Journal of Computational Physics **149,** 377–388 (1999)
Article ID jcph.1998.6159, available online at http://www.idealibrary.com on IDEAL®

A Multilevel Iterative Field Solver for Implicit, Kinetic, Plasma Simulation

D. A. Knoll, G. Lapenta,* and J. U. Brackbill

Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received June 8, 1998; revised November 19, 1998

The equation for electrostatic potential which arises from the implicit moment method in plasma simulation is a nonsymmetric elliptic equation. We present results using a simple multigrid method as a preconditioner to General Minimum RESidual (GMRES) to iteratively solve this nonsymmetric elliptic equation in two dimensions. It is demonstrated that a simple multigrid method produces an effective preconditioner. It is also demonstrated that under some conditions the required number of linear iterations is independent of grid dimension. Results are presented for both uniform and nonuniform grid problems. © 1999 Academic Press

Key Words: implicit plasma simulation; nonsymmetric systems; preconditioned Krylov methods; multigrid methods.

1. INTRODUCTION

Much effort has gone into the development of implicit methods for electromagnetic kinetic plasma simulation. Two such methods are the implicit moment method [1] and the direct implicit method [2]. In the implicit moment method a Taylor series expansion in time is used to approximate new time charge density and current as a function of old time charge density and current. The new time approximations are then used in Maxwell's equations to solve implicitly for the electric and magnetic fields. Substituting new time approximations of charge density into Poisson's equation results in a nonsymmetric elliptic equation for electrostatic potential. The asymmetry is a result of a convection-like term arising from $\mathbf{E} \times \mathbf{B}$ drifts of the charged particles. The asymmetry is bounded, and typically not as large as the asymmetries in convection diffusion problems. However, it precludes symmetric system solvers and decreases robustness and efficiency. The maximum allowable time step in this method has often been found to be governed by the iterative elliptic field solver, and not the implicit moment method itself [3]. In Ref. [3] it was shown, in one dimension, that

[*] Permanent address: Dipartimento di Energetica, Politecnico di Torino, Turin, Italy.

by using a direct method to solve the field equations, significantly larger time steps could be achieved. Along with frequently being the controlling factor in time step size selection, the iterative solution of the nonsymmetric elliptic equation usually dominates the CPU time of a simulation. Research into robust and efficient implicit field solvers has also been an important part of the direct implicit plasma simulation method [4]. There is currently significant motivation to perform large time step, fine grid 3-D, implicit plasma simulation of magnetic reconnection in Earth's magnetotail [5, 6].

Nonsymmetric systems of equations occur in many problems. In general circulation models, eddy transport processes are described by a tensor diffusivity that can be decomposed into symmetric and antisymmetric parts [7]. Pert [8] discusses issues arising in the numerical solution of the Braginskii plasma transport equations and introduces extremal principles, which are satisfied by the differential equations and should be satisfied by numerical approximations to these equations. Finally, implicit formulations of convection–diffusion result in nonsymmetric systems [9, 10]. Methods modeled on the conjugate gradient method have been developed for nonsymmetric systems [11–14]. A comment that is often repeated is that when the nonsymmetric part is relatively large, convergence may be slow [11, 15]. Improved convergence for nonsymmetric systems can be obtained by effective preconditioning [16, 10, 17, 18].

We have initiated research into the development of an efficient and robust iterative elliptic solver which will allow large time step implicit plasma simulations, on fine grids, in two and three dimensions. Our algorithm uses the multigrid method [19–21] as a preconditioner to a Krylov subspace based iterative method applicable to nonsymmetric systems, GMRES (General Minimal RESidual) [12]. It is known that ILU-based preconditioners can be effective for nonsymmetric problems, but they do not scale well to fine grids. It is also known that developing an effective multigrid solver for a nonsymmetric problem can be challenging. One is often required to use matrix-dependent prolongation operators, sophisticated smoothers, and expensive multigrid cycles, such as the W- and the F-cycle. In Ref. [17] sophisticated multigrid methods are studied as both solvers and preconditioners for nonsymmetric problems. For the problems considered, multigrid was found to be more robust and efficient as a preconditioner than it was as a stand-alone solver. Here, multigrid is considered only as a preconditioner, and the multigrid preconditioner will have very simple attributes in terms of prolongation operators, smoother, and cycle.

For this study we will work with the CELEST2D code, an unpublished modification of CELEST1D [3]. CELEST2D is electrostatic but contains a spatially varying (time independent) magnetic field. Indeed, it is the electrostatic potential equation, from the implicit moment method, which provides the greatest challenge to iterative solvers. The elliptic equation for electrostatic potential, ϕ, in the implicit moment method of CELEST2D, is

$$\nabla \cdot [\epsilon(\mathbf{r}) \cdot \nabla\phi] = \rho(\mathbf{r}). \tag{1}$$

Here, $\mathbf{r}$ is the position vector, $\epsilon(\mathbf{r})$ is a spatially varying dielectric tensor, and $\rho(\mathbf{r})$ is the space charge. The functional form of $\epsilon(\mathbf{r})$ in CELEST2D is

$$\epsilon = 1 + \sum_s \alpha_s \{1 - 1 \times \mathbf{\Omega}_s + \mathbf{\Omega}_s \mathbf{\Omega}_s\}, \tag{2}$$

where s is a species label (e.g., electrons and ions) and α_s and $\mathbf{\Omega}_s$ are given by

$$\mathbf{\Omega}_s = \frac{q_s \delta t}{2 m_s c} \mathbf{B}, \tag{3}$$

$$\alpha_s = \omega_{ps}^s \frac{\delta t^2}{2} \bigg/ (1 + \mathbf{\Omega}_s \cdot \mathbf{\Omega}_s). \tag{4}$$

q_s is the species charge, m_s is the species mass, $\mathbf{B}$ is the magnetic field, and c is the speed of light. The plasma frequencies, ω_{ps}, and cyclotron frequency, $\mathbf{\Omega}_s$, are comparable or larger in size than δt^{-1} in typical simulations with CELEST2D. Consider the asymmetry of the dielectric tensor $\boldsymbol{\epsilon}$. The symmetric and antisymmetric parts of $\boldsymbol{\epsilon}$ are given by

$$\boldsymbol{\epsilon}_S = \tfrac{1}{2}[\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T],$$
$$\boldsymbol{\epsilon}_A = \tfrac{1}{2}[\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^T].$$

One can verify by direct substitution that $\boldsymbol{\epsilon}^T(\mathbf{B}) = \boldsymbol{\epsilon}(-\mathbf{B})$, and thus that

$$\nabla \cdot \boldsymbol{\epsilon}_A \cdot \mathbf{E} = \nabla \cdot \sum_s \alpha_s (\mathbf{E} \times \mathbf{\Omega}_2). \tag{5}$$

As noted by Plumb [7], this is a convection term,

$$\nabla \cdot \sum_s (\mathbf{E} \times \mathbf{\Omega}_s) = +(\nabla \times \alpha_s \mathbf{\Omega}_s) \cdot \nabla \phi. \tag{6}$$

The convection "velocity," $\nabla \times \alpha_s \cdot \mathbf{\Omega}_s$, is solenoidal.

Equation (1) is discretized using finite volumes in a general coordinate system, which produces a 9-point numerical stencil in two dimensions and a 27-point stencil in three dimensions [22]. $\phi(\mathbf{r})$ and $\rho(\mathbf{r})$ are located at cell centers and $\boldsymbol{\epsilon}(\mathbf{r})$ is located at cell vertices. It should be noted that the resulting numerical stencil for a 2-D uniform grid, with $\boldsymbol{\epsilon}(\mathbf{r}) = 1$, is not the standard 5-point star, but rather a 5-point cross.

Again, the major difficulty arises from the fact that the off-diagonal components of $\boldsymbol{\epsilon}(\mathbf{r})$, which are proportional to the simulation time step δt, cause the resulting matrix equation for ϕ,

$$\mathbf{A}\phi = \rho, \tag{7}$$

to be nonsymmetric. Not only is $\mathbf{A}$ nonsymmetric, but in some cases diagonal dominance may be lost on nonuniform grids.

2. MULTILEVEL PRECONDITIONED GMRES

The base solver for Eq. (1) in CELEST2D is a preconditioned Krylov method. The Krylov method is GMRES [12] since the matrix equation is nonsymmetric. The base preconditioner is a Jacobi iteration, otherwise referred to as diagonal scaling. Preconditioned Krylov methods are modern, robust, general elliptic solvers, and they can be implemented in a matrix-free fashion (linear or nonlinear). However, they do not scale well with grid refinement. By this we mean that as the number of grid points increases, the number of iterations needed to achieve the same level of convergence also increases.

Probably the best known Krylov method is the conjugate gradient (CG) method made popular in the computational physics community by Kreshaw [23]. It is worthwhile to compare CG, which can be applied only to symmetric matrices, to GMRES, which can be applied to nonsymmetric matrices, in terms of work and storage as a function of iteration. In CG, work scales linearly, and required storage is constant, as the number of iterations increases because CG enjoys a short vector recurrence relationship which allows one to construct an orthogonal set of search directions without storing all of the search directions. In GMRES, work scales quadratically, and required storage scales linearly, as a function of iteration count because GMRES must store all of the search directions in order to maintain an orthogonal set. An often employed "fix" is to store only k Krylov vectors, GMRES(k). If linear convergence is not achieved after k iterations a new, temporary, linear solution is constructed from the existing k vectors and GMRES is restarted, with this temporary solution as the initial guess. This restarting can significantly affect the linear convergence rate. When one uses GMRES one is greatly motivated to keep the required number of GMRES iterations low, which translates into effective preconditioning.

The right preconditioned form of Eq. (7) is

$$(\mathbf{AP}^{-1})(\mathbf{P}\phi) = \rho, \tag{8}$$

where $\mathbf{P}$ represents symbolically the preconditioning matrix and $\mathbf{P}^{-1}$ represents its inverse. In practice, this inverse is only approximately realized through some standard iterative process. Each GMRES iteration requires a preconditioned matrix–vector multiply,

$$\mathbf{w} = (\mathbf{AP}^{-1})\mathbf{v}, \tag{9}$$

where $\mathbf{v}$ is the known, n^{th}, search direction and $\mathbf{w}$ represents the first step in forming the $(n+1)^{\text{st}}$ search direction. The multiply requires two steps. Step 1 is preconditioning, where $\mathbf{Py} = \mathbf{v}$ is iteratively solved for $\mathbf{y}$. Step 2 is the matrix–vector multiply, $\mathbf{w} = \mathbf{Ay}$. It is precisely step 1 where we will employ a multigrid method.

In solving linear systems arising from discretized PDEs it is most common to see variants of incomplete factorizations (ILU–GMRES or ICCG) as preconditioners or matrix splitting methods (Jacobi, Gauss-Seidel). For any single grid preconditioner, whose memory scales linearly with problem size, the inequality holds,

$$\kappa\left(\mathbf{AP}_{SG}^{-1}\right)_{N_2} > \kappa\left(\mathbf{AP}_{SG}^{-1}\right)_{N_1}. \tag{10}$$

Here, $\kappa(\mathbf{M})$ is the condition number of matrix $\mathbf{M}$, N is the dimension of the linear system $N_2 > N_1$, and the subscript SG denotes single grid. The number of Krylov iterations required to achieve a selected convergence tolerance is linearly proportional to the condition number of the preconditioned system. Thus, as N increases, not only will the cost of a GMRES iteration increase but also the number of GMRES iterations, for a given convergence tolerance, will increase. An approach for overcoming this scaling is the multigrid method [19–21]. Multigrid methods promise optimal scaling with problem size, which means that the number of iterations required, for a given convergence tolerance, is independent of grid dimension. However, as stated previously, developing a robust multigrid method for nonsymmetric problems can be challenging, and very time consuming, often requiring significant problem specific tuning. We will consider the performance of a simple multigrid method as a preconditioner, knowing that

$$\kappa\left(\mathbf{AP}_{MG}^{-1}\right)_{N_2} \approx \kappa\left(\mathbf{AP}_{MG}^{-1}\right)_{N_1}. \tag{11}$$

The subscript *MG* here denotes a multigrid based preconditioner. This basic concept is not new [24]. The application of multigrid methods as preconditioners to CG on symmetric problems has been considered for problems arising in incompressible fluid flow [25], semiconductor simulation [26], and porous media flow [27]. For nonsymmetric problems, sophisticated multigrid has been considered a preconditioner in Ref. [17], and simple multigrid methods have been considered a preconditioner in Ref. [18].

For completeness, and clarity, we review briefly a standard 2-Grid V-Cycle [19–21]. We wish to obtain the iterative solution to $\mathbf{Py} = \mathbf{v}$ with $f \equiv$ fine, and $c \equiv$ coarse.

1. Relax $\mathbf{P}^f \mathbf{y}_f^0 = \mathbf{v}$ for $\mathbf{y}_f^1$ (Jacobi, Gauss-Seidel, ILU, ...).
2. Evaluate the linear residual $\mathbf{res}^f = \mathbf{v} - \mathbf{P}^f \mathbf{y}_f^1$, and restrict it to a coarse grid, $\mathbf{res}^c = \mathcal{R} * \mathbf{res}^f$.
3. Solve the coarse grid problem, $\mathbf{P}^c \delta \mathbf{y}_c = \mathbf{res}^c$, for the coarse grid correction $\delta \mathbf{y}_c$.
4. Prolong the coarse grid correction and update the fine grid solution vector, $\mathbf{y}_f^2 = \mathbf{y}_f^1 + \mathcal{P} * \delta \mathbf{y}_c$.
5. Relax $\mathbf{P}^f \mathbf{y}_f^2 = \mathbf{v}$ for $\mathbf{y}_f^3$ (Jacobi, Gauss-Seidel, ILU, ...).

Still to be defined are: (1) a definition of the prolongation and restriction operators $\mathcal{P}$ and $\mathcal{R}$, and (2) a definition of $\mathbf{P}^c$. A standard multigrid (greater than 2) V-Cycle is realized by recursively inserting steps 1 through 5 into step 3 until a "very coarse" grid has been reached, upon which a direct solve is performed. This will be the preconditioner for solving Eq. (8) with preconditioner GMRES.

3. MULTIGRID SPECIFICS

To complete the definition of the multigrid preconditioner one must define restriction, $\mathcal{R}$, and prolongation, $\mathcal{P}$, operators, and a method for constructing the required coarse grid matrices. In the multigrid method, restriction, $\mathcal{R}$, is used to transfer residuals (i.e., $\mathbf{v} - \mathbf{P}^f \mathbf{y}_f^1$) from a fine to a coarse grid, and prolongation, $\mathcal{P}$, is used to transfer solution vector corrections (i.e., $\delta \mathbf{y}_c$) from a coarse to a fine grid. In this study, piece-wise constant interpolation is used for both $\mathcal{R}$ and $\mathcal{P}$, with $\mathcal{R}$ possibly volume weighted. This is a very simple choice and most likely would not produce an optimal multigrid method as a stand-alone solver [21].

Next, one must define a method for calculating the coarse grid representation of $\mathbf{P}$. There are at least two distinct options. The first is to apply $\mathcal{R}$ to $\epsilon(\mathbf{r})$ to transfer the essential physics from the fine to the coarse grid and re-discretize Eq. (1). A second approach is to use the restriction and prolongation operators, along with the fine grid linear operator, to construct a coarse grid operator. This would come from

$$\mathbf{P}^c = \mathcal{R} * \mathbf{P}^f * \mathcal{P}. \tag{12}$$

Here $\mathbf{P}^c$ is the $(N/2) \times (N/2)$ coarse grid operator and $\mathbf{P}^f$ is the $N \times N$ fine grid operator. This approach is often referred to as a Galerkin coarse grid operator [21, 20]. In this study the second approach is employed due to its simplicity and "Black Box" nature. Clearly, only the fine grid matrix needs to be supplied to the preconditioner. Our approach is related to the Black Box multigrid concept [28] with the main exception that in Ref. [28] more sophisticated inter-grid transfer operators are used.

3.1. *Additive Correction Multigrid*

We present a detailed algebraic explanation of the preconditioner as it would be applied to a simple 2-D Poisson problem on a uniform gird with a 5-point star stencil. The origin of this simple multilevel idea can be traced back to Ref. [29], and its application as a multigrid solver can be found in Ref. [30]. Assume a simple 2-D 5-point operator for the problem $\nabla^2\mathbf{y}=\mathbf{v}$, which results in a penta-diagonal matrix, $\mathbf{P}^f$,

$$\mathbf{P}^f = (P1^f, P2^f, P3^f, P4^f, P5^f). \tag{13}$$

We desire the solution to $\mathbf{Py}=\mathbf{v}$ for each finite volume (i, j). That is,

$$P1^f_{i,j} * y_{i-1,j} + P2^f_{i,j} * y_{i,j-1} + P3^f_{i,j} * y_{i,j} + P4^f_{i,j} * y_{i,j+1} + P5^f_{i,j} * y_{i+1,j} = v_{i,j}. \tag{14}$$

Given an initial fine grid solution y^*, the linear residual at finite volume (i, j) is

$$res^f_{i,j} = v_{i,j} - P1^f_{i,j} * y^*_{i-1,j} - P2^f_{i,j} * y^*_{i,j-1} - P3^f_{i,j} * y^*_{i,j} - P4^f_{i,j} * y^*_{i,j+1} - P5^f_{i,j} * y^*_{i+1,j}. \tag{15}$$

For the two-level grid shown in Fig. 1, the four fine grid volumes, (i, j), $(i+1, j)$, $(i, j+1)$, and $(i+1, j+1)$, are combined to produce one coarse grid volume, (I, J). The coarse grid

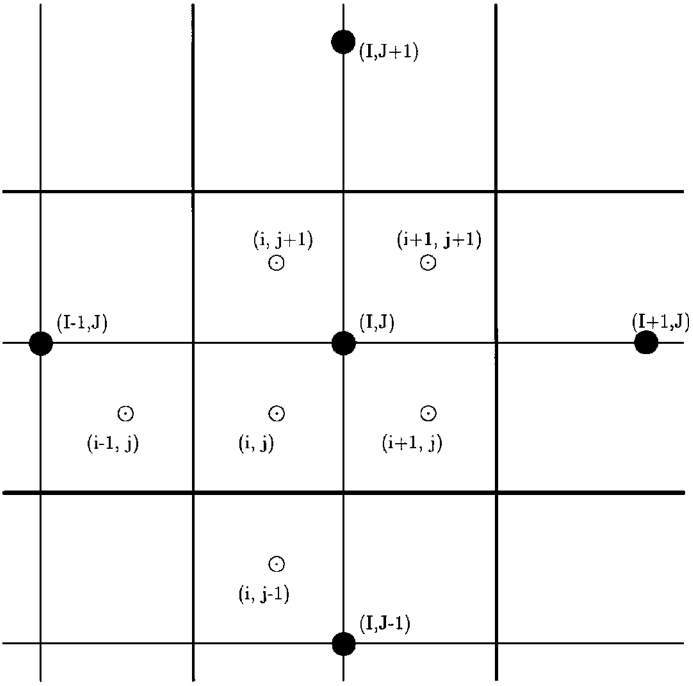

FIG. 1. Two-level finite volume grid.

correction is defined using piece-wise constant prolongation, $\mathcal{P}$,

$$\begin{aligned} y_{i,j} &= y^*_{i,j} + \delta y_{I,J}, \\ y_{i+1,j} &= y^*_{i+1,j} + \delta y_{I,J}, \\ y_{i,j+1} &= y^*_{i,j+1} + \delta y_{I,J}, \\ y_{i+1,j+1} &= y^*_{i+1,j+1} + \delta y_{I,J}. \end{aligned} \tag{16}$$

The next step is to add the four equations corresponding to volumes $(i, j), (i+1, j), (i, j+1), (i+1, j+1)$ that result from inserting Eq. (16) into Eq. (14). The contribution to this sum from volume (i, j) is

$$\begin{aligned} &P1^f_{i,j} * (y^*_{i-1,j} + \delta y_{I-1,J}) + P2^f_{i,j} * (y^*_{i,j-1} + \delta y_{I,J-1}) + P3^f_{i,j} * (y^*_{i,j} + \delta y_{I,J}) \\ &\quad + P4^f_{i,j} * (y^*_{i,j+1} + \delta y_{I,J}) + P5^f_{i,j} * (y^*_{i+1,j} + \delta y_{I,J}) = v_{i,j}. \end{aligned} \tag{17}$$

The coarse grid residual is defined by piece-wise constant restriction, $\mathcal{R}$,

$$res^c_{I,J} = res^f_{i,j} + res^f_{i+1,j} + res^f_{i,j+1} + res^f_{i+1,j+1}. \tag{18}$$

After the four equations from the fine grid are added and terms are collected the following coarse grid correction equation results for volume (I, J):

$$\begin{aligned} &P1^c_{I,J} * \delta y_{I-1,J} + P2^c_{I,J} * \delta y_{I,J-1} + P3^c_{I,J} * \delta y_{I,J} + P4^c_{I,J} * \delta y_{I,J+1} \\ &\quad + P5^c_{I,J} * \delta y_{I+1,J} = res^c_{I,J}. \end{aligned} \tag{19}$$

As a result it can be seen that the coarse grid matrix $(\mathbf{P}^c = \mathcal{R} * \mathbf{P}^f * \mathcal{P})$ is defined as

$$\begin{aligned} P1^c_{I,J} &= P1^f_{i,j} + P1^f_{i,j+1}, \\ P2^c_{I,J} &= P2^f_{i,j} + P2^f_{i+1,j}, \\ P4^c_{I,J} &= P4^f_{i,j+1} + P4^f_{i+1,j+1}, \\ P5^c_{I,J} &= P5^f_{i+1,j} + P5^f_{i+,j+1}, \\ P3^c_{I,J} &= \text{Sum of all 20 } P^f\text{'s from the 4 fine grid volumes} \\ &\qquad - P1^c_{I,J} - P2^c_{I,J} - P4^c_{I,J} - P5^c_{I,J}. \end{aligned} \tag{20}$$

Thus a simple coarse grid matrix that is straightforward to form has resulted from simple choices for $\mathcal{R}$ and $\mathcal{P}$. The boundary conditions in $\mathbf{P}^f$ are automatically included in $\mathbf{P}^c$. This multigrid preconditioner is easy to add to an existing preconditioned GMRES solver. It is well known that for a second order equation one should use either bi-linear prolongation or restriction to obtain an optimal multigrid algorithm [21]. However, results suggest that the simple algorithm above is an effective preconditioner. Recall that, for our problem, in 2-D, $\mathbf{P}^f$ is actually a matrix with nine diagonals.

3.2. *Smoother*

Finally, a smoother for the multigrid algorithm must be chosen. We will use a simple damped Jacobi iteration where the damping coefficient may change from cell to cell. The preconditioner will be one V-cycle [19] with an equal number of pre- and post-smoothing steps (sweeps), a $V(\nu, \nu)$ cycle. Traditionally in multigrid a direct solve is used on the coarsest grid, but that is not done here. While Jacobi may not be the most effective smoother, its use results in potentially significant storage savings, especially in 3-D with a 27-point stencil, in the following manner. Multipass Jacobi for $\mathbf{A}x = b$ with $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ can be written as

$$x^{n+1} = \mathbf{D}^{-1}[b - (\mathbf{L} + \mathbf{U})x^n]. \tag{21}$$

This requires the formation of **D**, and **L**, and **U**, i.e., **A**. In plasma simulation, particles compete for memory since the more available memory one has the more particles one can simulate. Thus one is motivated to consider all options for memory savings in the elliptic field solver. A reduced storage approach exploits the relation, $(\mathbf{L} + \mathbf{U})x^n = \mathbf{A}x^n - \mathbf{D}x^n$, and iterates the equation

$$x^{n+1} = \mathbf{D}^{-1}[b - (\mathbf{A}x^n - \mathbf{D}x^n)], \tag{22}$$

where $\mathbf{A}x^n$ formed in matrix-free fashion [31]. Thus one need only form **D** (main diagonal, i.e., one non-zero diagonal) and be able to form $\mathbf{A}x^n$ and **D** on coarse grids. This is our primary motivation for using the Jacobi smoother. The damped Jacobi iteration is realized by

$$x^{n+1} = \tilde{\mathbf{D}}^{-1}[b - (\mathbf{A}x^n - \mathbf{D}x^n)], \tag{23}$$

where $\tilde{\mathbf{D}}$ is a locally modified version of **D**.

4. PERFORMANCE RESULTS

For the first performance result the single grid Jacobi based preconditioner is compared to the multigrid Jacobi based preconditioner. The physical parameters for this problem are $m_i/m_e = 100$, $\omega_{p,e} = 1$, $\Omega_e/\omega_{p,e} = 0.5$, $\omega_{p,e} \cdot \delta t = 2.0$. These results are for a uniform grid. The number of sweeps is 3 ($\nu = 3$) and the damping coefficient is a uniform 2/3, for both the single grid preconditioner and the multigrid smoother. Five GMRES vectors are stored and restart is employed with a linear convergence tolerance of 1.0×10^{-6}. The boundary conditions on ϕ are Dirichlet on three sides and Neumann of the fourth side. Table I contains performance data as a function of grid refinement for 32 × 32, 64 × 64, 128 × 128, and 256 × 256 grids. The data are averaged over three time steps, with the same time step on each cycle and on each grid. All runs were made on a CRAY Y-MP.

TABLE I

Algorithm Performance for a Uniform Grid as a Function of Grid Refinement: Average Number of GMRES Iterations per Time Step, and Ratio of CPU Time

Grid	Single grid (SG) preconditioner	Multigrid (MG) preconditioner	CPU ratio SG/MG
32 × 32	48	8.7	1.8
64 × 64	116	9	2.8
128 × 128	398	11	5.65
256 × 256	691	9	10.5

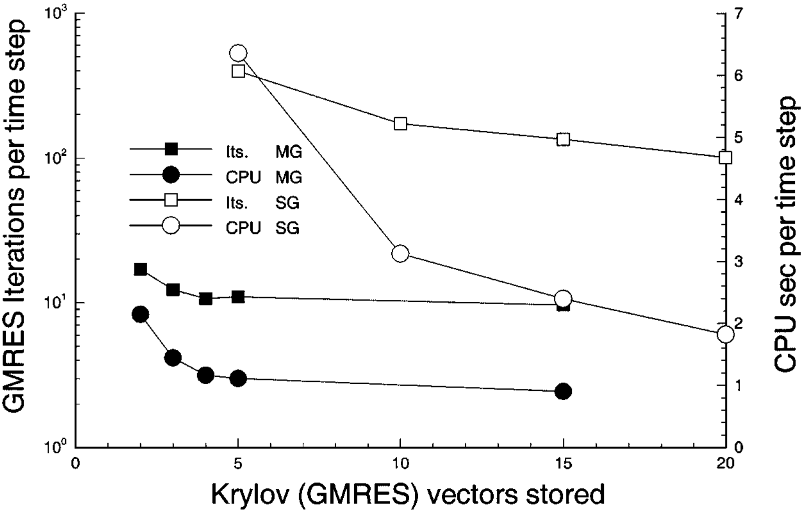

FIG. 2. Performance as a function of GMRES vectors stored.

It is quite apparent that the multigrid preconditioner has not only significantly reduced the required number of GMRES iterations; it has also decreased the growth of required iterations as a function of grid refinement. In fact, the number of required GMRES iterations appears to be independent of the grid dimension. On the 256 × 256 grid a factor of 10 speedup has been achieved.

As a second study we vary the Krylov subspace dimension, GMRES(k), with the results plotted in Fig. 2. This is the same problem as that above on the 128 × 128 uniform grid. Restart is employed with a linear convergence tolerance of 1.0×10^{-6}. The same time step is used on all grids, and the data are averaged over three time steps.

It should be noted that the storage of the matrix on the fine grid is equivalent to nine GMRES vectors, and that the additional multigrid storage is roughly equivalent to four GMRES vectors. Given this we can see that the multigrid preconditioner can be viewed as providing equivalent CPU performance for about half of the storage. If this performance carries over to the reduced storage Jacobi smoother there could be a factor of 10 savings in storage for the same CPU performance. With the multigrid based preconditioner, where the average number of GMRES iterations per time step is 10, there is no benefit to having a Krylov space larger than 5.

Next, the performance of the single grid Jacobi based preconditioner is compared to the multigrid Jacobi based preconditioner, for a nonuniform grid. These results have no volume weighting in the restriction operator. The number of sweeps is 1 ($\nu = 1$) and the damping coefficient is locally determined to ensure diagonal dominance, for both the single grid preconditioner and the multigrid smoother. Five GMRES vectors are stored and restart is employed with a linear convergence tolerance of 1.0×10^{-6}. Again the boundary conditions on ϕ are Dirichlet on three sides and Neumann on the fourth side. Table II contains performance data as a function of grid refinement for 32 × 32, 64 × 64, 128 × 128, and 256 × 256 grids. The data were averaged over three time steps, with the same time step on each cycle and on each grid.

It is again apparent, in the results of Table II, that the multigrid preconditioner has not only significantly reduced the required number of GMRES iterations; it has also decreased

TABLE II

Algorithm Performance for a Nonuniform Grid as a Function of Grid Refinement: Average Number of GMRES Iterations per Time Step, and Ratio of CPU Time

Grid	Single grid (SG) preconditioner	Multigrid (MG) preconditioner	CPU ratio SG/MG
32 × 32	451	24	8.2
64 × 64	1672	39.5	11.5
128 × 128	6217	64	14.5
256 × 256	7322	39	19.8

the growth of required iterations as a function of grid refinement. The nonuniform grid is a much more difficult problem, especially for our simple Jacobi smoother. However, inside a simple multigrid preconditioner we still obtain good performance and excellent performance relative to no multigrid. Note that the CPU gains for this problem are substantial, even on the smaller grids. Our experience suggests that 40 GMRES iterations is a respectable number on a 256 × 256 nonuniform grid, considering that the Krylov subspace dimension is only 5.

As a measure of diagonal dominance we have computed the average, over all rows in the matrix, of the main diagonal divided by the sum of the absolute values of all off-diagonal entries in that row. For the 32 × 32 uniform grid this number was approximately 2.0, while for the 32 × 32 nonuniform grid this number was approximately 1.0. The nonuniform grid problem has specific rows in which this diagnostic is less than 1.0. This suggests using a more robust smoother such as ILU. However, for our specific problem we are strongly motivated by the potential memory savings in the reduced storage Jacobi smoother, and thus we do not consider ILU.

The results given in Table II seem to indicate that the 128 × 128 grid was particularly difficult. We have redone this calculation with a Krylov subspace dimension of 10 and 20 with the multigrid preconditioner. With GMRES(10) the average GMRES iterations dropped to 39, and the GMRES(20) the average GMRES iterations dropped to 33. Thus, for more difficult nonuniform grid problems there is a possible benefit to retaining a larger Krylov subspace dimension. However, if storage is an issue, the performance of GMRES(5) acceptable.

Finally, we mention that the possibility of volume weighting in our restriction operator $\mathcal{R}$ was considered on the nonuniform grid problem. It did not seem to provide any consistent improvement on this problem.

5. CONCLUSIONS

We have developed and implemented a multigrid based preconditioner for GMRES and have applied this to the solution of a nonsymmetric elliptic problem that arises in implicit plasma simulation. A simple multigrid method as a preconditioner to GMRES appears to make a robust and efficient method for the elliptic field solver required in implicit plasma simulation codes on the problems considered. The multigrid based preconditioner can render the simple Jacobi based smoother effective, even on problems where diagonal dominance is marginal. On the uniform grid problem, the simple multigrid based preconditioner inside of GMRES(5) produced an effective iterative solver, with the number of required iterations independent of grid dimension. The CPU gains, as compared to the single grid Jacobi preconditioner, were significant with a factor of 10 on fine uniform grids and a factor of 20 on fine nonuniform grids.

It is not straightforward to compare the results of this study to similar studies involving convection–diffusion equations, which also yield nonsymmetric systems.These studies use more sophisticated multigrid preconditioners [17]. However, the same simple multigrid preconditioner presented here has been used inside of a Newton–GMRES method to solve the incompressible, convection dominated, Navier–Stokes equations in Ref. [18]. The simple multigrid preconditioner significantly outperformed an ILU based preconditioner and the multigrid preconditioned Newton–GMRES method was shown to be competitive with a more sophisticated nonlinear multigrid method. For the specific application to the elliptic field solve required in implicit plasma simulation the results of a simple multigrid based preconditioner are quite positive, and thus should be considered for other nonsymmetric systems. In the future we hope to combine the positive results of this study and the results in Ref. [18] and apply our multilevel solve to the streamlined Darwin field equations [4].

Finally, our true “measure of merit” for the simple multigrid preconditioner applied to our problem is not producing an algorithmically optimal solver, although the result for the uniform grid problem presented appears to approach this. The true “measure of merit” is the impact this method can have on implicit plasma simulation with a large time step, where there is a clear need for a robust and efficient elliptic equation solver.

ACKNOWLEDGMENTS

This work was supported under the auspices of the U.S. Department of Energy under DOE Contract W-7405-ENG-36 at Los Alamos National Laboratory. The supercomputer used in this investigation was provided by funding from the NASA Offices of Earth Science, Aeronautics, and Space Science.

REFERENCES

1. J. U. Brackbill and D. W. Forslund, An implicit method for electromagnetic plasma simulation in two dimensions, *J. Comput. Phys.* **46**, 271 (1982).
2. D. W. Hewett and A. Bruce Langdon, Electromagnetic direct implicit plasma simulation, *J. Comput. Phys.* **72**, 121–155 (1987).
3. H. X. Vu and J. U. Brackbill, CELEST 1D: An implicit, fully kinetic model for low-frequency, electromagnetic plasma simulation, *Comput. Phys. Comm.* **69**, 253 (1992).
4. D. W. Hewett, D. J. Larson, and S. Doss, Solution of simultaneous partial differential equations using dynamic ADI: Solution of the streamlined Darwin field equations, *J. Comput. Phys.* **101**, 11–24 (1992).
5. P. L. Pritchett, F. V. Coroniti, and V. K. Decyk, Three-dimensional stability of thin quasi-neutral current sheets, *J. Geophys. Res.* **101**, 27413–27429 (1996).
6. G. Lapenta and J. U. Brackbill, An exploration of the connection between the tearing and kink instabilities, submitted for publication.
7. R. A. Plumb and J. D. Mahlman, The zonally averaged transport characteristics of the BFDL general circulation transport model, *J. Atmos. Sci.* **44**, 298 (1987).
8. G. J. Pert, Physical constraints in numerical calculations of diffusion, *J. Comput. Phys.* **42**, 20 (1981).
9. C. I. Goldstein, Preconditioning convection dominated convection–diffusion equations, *Int. J. Numer. Methods Heat Fluid Flow* **5**, 99 (1995).
10. P. R. McHugh and D. A. Knoll, Fully implicit finite volume solutions of the incompressible Navier–Stokes and energy equations using inexact Newton’s method, *Int. J. Numer. Methods Fluids* **18**, 439–455 (1994).

11. S. C. Eisenstat, H. C. Elman, and M. H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.* **20**, 345 (1983).

12. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM J. Sci. Statist. Comput.* **7**, 856 (1986).

13. R. W. Freund, A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems, *SIAM J. Sci. Comput.* **14**, 470–482 (1993).

14. H. A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* **13**, 631–644 (1992).

15. G. Starke, Multilevel minimal residual methods for nonsymmetric elliptic problems, *SIAM J. Numer. Anal.* **3**, 351 (1996).

16. J. N. Shadid and R. S. Tuminaro, A comparison of preconditioned nonsymmetric Krylov methods on a large-scale MIMD machine, *SIAM J. Sci. Comput.* **15**, 440 (1994).

17. C. W. Oosterlee and T. Washio, An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems, *SIAM J. Sci. Comput.* **19**, 87 (1998).

18. D. A. Knoll and W. J. Rider, A multigrid preconditioned Newton–Krylov method, Los Alamos National Laboratory Report LA-UR-97-4013; to appear in *SIAM J. Sci. Comput.*

19. A. Brandt, Multi-level adaptive solutions to boundary value problems, *Math. Comp.* **31**, 333 (1977).

20. W. L. Briggs, *A Multigrid Tutorial* (SIAM, Philadelphia, 1987).

21. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, Chichester, 1992).

22. D. Sulsky and J. U. Brackbill, A numerical method for suspension flow, *J. Comput. Phys.* **96**, 339 (1991).

23. D. S. Kershaw, The incomplete cholesky–conjugate gradient method for the iterative solution of systems of linear equations, *J. Comput. Phys.* **26**, 43–65 (1978).

24. R. Kettler, Ph.D. thesis, University of Technology, Delft, 1987. [unpublished]

25. W. J. Rider, D. B. Kothe, S. J. Mosso, J. H. Cerruti, and J. I. Hochstein, *Accurate Solution Algorithms for Incompressible Multiphase Fluid Flows*, Technical Report AIAA 95-0699, AIAA, 1995. Presented at the 33rd Aerospace Sciences Meeting and Exhibit.

26. J. C. Meza and R. S. Tuminaro, A multigrid preconditioner for the semiconductor equations, *SIAM J. Sci. Comput.* **17**, 118–132 (1996).

27. S. F. Ashby and R. D. Falgout, A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations, *Nucl. Sci. Engrg.* **124**, 145–159 (1996).

28. J. E. Dendy, Black box multigrid, *J. Comput. Phys.* **48**, 366 (1982).

29. A. Settari and K. Aziz, A generalization of the additive correction methods for the iterative solution of matrix equations, *SIAM J. Numer. Anal.* **10**, 506–521 (1973).

30. B. R. Hutchinson and G. D. Raithby, A multigrid method based on the additive correction strategy, *Numer. Heat Transfer* **9**, 511–537 (1986).

31. W. J. Rider, Personal communication, 1997.