JOURNAL OF COMPUTATIONAL PHYSICS **133,** 186–191 (1997)
ARTICLE NO. CP975661

NOTE

An Efficient and Robust Spectral Solver for Nonseparable Elliptic Equations

1. INTRODUCTION

The objective of this note is to demonstrate the computational efficiency and robustness of a preconditioned biconjugate gradient spectral solution of a nonseparable elliptic equation. We accomplish this by evaluating the performance of the proposed scheme based on the BiCGstab(l) algorithm [1], with a spectral preconditioner based on an iterative scheme proposed by Concus and Golub [2] and a fast direct spectral solver for Helmholtz equations with constant coefficients [3], for different functional coefficients and different test solutions, among the most complicated ones which have appeared in the literature. For comparison purposes, the results obtained with two other schemes are also reported here.

The nonseparable elliptic equation that is considered in this work is a modified Helmholtz equation with a nonconstant coefficient $g(x, y)$,

$$\frac{\partial^2 P}{\partial x^2} + M\frac{\partial^2 P}{\partial y^2} - g(x, y)P = f(x, y), \tag{1}$$

where $f(x, y)$ is an arbitrary function, M is a constant and $x, y \in [-1, 1]$. Although the method discussed in this work can be applied with any boundary conditions, for illustrative purposes we focus our attention to periodic boundary conditions along the x-direction and Dirichlet, Neumann along the $y = -1, y = 1$ boundaries, respectively:

$$P(-1, y) = P(1, y),$$

$$\left.\frac{\partial P}{\partial x}\right|_{(-1,y)} = \left.\frac{\partial P}{\partial x}\right|_{(1,y)}, \tag{2}$$

$$P(x, -1) = 0, \quad \left.\frac{\partial P}{\partial y}\right|_{(x,1)} = 0. \tag{3}$$

Correspondingly, a Fourier and Chebyshev expansion is utilized along the x and y directions, respectively. Since, for a general function $g(x, y)$, a fast direct Helmholtz spectral solver is not available, an iterative method needs to be used for an efficient spectral solution, such as conjugate gradient or multigrid [3].

Equation (1) arises during the numerical simulation of flows in certain domains such as within an undulating channel, when an orthogonal conformal-like mapping [4] of the original domain (x', y') to a rectangular domain (x, y) is utilized, where $M = h_x^2/h_y^2$ is the constant ratio of the two shape factors of the mapping and $g(x, y) = h_x^2$. Equation (1) is also a rescaled form of a generalized Helmholtz equation [2]:

$$\nabla^2 v - s(x, y)v = r(x, y). \tag{4}$$

Generalized Helmholtz equations arise frequently in fields such as optics [5], geophysics [6], and plasma physics [7]. In addition, nonseparable elliptic equations of the form

$$\nabla \cdot (a(x, y)\nabla u) - b(x, y)u = c(x, y), \tag{5}$$

can also be transformed to the form of a generalized Helmholtz equation (4) through a change of variable $v = a^{1/2}u$, when $a(x, y)$ is positive in the domain of definition [2].

Most of the early work on the efficient solution of nonseparable elliptic equations involved finite difference approximations (especially second order). Initially general iterative approaches like SOR, Richardon iteration, and ADI [8] were used. For a properly selected relaxation parameter value, the ADI method is proven to converge in a number of iterations which is constant with the mesh size [9–11]. However, this number is small (often as low as 3 or 4) only for $g(x, y)$ constant. For an arbitrary $g(x, y)$ the number of iterations required to achieve a certain error can increase to unacceptably high values. More recently, iterative techniques have been proposed that involve preaveraging of the function $g(x, y)$ in one [2] or two [12] directions, where fast solvers can be used, based on cyclic reduction and FFT methods, respectively. The performance of these methods is better than the ADI, but still a large number of iterations can be required.

The number of iterations can decrease substantially if any one of the above-mentioned iterative techniques are used as preconditioners in conjunction with a conjugate

0021-9991/97 $25.00

gradient method [13]. In particular, Elman and Schultz [14] applied a conjugate gradient scheme which utilizes as a preconditioner an approximate, but separable, problem which can be solved directly. They saw a significant reduction in the number of iterations, especially when they used partial averaging (along one direction only) of the variable coefficients in the elliptic equation under investigation.

However, finite difference solutions converge slowly with mesh refinement, the error decreasing as a low-order power of the number of mesh points used. In contrast, spectral methods converge exponentially fast, the error decreasing exponentially with the number of modes involved. This drives the interest to utilize spectral methods in conjunction with an efficient solver. As with finite differences, iterative techniques such as that proposed by Concus and Golub [2] can be formulated. More recently Zhao and Yedlin [6] have solved pseudospectrally the equation $\nabla \cdot (a\nabla u) = f$, by rewriting it as $\nabla^2 u = (f - \nabla a \cdot \nabla u)/a$ and iterating. However, they have only presented results for $a(x, y)$ which exhibits only mild x, y dependence.

As with finite differences, conjugate gradient methods appear to be the most efficient and they have been proposed to be used with spectral methods involving either finite difference [3], or finite element preconditioners [15], but with mixed results. Spectral preconditioners seem to be more appropriate. Guillard and Desideri [16] proposed two spectral preconditioners, which correspond to separable operators, along with a minimal residual (MR) scheme that has been used with finite difference preconditioning in the past [3]. However, for the first type of preconditioner, the number of iterations can increase significantly with increasing $g(x, y)$ variability and mesh size, whereas for the second preconditioner, for which the number of iterations is always small (less then 10), the computation of the eigenvectors of the elliptic operators is required, making that scheme much more demanding in computational power. More recently, Strain [17] has used a separable operator as a preconditioner to a generalized minimum residual scheme (GMRES) to solve a fully periodic problem spectrally. However, from the results presented, it was clear that the method needed a large number of iterations to converge (sometimes more than 100). Thus, the efficiency and robustness of such a method has not yet been convincingly demonstrated.

In this work, we have investigated the performance of the numerical solution of (1) subject to the boundary conditions (2) and (3) by several algorithms: First, using the iterative method due to Concus and Golub [2] and a spectral fast Helmholtz solver; second, applying the BiCGstab(l) method using as preconditioners either an ADI scheme based on fourth-order finite differences, or the iterative spectral method discussed above. Two sets of M and $g(x, y)$ and several right-hand sides $f(x, y)$ were used. Specifically, the first set of M and $g(x, y)$ corresponds to the Helmholtz equation following the orthogonal mapping of an undulating channel to a unit square. The second set was selected based on the analysis in [2], such that the Concus and Golub scheme exhibits a slow rate of convergence.

2. METHODOLOGY

2.1. *Concus and Golub Method*

Concus and Golub [2] have proposed an iterative scheme which uses fast direct solvers for the repeated solution of a Helmholtz problem with constant coefficients:

$$\left(\frac{\partial^2}{\partial x^2} + M\frac{\partial^2}{\partial y^2} - K\right) P_{n+1} = (g(x, y) - K)P_n + f(x, y), \tag{6}$$

subject to the same boundary conditions (2) and (3). In Eq. (6) K is a free parameter, which usually has the so-called min–max value,

$$\tfrac{1}{2}(\min(g(x, y)) + \max(g(x, y))), \tag{7}$$

but which can be optimized for higher rates of convergence. In the past it has been demonstrated using finite differences [2] that the number of necessary iterations can vary dramatically, depending on the function $g(x, y)$ which has a critical role on the rate of convergence of (6). The smoother $g(x, y)$ is, the faster the rate of convergence. However, it has not yet been utilized in conjunction with a spectral solution.

The efficiency of Concus and Golub's method can be increased by extending its formulation to accommodate the use of a parameter K which is a one-dimensional function instead of a constant [12]. Unfortunately, for a spectral solution, that would have required a prohibitively large computational time since there is no fast Helmholtz solver available for variable coefficients. Thus, the best way to improve the spectral solution's efficiency is to use a conjugate gradient scheme such as BiCGstab(l), with this iterative procedure as a preconditioner, as explained at the end of Section 2.2 below.

2.2. BiCGstab(l) *Method*

Conjugate gradient methods are probably the most popular iterative techniques for solving systems of linear equations. They are often referred to as subspace iteration methods, since they solve a system of linear equations $A \cdot x = b$ by minimizing quadratic functionals in Krylov subspaces, which are spanned by a series of vectors generated by repeated multiplication by A. From the plethora

of such methods that have been developed, of interest is the recently proposed by Sleijpen and Fokkema [1] BiCGstab(l), which is a generalization of an earlier algorithm (BiCGstab) by Van der Vorst [18]. This algorithm overcomes some shortcomings of BiCGstab, by combining the BiCG algorithm with the GMRES(l) algorithm [1]. The parameter l is the degree of the minimum residual polynomial used in the algorithm. Increasing the value of l can make the algorithm more accurate, but at the expense of computational cost, due to the $2l$ matrix–vector multiplications required. As with the original BiCGstab algorithm, products involving $A^{\mathrm{T}} \cdot x$ are not required. The only matrix–vector products $A \cdot x$ appearing in this algorithm are evaluated efficiently in $\mathcal{O}(N \log_2 N)$ operations directly from the spectral residuals through fast Fourier transforms. However, as with all conjugate gradient methods, a preconditioner needs to be used for rapid convergence [3]. Finite difference and spectral preconditioners were considered in this work.

The first preconditioner considered, is a fourth order finite difference solution of (1) obtained by the ADI method [9–11]. The finite difference discretization is obviously defined on the same type of grid as the spectral method. The ADI method is based on the principle of operator splitting, where the problem is discretized in each direction separately and a series of one-dimensional problems are solved in alternating order. A relaxation parameter is also used in the form of an effective time Δt, $u^{\mathrm{new}} = u^{\mathrm{old}} + \Delta t$(residual equation), which serves to stabilize and accelerate the convergence of the numerical scheme. This parameter is important and it needs to be optimized, together with the number of iterations within the ADI scheme (which does not need to converge fully in the preconditioning step). The optimum for each specific problem is found by trial and error, taking into account the fact that the number of iterations in the biconjugate gradient algorithm should remain relatively low so that the generated vectors do not start losing their orthogonality due to truncation error.

The applicability of the Concus and Golub iterative procedure as a spectral preconditioner is also considered here. As in the ADI finite difference case, it was found that the internal iterations corresponding to the Concus and Golub preconditioner also do not need to converge in order to reach optimum performance (minimum CPU time). Indeed, no more than two iterations within the preconditioner were found necessary in the test cases presented here, which also implies that K need not always have the optimum value for faster convergence.

2.3. *CPU Requirements*

The CPU requirements are proportional to the number n of conjugate gradient iterations and the characteristic parameter l of the BiCGstab(l) algorithm. The FFTs require an $\mathcal{O}(N \log_2 N)$ number of operations and both the finite difference ADI and the fast spectral Helmholtz solver an $\mathcal{O}(N)$, where N is the number of unknowns. Both terms also exist for the Concus and Golub and the BiCGstab(l) algorithms so that the overall CPU time in any case can be considered approximately as $nl((\alpha k + \beta)N \log_2 N + (\gamma k + \delta)N)$, where k is the number of iterations within the preconditioner and α, β, γ, δ small integer numbers $\mathcal{O}(10)$, depending on the number of FFT calls and vector operations needed per preconditioner step and within the BiCGstab(l) algorithm. Thus, it is evident that the algorithms have almost linear scalability with the number of unknowns, with the proportionality factor roughly proportional to nlk, provided the parameters α, β, γ, δ are independent on the mesh size and $\beta \leq \alpha$, $\delta \leq \gamma$, as it was found in all examples examined here.

3. RESULTS AND DISCUSSION

The results presented here correspond to two different sets for the function $g(x, y)$ and the parameter M in (1). In the first set, $g(x, y) = g1(x, y) \equiv h_x^2$ which is relatively smoothly varying, where h_x is the shape factor resulting from the orthogonal mapping of an undulating channel defined as $x' \in [-1, 1]$ and $y' \in [-1, 1 - 0.5 \cos x']$ to the square $x \in [-1, 1]$ and $y \in [-1, 1]$. The range of values of $g1(x, y)$ is from 0.016 to 8.5 approximately with $M = M1 \approx 6.7$. In the second set,

$$g2(x, y) \equiv \frac{-8\pi \sin(2\pi(x + y))}{\frac{3}{2} + \sin(2\pi(x + y))}, \tag{8}$$

with $M2 = 1$, and was patterned from a function in Concus and Golub [2], corresponding to a case for which their algorithm exhibited very slow convergence.

Equation (1) was solved for various right-hand sides $fi(x, y)$ which correspond to the following solutions $Pi(x, y)$, $i = 1 - 4$, with respect to which the maximum absolute error of the numerical solution was calculated and reported:

$$\begin{aligned}
P1(x, y) &\equiv (y^2 - 1)(y - 1)\sin(\pi x), \\
P2(x, y) &\equiv (y^2 - 1)(y - 1)\,(e^{y}\sin(\pi x) \\
&\quad + e^{-y}\cos(\pi x)), \\
P3(x, y) &\equiv (y^2 - 1)(y - 1)\,(e^{y}\sin(2\pi x) \\
&\quad + e^{-y}\cos(2\pi x)), \\
P4(x, y) &\equiv (y^2 - 1)(y - 1)\,(e^{5y}\sin(4\pi x) \\
&\quad + e^{-5y}\cos(4\pi x)).
\end{aligned} \tag{9}$$

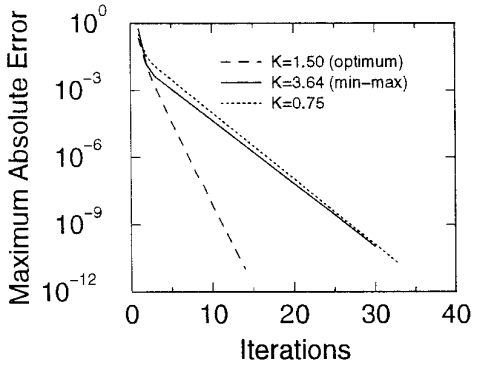

FIG. 1. Effect of the parameter K in Concus and Golub's method, for $g1(x, y)$, $f4(x, y)$, and 32×33 mesh. (The error is calculated with respect to the exact solution.)

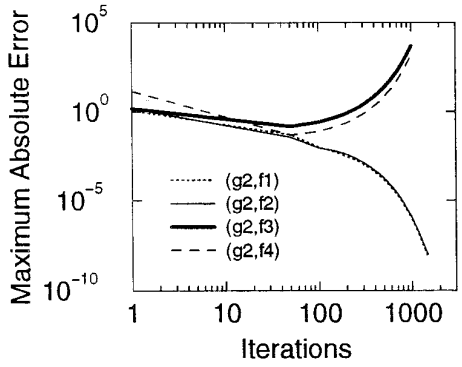

FIG. 2. Rate of convergence of Concus and Golub's method for $g2(x, y)$, K = min − max value and various right-hand sides for a 32×33 mesh.

The spectrally preconditioned BiCGstab(l) algorithm's performance was also tested for right-hand sides corresponding to the solution,

$$P5(x, y) \equiv \frac{(y^2 - 1)(y - 1)}{y^2 \cos(4\pi x) + 1.1a}, \tag{10}$$

where a is a parameter. The difficulty of the spectral approximation of this solution increases sharply as a approaches 1/1.1, at which value $P5(x, y)$ becomes singular.

The convergence criterion used was that the maximum absolute difference within the computational mesh of two successive iterates becomes smaller than 10^{-10}. This L_∞ criterion is effective at high convergence rates, as those encountered in the preconditioned BiCGstab(l) method. All runs were performed on a IBM RS6000/39H workstation, which has a typical performance of 130Mflops in linear algebra calculations. The mesh resolutions considered ranged from 8×9 to 512×513, but the main range of interest was from 32×33 to 128×129.

First, it needs to be mentioned that the number of conjugate gradient iterations required was observed to be essentially constant with varying mesh size to within 1–2 iterations. Thus, the interest is focused on how that number changes as the method or the problem difficulty varies.

Figures 1 and 2 correspond to Concus and Golub's iterative scheme using a fast direct spectral solver [3]. It is evident from the first figure, that for this method the parameter K determines its efficiency. From Fig. 1, one can see that the suggested min–max value $K = 3.64$ results for the same error in twice as many iterations for convergence as the optimum value $K = 1.5$. Additionally, from Fig. 2 one can see that for $g(x, y) = g2(x, y)$ this method does not converge for all right-hand sides and, for those for which it converges, a very large number of iterations are required. This behavior does not depend on the resolution and, therefore, any mesh increase does not alter the form of Fig. 2. The applicability of such an algorithm is therefore limited.

The BiCGstab(l) method with ADI finite difference preconditioning exhibits greater applicability since it converges in every case. It has however, three shortcomings. As it can be seen in Fig. 3, for large mesh sizes N it requires a substantial amount of time to converge. It also shows a dependence on $g(x, y)$ and is slower for strongly varying functions. This can also be seen from Fig. 4. However, the method is stable since, as Fig. 4 shows, the iterations can be carried on even after the error has reached the levels of truncation error, without the scheme diverging. However, for this method the truncation error was usually of the order of 10^{-10}, substantially higher than the machine accuracy for double precision calculations.

The BiCGstab(l) method with spectral preconditioning had the best performance, as it is evident from Figs. 5 and 6. The time needed for the method to converge is much less than when finite difference ADI preconditioning is used, and the dependence of the rate of convergence on $g(x, y)$ is also less. Additionally, its rate of convergence is

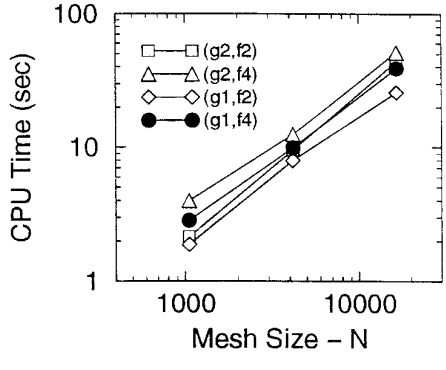

FIG. 3. Performance of BiCGstab(l) with ADI preconditioning with respect to the mesh size.

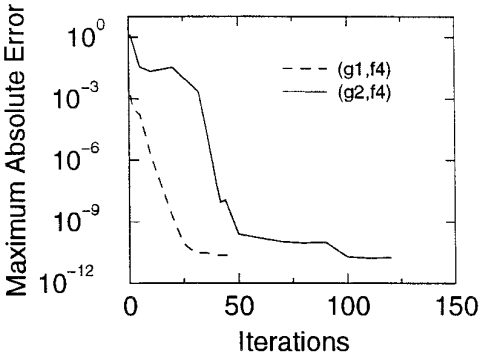

FIG. 4. Rate of convergence of BiCGstab(l) with ADI preconditioning for a 32 × 33 mesh.

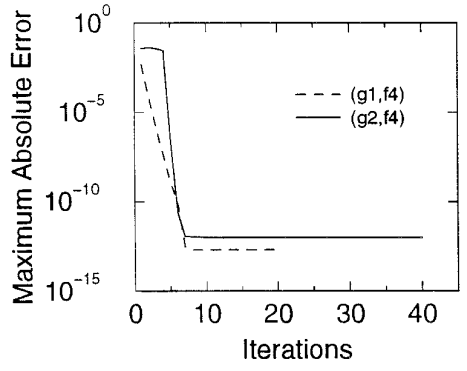

FIG. 6. Rate of convergence of BiCGstab(l) with spectral preconditioning for a 32 × 33 mesh.

much greater and the truncation error is of the order of machine precision. As with the other methods, there is no dependence of the number of iterations on the mesh size and that number is much less than that for the ADI preconditioner (compare Figs. 4 and 6). Moreover, it is exceptionally stable, as it can converge for a machine zero value of iteration updates without any problems, without even the few fluctuations observed in Fig. 4 for the ADI preconditioner. As Fig. 7 shows, with this solution method we can clearly observe the exponential convergence of the approximation (until truncation error levels are reached).

The performance of BiCGstab(l) with spectral preconditioning was further tested for functions which are very difficult to approximate spectrally, such as $P5(x, y)$ for different values of the parameter a. Figure 8 shows that the time needed to converge with respect to the mesh size is of the same order as in Fig. 5. Finally, Fig. 9 shows the exponential rate of convergence of the method. For a = 2 and a = 1, larger meshes (256 × 257 and 512 × 513, respectively) were required to reach machine accuracy (not shown in Fig. 9), but the rate with which the error decreases with mesh refinement remains exponential, even at these higher error values. It must be noted that this method was also tested successfully for the functions $g(x, y)$ used in [16], for which a large amount of iterations were needed for convergence.

With all the functions used, the spectrally preconditioned BiCGstab(l) requires at most nine iterations for convergence. The time per iteration varies with the characteristic parameter l and the number of internal iterations in the preconditioner k, but remains roughly 3–18 times that required for a fast Helmholtz solver solution equivalent to one step in the Concus and Golub scheme.

4. CONCLUSIONS

Three different spectral iterative solvers based on spectral approximations for generalized Helmholtz equations have been presented. The comparison between them has shown that the BiCGstab(l) algorithm with spectral preconditioning is undisputably the best method examined here, in terms of accuracy, efficiency and stability. It can be classified as a mixed spectral method, since it uses a Chebyshev-tau approximation for the linear terms and the

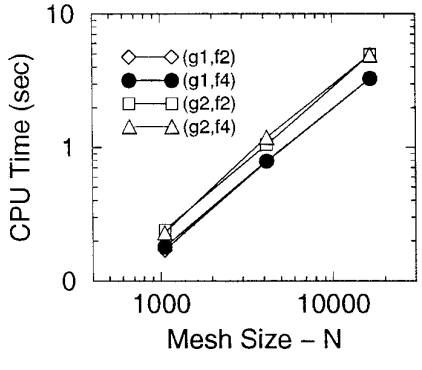

FIG. 5. Performance of BiCGstab(l) with spectral preconditioning with respect to the mesh size.

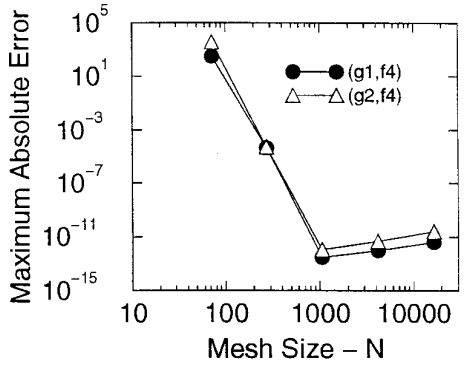

FIG. 7. Exponential convergence of BiCGstab(l) with spectral preconditioning with increasing resolution.

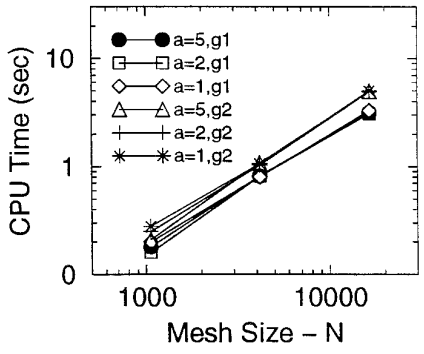

FIG. 8. Performance of BiCGstab(l) with spectral preconditioning with respect to the mesh size for $f5(x, y)$.

preconditioner and a Chebyshev collocation for the nonlinear terms. It is capable of providing with machine precision accuracy solutions in an extremely efficient manner due to its exponential convergence characteristics. The method's performance was successfully tested in problems of varying difficulty and generality, incorporating a mix of all possible boundary conditions (Dirichlet, Neumann, and periodic). It is therefore anticipated to perform well in a variety of physical applications.

ACKNOWLEDGMENTS

The authors acknowledge the financial support provided by the Office of Naval Research under Grant N00014-94-1-0581. The contribution of Dr. K. S. Chae during the early stages of this work is also gratefully acknowledged.

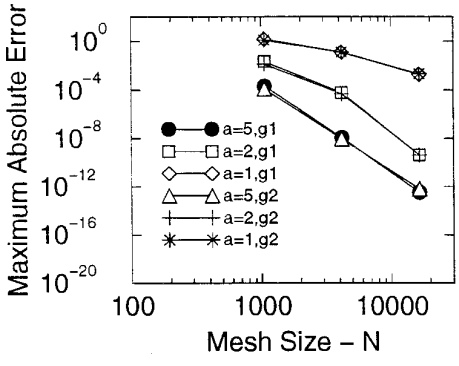

FIG. 9. Exponential convergence of BiCGstab(l) with spectral preconditioning with increasing resolution for $f5(x, y)$.

REFERENCES

1. G. L. G. Sleijpen and D. R. Fokkema, *Electron. Trans. Numer. Anal.* **1**, 11 (1993).
2. P. Concus and G. H. Golub, *SIAM J. Numer. Anal.* **10** (6), 1103 (1973).
3. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, 2nd ed. (Springer-Verlag, Berlin, 1992).
4. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *J. Comput. Phys.* **47,** 1 (1982).
5. G. R. Hadley, *Opt. Lett.* **19** (2), 84 (1994).
6. S. Zhao and M. Yedlin, *J. Comput. Phys.* **113,** 215 (1994).
7. D. W. Hewett, D. J. Larson, and S. Doss, *J. Comput. Phys.* **101,** 11 (1992).
8. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran*, 2nd ed. (Cambridge Univ. Press, Cambridge, 1992).
9. D. W. Peaceman and H. H. Rachford Jr., *J. Soc. Indust. Appl. Math.* **3** (1), 28 (1955).
10. E. G. D' Yakonov, *Dokl. Akad. Nauk SSSR* **138,** 522 (1961).
11. J. E. Gunn, *Numer. Math.* **6,** 181 (1964).
12. W. M. Pickering and P. J. Harley, *Intern. J. Computer Math.* **55**, 211 (1995).
13. D. J. Evans (Ed.), *Preconditioning Methods: Analysis and Applications*, Gordon & Breach, New York, 1983.
14. H. C. Elman and M. H. Schultz, *SIAM J. Numer. Anal.*, **23** (1), 44 (1986).
15. M. O. Deville and E. H. Mund, *SIAM J. Sci. Stat. Comp.* **11** (2), 311 (1990).
16. H. Guillard and J-A. Desideri, *Comput. Methods Appl. Mech. Eng.* **80,** 305 (1990).
17. J. Strain, *Proc. Amer. Math. Soc.* **122** (3), 843 (1994).
18. H. A. Van der Vorst, *SIAM J. Sci. Stat. Comput.* **13** (2), 631 (1992).

Received July 31, 1996; revised December 2, 1996

COSTAS D. DIMITROPOULOS
ANTONY N. BERIS*

Department of Chemical Engineering
University of Delaware
Newark, Delaware 19716
E-mail: beris@che.udel.edu

* Author to whom all correspondence should be addressed.