section and an aspect ratio of two. Figure (1) displays the results for DTILU-GMRES with no scaling, right diagonal preconditioned GMRES, and the direct method. (In these and all subsequent figures in which the ϵ scale is logarithmic, the point corresponding to the direct method ($\epsilon = 0$) is shown as the intersection of a dashed segment with the ordinate axis.) Table (1) reveals that optimal DTILU-GMRES ($\epsilon = 0.152$) required 25% less CPU time than the direct method and right or left diagonal preconditioning. This smallest example led to the smallest observed relative gain for GMRES.

Heat Conduction in Rectangular Region

A rectangular region with aspect ratio 12 was discretized into 104 elements and 208 nodes. The long sides of the rectangle were insulated, one end was fixed at zero degrees, and the other end fixed at 1000 degrees. The results are displayed in Figure (2) and Table (2). This extremely simple problem shows that the optimal drop tolerance is sometimes equivalent to diagonal preconditioning. Diagonally preconditioned GMRES used 63% less CPU time than the direct method. Also note that even unpreconditioned GMRES outperformed the direct method.

Cube Under Tension

A cube under uniform uniaxial applied tension was discretized into 24 quadrilateral elements, 74 nodes, and 222 degrees of freedom. Results for this example are shown in Figure (3) and Table (3). This example demonstrates an extremely sharp threshold value for the drop tolerance at $\epsilon \approx 0.02$, below which GMRES is much less efficient than the direct method, and above which GMRES is more efficient than the direct method. As in the previous example, DTILU-GMRES is practically equivalent to diagonally preconditioned GMRES, and unpreconditioned GMRES outperformed the direct method. The preconditioned iterative methods require 68% less CPU time than the direct method.

Rectangular Region Under Tension

A rectangular region with aspect ratio 12 subjected to uniform uniaxial tension was discretized into 104 elements, 208 nodes, and 416 degrees of freedom. Figure (4) demonstrates that CPU time has a local minimum at $\epsilon \approx 0.015$ but the global minimum is at the diagonal limit. Table (4) reveals that diagonally preconditioned GMRES requires 84% less CPU

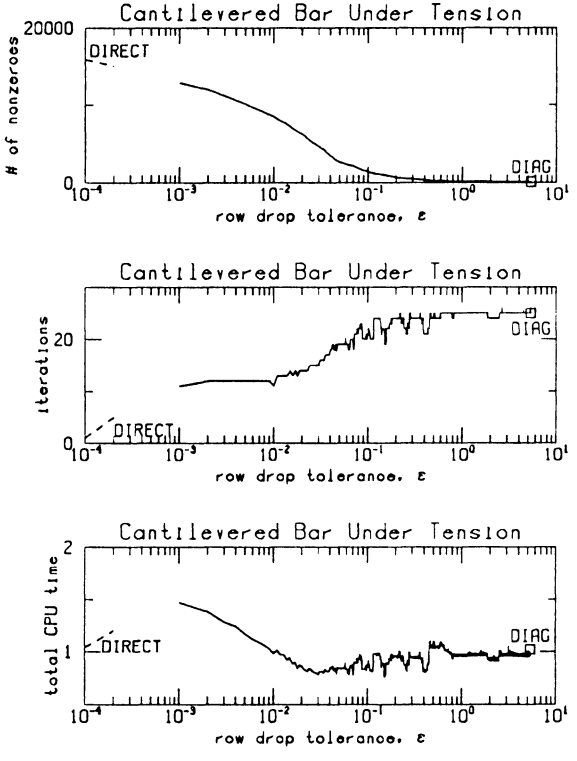

Figure 1: The number of nonzero coefficients used in the preconditioner, iteration count, and total CPU time required for convergence as functions of the row drop tolerance for the cantilevered bar problem.

scaling	method	$N_{A'}$	iterations	CPU seconds
no	direct	—	—	1.04
no	GMRES	—	69	3.06
yes	GMRES	—	24	1.02
no	diag-GMRES	126	25	1.02
no	DTILU-GMRES	969	19	0.76
yes	DTILU-GMRES	2290	16	0.81

Table 1: Statistics for the cantilevered bar problem.

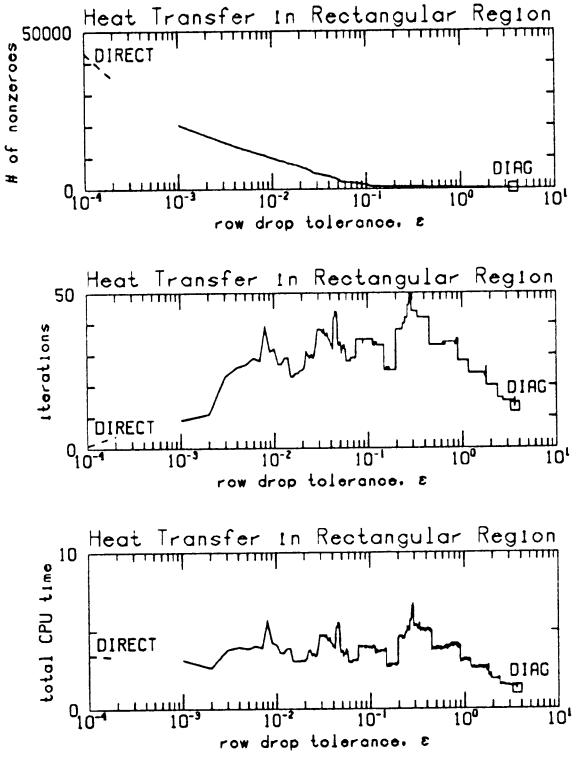

Figure 2: The number of nonzero coefficients used in the preconditioner, iteration count, and total CPU time required for convergence as functions of the row drop tolerance for the rectangular heat conduction problem.

scaling	method	$N_{A'}$	iterations	CPU seconds
no	direct	—	—	3.47
no	GMRES	—	22	2.09
yes	GMRES	—	12	1.34
no	diag-GMRES	208	13	1.26
no	DTILU-GMRES	208	13	1.26
yes	DTILU-GMRES	208	12	1.49

Table 2: Statistics for the rectangular heat conduction problem.

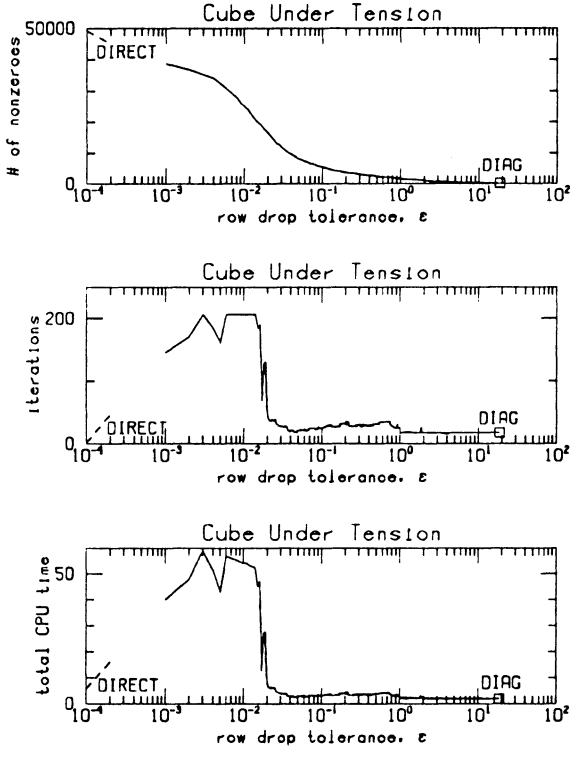

Figure 3: The number of nonzero coefficients used in the preconditioner, iteration count, and total CPU time required for convergence as functions of the row drop tolerance for the cube problem.

scaling	method	$N_{A'}$	iterations	CPU seconds
no	direct	—	—	5.71
no	GMRES	—	31	3.36
yes	GMRES	—	17	2.03
no	diag-GMRES	222	17	1.83
no	DTILU-GMRES	230	17	1.85

Table 3: Statistics for the cube problem.

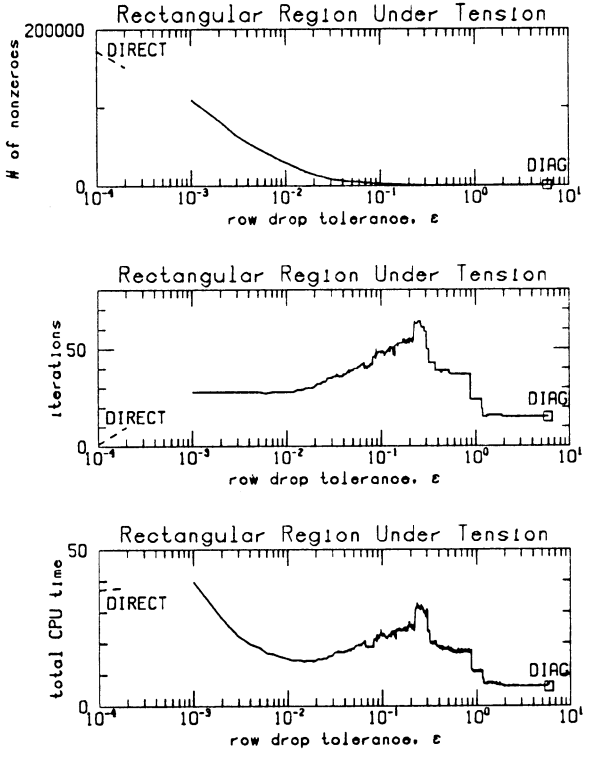

Figure 4: The number of nonzero coefficients used in the preconditioner, iteration count, and total CPU time required for convergence as functions of the row drop tolerance for the rectangular elasticity problem.

scaling	method	$N_{A'}$	iterations	CPU seconds
no	direct	—	—	37.18
no	GMRES	—	27	10.65
yes	GMRES	—	14	6.28
no	diag-GMRES	416	15	5.93
no	DTILU-GMRES	422	15	6.14

Table 4: Statistics for the rectangular elasticity problem.

time than the direct method. Once again, even unpreconditioned GMRES outperforms the direct method.

Pressurized Hollow Cylinder

This example is a quarter symmetry hollow cylinder subjected to internal pressure, with 76 triangular elements, 154 nodes, and 462 degrees of freedom. As in the first example, optimal DTILU-GMRES outperforms the other methods examined, as shown in Figure (5) and Table (5). The optimal drop tolerance occurs at $\epsilon = 0.07$, but actually, the graph of CPU time displays a relatively flat shape in the region $\epsilon \in [0.015, 0.15]$. Choosing any drop tolerance within this range allows DTILU-GMRES to outperform diagonally preconditioned GMRES. Optimal DTILU-GMRES uses 83% less CPU time than the direct method, and 32% less CPU time than diagonally preconditioned GMRES. The direct method outperformed unpreconditioned GMRES in this example.

Heat Conduction in Hollow Sphere

The conduction of heat through a hollow sphere is modeled with one-eighth symmetry using 276 triangular elements, and 554 nodes. The inner and outer surface temperatures are held constant at 100 and 1000 degrees, respectively. Figure (6) and Table (6) reveal results similar to those of the previous example. DTILU-GMRES is optimal at $\epsilon = 0.095$, uses 89% less CPU time than the direct method, and 26% less than diagonally preconditioned GMRES. This relatively large problem exhibited the greatest savings for GMRES over direct solution. Note that the cost curves in Figures (1)–(6) become smoother as problem size increases and the boundary element meshes become finer.

Concluding Remarks

Several iterative solution techniques for solving the dense, unsymmetric systems of equations which derive from the boundary element method have been presented and tested on a small number of examples from elastostatics and heat conduction. In each example, preconditioned iterative methods outperformed the direct method in terms of the CPU time required for solution. The relative (and, *a fortiori*, absolute) advantage of the preconditioned iterative methods over the direct method is observed to grow with the problem size.

The DTILU preconditioners establish a continuum between the direct method, on one hand, and diagonal preconditioning on the other. Figures (1)–(6) demonstrate that a tradeoff exists between iteration count and the

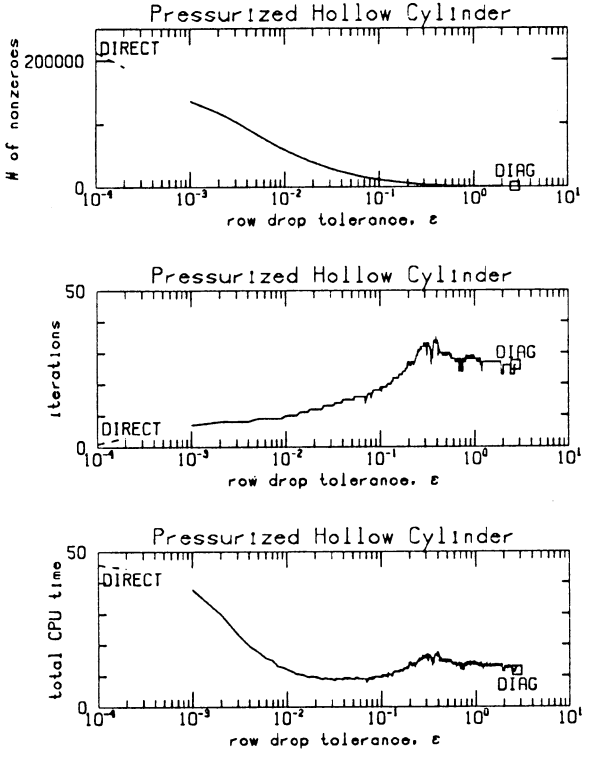

Figure 5: The number of nonzero coefficients used in the preconditioner, iteration count, and total CPU time required for convergence as functions of the row drop tolerance for the hollow cylinder problem.

scaling	method	$N_{A'}$	iterations	CPU seconds
no	direct	—	—	45.74
no	GMRES	—	128	62.96
yes	GMRES	—	25	11.89
no	diag-GMRES	462	26	11.49
no	DTILU-GMRES	16036	14	7.86
yes	DTILU-GMRES	29635	11	8.47

Table 5: Statistics for the hollow cylinder problem.

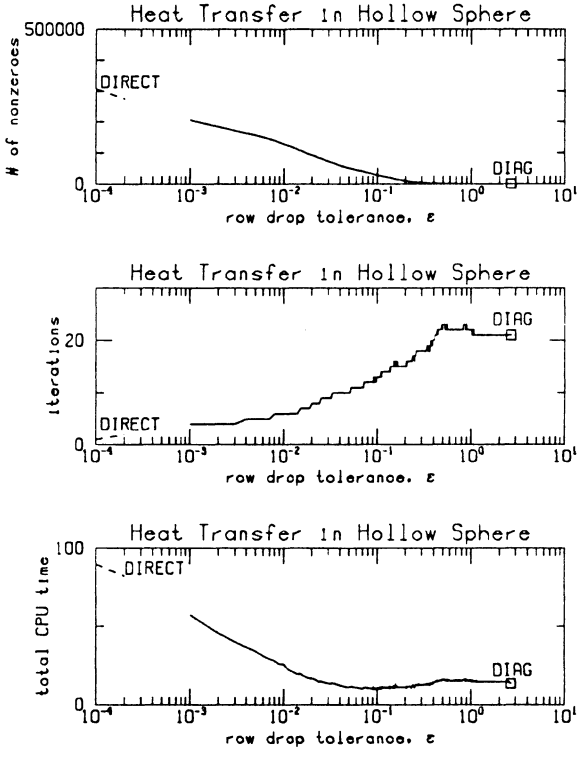

Figure 6: The number of nonzero coefficients used in the preconditioner, iteration count, and total CPU time required for convergence as functions of the row drop tolerance for the hollow sphere problem.

scaling	method	$N_{A'}$	iterations	CPU seconds
no	direct	—	—	89.70
no	GMRES	—	28	17.59
yes	GMRES	—	21	14.46
no	diag-GMRES	554	21	13.28
no	DTILU-GMRES	30035	12	9.82
yes	DTILU-GMRES	29375	12	11.33

Table 6: Statistics for the hollow sphere problem.

per-iteration CPU time. This tradeoff makes possible the existence of an optimal drop tolerance which minimizes the total CPU time required to solve a system of equations arising from an application of boundary element methods. In three of the example problems, DTILU-GMRES outperformed both diagonally preconditioned GMRES and the direct method when the optimum drop tolerance was used. However, the other three problems showed the optimal DTILU preconditioning essentially to coincide with diagonal preconditioning. It remains an open question whether a procedure for *a priori* determination of the optimal drop tolerance can be found. Considering that the meaningful range of ϵ for these problems extends from orders of magnitude as small as 10^{-23} (the first rejected element) to as large as 10^{+2} (the last rejected element), it is encouraging that the optimal drop tolerances, when they differ from infinity (the diagonal case), seem to reside in a relatively narrow bandwidth between approximately 10^{-2} and 10^{-1} in the unscaled formulations.

Acknowledgements

This work has been supported in part by grants from the U. S. National Science Foundation (DDM-9020733) to Yale and Clarkson Universities. In addition, the first author receives support from the U. S. Department of Energy through the Computational Science Graduate Fellowship Program. Test code development was made possible through additions and modifications to the PCGPAK2 software package licensed without cost to the authors by Scientific Computing Associates, Inc., New Haven, CT 06510, USA.

References

[1] Brebbia, C. A., *The Boundary Element Method for Engineers*, Pentech Press, London; Halstead Press, New York, 1978.

[2] Golub, G. H. and Van Loan, C. F., *Matrix Computations, 2nd ed.*, Johns Hopkins University Press, Baltimore and London, 1989.

[3] Dongarra, J. J., Duff, I. S., Sorensen, D. C. and van der Vorst, H. A., *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, 1991.

[4] Saad, Y. and Schultz, M. H., 'GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,' *SIAM J. Sci. Stat. Comput.*, **7**:856–869, 1986.

[5] Kane, J. H., Keyes, D. E. and Prasad, K. Guru, 'Iterative Solution Techniques in Boundary Element Analysis,' *Int. J. Numer. Meth. Eng.*, **31**:1511–1536, 1991.

[6] Vavasis, S. A., 'Preconditioning for Boundary Integral Equations,' *SIAM J. Matrix Anal. Appl.*, **13**:905–925, 1992.

[7] Mansur, W. J., Araújo, F. C. and Malaghini, J. E. B., 'Solution of BEM Systems of Equations via Iterative Techniques,' *Int. J. Numer. Meth. Eng.*, **33**:1823–1841, 1992.

[8] Barra, L. P. S., Coutinho, A. L. G. A., Mansur, W. J. and Telles, J. C. F., 'Iterative solution of BEM Equations by GMRES Algorithm,' *Comp. Struct.*, **44**:1249–1253, 1992.

[9] Saad, Y., 'ILUT: A Dual Threshold Incomplete LU Factorization,' Tech. Rep. UMSI 92/38, University of Minnesota Supercomputer Institute, Minneapolis, March, 1992.

[10] Ortega, J. M. and Voigt, R. G., *Solution of Partial Differential Equations on Vector and Parallel Computers*, SIAM, Philadelphia, 1985.

[11] Urekew, T. J. and Rencis, J. J., 'An Iterative Solution Strategy for Boundary Element Equations from Mixed Boundary Value Problems,' *Comp. Meth. in Appl. Mech.*, (submitted for review).

[12] Dongarra, J. J., Moler, C. B., Bunch, J. R. and Stewart, G. W., *LINPACK Users' Guide*, SIAM, Philadelphia, 1979.

[13] *PCGPAK2 User's Guide (Version 1.0)*, Scientific Computing Associates, Inc., New Haven, CT 06510, USA, August 1989.

[14] Eisenstat, S. C., Gursky, M. C., Schultz, M. H. and Sherman, A. H., 'Yale Sparse Matrix Package. II. The Nonsymmetric Codes,' Tech. Rep. 114, Yale University Department of Computer Science, New Haven, CT, 1977.