The Astrophysical Journal Supplement Series, 153:369–387, 2004 July

A COMPARISON OF ALGORITHMS FOR THE EFFICIENT SOLUTION OF THE LINEAR SYSTEMS ARISING FROM MULTIGROUP FLUX-LIMITED DIFFUSION PROBLEMS

F. Douglas Swesty
Department of Physics and Astronomy, SUNY at Stony Brook, NY 11794; dswesty@mail.astro.sunysb.edu

Dennis C. Smolarski, S. J.[1]
Department of Mathematics and Computer Science, Santa Clara University, Santa Clara, CA 95053; dsmolars@math.scu.edu

and

Paul E. Saylor
Department of Computer Science, University of Illinois, Urbana, IL 61801; saylor@cs.uiuc.edu
Received 2003 October 8; accepted 2004 March 2

ABSTRACT

Flux-limited diffusion has become a popular method for treating radiation transport in multidimensional astrophysical simulation codes with multi-group flux-limited diffusion (MGFLD) undergoing increasing use in a number of applications. The most computationally demanding aspect of this technique is the solution of the large linear systems that arise from the implicit finite-difference scheme that is used to solve the underlying integro-PDEs that describe MGFLD. The solution of these linear systems often dominates the computational cost of carrying out astrophysical simulations. Hence, efficient methods for solving these systems are highly desirable. In this paper we examine the numerical efficiency of a number of iterative Krylov subspace methods for the solution of the MGFLD linear systems arising from a series of challenging test problems. The problems we employ were designed to test the capabilities of the linear-system solvers under difficult conditions. The algorithms and preconditioners we examine in this study were selected on the basis that they are relatively easy to parallelize. We find that certain algorithm/preconditioner combinations consistently outperform others for a series of test problems. Additionally, we find that the method of preparing the linear system for solution by scaling the system has a dramatic effect on the convergence behavior of the iterative methods.

Subject headings: methods: numerical — radiative transfer

1. INTRODUCTION

Multi-group flux-limited diffusion (MGFLD) can be applied to radiation transport problems that arise in a wide variety of astrophysical phenomena. The radiation can consist of photons, neutrons, neutrinos, or, in some circumstances, charged particles. Despite the quantum behavior of each of these types of radiation being distinct, the macroscopic flow of the radiation is described by similar radiation transport equations. Typically, the physical problem one encounters is posed as an initial value problem and requires the time integration of these equations. One can attempt to solve these equations either by deterministic approaches or by Monte Carlo simulation. In the case of deterministic approaches, the characteristic structure of these equations usually necessitates the use of implicit finite-difference techniques. In turn, these implicit methods give rise to large linear systems. The linear systems are sparse and possess a structure that is in many cases a combination of diagonal blocks plus outlying bands. The modest storage requirements and overall efficiency of Krylov subspace iterative methods are suitable for these problems. In the multidimensional case, these systems are immense. Therefore, it is highly desirable to solve these systems on parallel architectures, which possess the required memory and CPU resources. Yet in order to accomplish this, one needs to develop effective parallel preconditioners. We report on our investigations into this area for a number of sparse approximate inverse (SPAI) preconditioners in combination with several popular Krylov subspace methods, including the generalized minimum residual method with fixed restart length n (GMRES(n), Saad & Schultz 1986), as well as the stabilized bi-conjugate gradient (BiCGSTAB, van der Vorst 1992) and the conjugate gradient squared (CGS, Sonneveld 1989) methods. Additionally, we have considered Chebyshev iteration (Elman et al. 1985) as another Krylov subspace method that has the added advantage of fewer inner products.

The objective of this paper is to present a comparison of the effectiveness of certain iterative sparse linear system methods for the solution of implicitly differenced MGFLD equations. The flux-limited approximation is often employed in computational astrophysics as a method for modeling the multidimensional flow of radiation. For example, the widely used ZEUS-2D code (Stone & Norman 1992a, 1992b; Stone et al. 1992) has recently included gray flux-limited diffusion as a technique for describing radiation flow (Turner & Stone 2001). In most simulations where radiation transport equations are handled by implicit finite-difference techniques the solution of the sparse linear systems dominates the computational cost of the simulation. Therefore, the efficient solution of the sparse linear systems arising from the discretization is imperative. To date, the only comparison of numerical methods for the solution of implicitly differenced flux-limited diffusion equations has been undertaken by Baldwin et al. (1999), who compared several algorithms for the iterative solution of gray flux-limited diffusion equations in a radiation-hydrodynamic context.

[1] On sabbatical at the Department of Computer Science, University of Illinois, Urbana, IL 61801.

However, the test problems in the Baldwin et al. paper exhibit behavior that is not necessarily relevant to astrophysical phenomena. Furthermore, this work did not consider a class of preconditioners that we explore. A brief preliminary comment about preconditioning is appropriate, a topic that will be addressed in more detail in § 6. The iterative convergence of sparse linear system methods for solving $\mathcal{A}x = b$ can often be improved by the transformation of the system by another matrix M^{-1} into a new system $M^{-1}\mathcal{A}x = M^{-1}b$ that transforms the coefficient matrix into one with a more favorable spectrum (Barrett et al. 1994). The matrix M^{-1} is chosen so that $M^{-1}\mathcal{A}$ is a good approximation of the identity matrix. A good preconditioner accelerates the convergence of an iterative method and, without any such preconditioning, many methods will fail to converge.

For these reasons we investigate the efficiency of several popular sparse linear system algorithms as applied to the MGFLD linear systems. We have focused on Krylov subspace algorithms because they are well suited for linear systems arising from discretized partial differential equations and because of the relative ease with which they can be implemented on massively parallel architectures. In addition we consider several possible sparse approximate inverse preconditioners that admit parallel implementations. The goal of these studies is twofold: first, to assess how the iteration number for the various preconditioner/iterative-method combinations scaled with the size of the system on realistic MGFLD problems; second, to compare the performance of the various preconditioner/iterative-method combinations on these same MGFLD problems.

In this paper we restrict ourselves to problems with one spatial and one spectral dimension. The reason for this is that the size of the linear systems allows us to gain some insight into the character of the preconditioner/iterative-method combinations without the strenuous computation required for problems in two and three spatial dimensions. Our work in the paper is focused on the comparison of iterative methods without the complicating issues of parallel implementation. Parallel scalability of the algorithms introduces another dimension to the problem that complicates any comparative analysis of iterative methods. In future work, we will consider scalability of the methods on linear systems corresponding to problems with higher spatial dimensionality.

In § 2 we delineate the differencing scheme that gives rise to these problems. In § 3 we discuss the characteristics of the linear systems in several important physical limits. In § 4 we describe the model problems that we employ in these investigations. In § 5 we briefly discuss the Krylov subspace methods, and in § 6 we discuss the various preconditioners that we have employed along with their motivation. In § 7 we report on our results. And finally in § 8 we offer some comments regarding the parallelism of the preconditioners and iterative methods as well as some conclusions drawn from this work.

2. THE RADIATION TRANSPORT MODEL

A widely employed model for the flow of radiation is the *flux-limited diffusion* approximation in which the flux of radiation is related to the gradient of either the energy density or the number density in a manner that satisfies Fick's law (Kittel & Kroemer 1980). The *energy density*, $E(\varepsilon)$, is defined as the total energy of "particles" of radiation, with *energy* ε, per unit volume and is related to the *number density*, n, by $E(\varepsilon) = \varepsilon n(\varepsilon)$. If the spectral distribution of the radiation cannot be described by a simple thermodynamic distribution function, one must discretize the spectrum into a series of energy zones or "groups" and solve a flux-limited diffusion equation for each group. This model of radiation transport is known as *multi-group flux-limited diffusion* (*MGFLD*). Each group may be connected to every other group by physical processes such as nonconservative scattering. Other physical processes that may be present are conservative scattering processes, i.e., scattering events in which the energy of a "particle" of radiation remains unchanged, and emission/absorption processes which either create or destroy particles of radiation. In general, for this problem one can write the *monochromatic, radiation number/energy equation* for radiation "particles" of energy ε as (Mihalas & Mihalas 1984)

$$\frac{\partial E(\varepsilon)}{\partial t} + \nabla \cdot \boldsymbol{F}(\varepsilon) = S(\varepsilon) - E(\varepsilon) c \kappa^a(\varepsilon) + c \int E(\varepsilon') \kappa^s(\varepsilon', \varepsilon) d\varepsilon' - E(\varepsilon) c \int \kappa^s(\varepsilon, \varepsilon') d\varepsilon', \quad (1)$$

where $\boldsymbol{F}$ is the *flux* of the radiation, S is the radiation *emissivity*, κ^a is the *absorption opacity*, κ^s is the *nonconservative scattering opacity*, and c is the speed of light. It is understood that E, $\boldsymbol{F}$, S, κ^a, and κ^s are all functions of the radiation energy ε, and we will hereafter omit the ε argument for the sake of compact notation. The solution of equation (1) requires the specification of an additional relationship, commonly referred to as a *closure*, between the monochromatic energy density E and the monochromatic flux $\boldsymbol{F}$. The physical behavior of this relationship in the limit where the surroundings are either opaque to the radiation ("optically thick") or transparent to the radiation ("optically thin") is well understood. In one dimension in the optically thin limit, the monochromatic energy density E and the magnitude of the monochromatic flux F are related by

$$F = cE, \quad (2)$$

while in the optically thick limit they are related by

$$F = D(\varepsilon) \nabla E, \quad (3)$$

where $D(\varepsilon)$ is a *diffusion coefficient*, with an energy-dependent value, that depends on the problem-specific microphysical interactions of radiation with matter. In intermediate ("optically translucent") regimes between the optically thin and optically thick limits, the behavior is more complex. A number of closures have been developed (Alme & Wilson 1974; Bruenn et al. 1978; Minerbo 1978; Levermore 1984; Cernohorsky & Bludman 1994) that try to bridge the gap between the two limits. The form of these closures is

$$F = D(\varepsilon, \nabla\varepsilon) \nabla E, \quad (4)$$

where $D(\varepsilon, \nabla\varepsilon)$ is a function of both ε and $\nabla\varepsilon$ such that F achieves the correct value in both the optically thick and optically thin limits. One of the more popular prescriptions for this closure is that of Levermore & Pomraning (1981, hereafter LP), which we employ in this work. The LP closure is given by

$$D(\varepsilon, \nabla\varepsilon) = -\frac{c}{\kappa^t \omega} \left[\frac{\coth(R)}{R} - \frac{1}{R^2} \right], \quad (5)$$

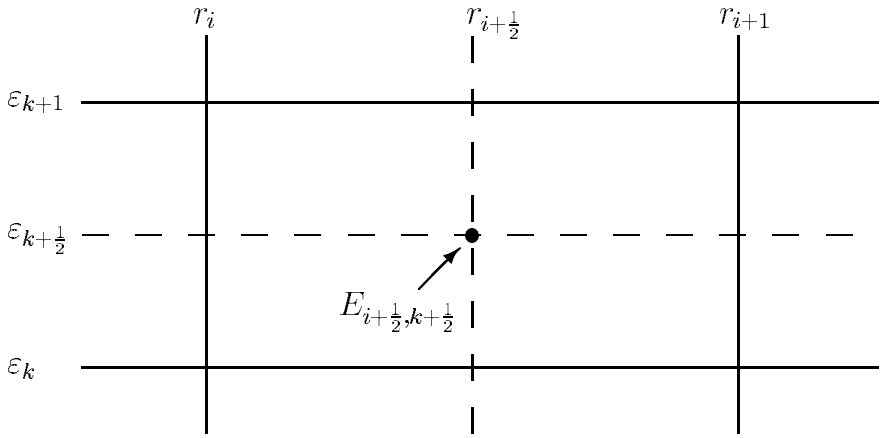

FIG. 1.—Diagram of the energy-space mesh illustrating the location at which the monochromatic energy density is defined.

where R is the *Knudsen number* given by

$$R = -\frac{\nabla E}{\kappa^t \omega E}. \tag{6}$$

In equations (5) and (6), ω is the *effective albedo* given by

$$\omega = \frac{S + \kappa^{\rm sc} E}{\kappa^t E}, \tag{7}$$

where $\kappa^{\rm sc}$ is the *total scattering opacity* including both conservative and nonconservative scattering contributions, and $\kappa^t = \kappa^a + \kappa^{\rm sc}$ is the total opacity including absorption (κ^a) and total scattering ($\kappa^{\rm sc}$) contributions. In the test problems considered in this paper, we will ignore absorption and emission (for reasons we will explain in § 4); thus, $S = 0$, $\kappa^t = \kappa^{\rm sc}$, and $\omega = 1$. As detailed in LP, this closure achieves the appropriate limits as $R \longrightarrow 0$ and $R \longrightarrow \infty$. The reader is referred to LP for further details. With this closure, equation (1) assumes the form of an integro-PDE in the variable E.

The implicit finite differencing scheme used to solve equation (1) is the spatially staggered mesh scheme that has been formulated by Myra et al. (1987). This scheme is illustrated in Figure 1. The same spatial discretization has also been adopted by Turner & Stone (2001) in the widely employed ZEUS-2D astrophysical radiation-hydrodynamics code. For the problems considered in this paper we will assume spherical symmetry and employ spherical polar coordinates. We discretize the spatial coordinate r and the spectral coordinate ε into N_r *radial zones* and N_g *energy groups*, respectively. We denote the edges of these discretized radial zones by r_i using integer subscripts and the zone centers by $r_{i+(1/2)}$ using half-integer subscripts. Similarly, group edges and group centers are denoted by ε_k and $\varepsilon_{k+(1/2)}$. Employing standard finite-differencing notation we denote the time-level of the quantities by superscripts, i.e., the number density at zone center $r_{i+(1/2)}$ and group center $\varepsilon_{k+(1/2)}$ is denoted at time t^n by $E^n_{i+(1/2),k+(1/2)}$. Employing this notation we can discretize equation (1) as

$$\frac{E^{n+1}_{i+(1/2),k+(1/2)} - E^n_{i+(1/2),k+(1/2)}}{\Delta t} =$$

$$-\frac{(r_{i+1})^2 F^{n+1}_{i+1,k+(1/2)} - (r_i)^2 F^{n+1}_{i,k+(1/2)}}{(\Delta V)_{i+(1/2)}}$$

$$+ c\sum_{j=1}^{j=N_g} E^{n+1}_{i+(1/2),j+(1/2)} \kappa^s_{i+(1/2),j+(1/2),k+(1/2)} \Delta\varepsilon_{j+(1/2)}$$

$$- cE^{n+1}_{i+(1/2),k+(1/2)} \sum_{j=1}^{j=N_g} \kappa^s_{i+(1/2),k+(1/2),j+(1/2)} \Delta\varepsilon_{j+(1/2)}, \tag{8}$$

where

$$(\Delta V)_{i+(1/2)} \equiv \frac{1}{3}\left[(r_{i+1})^3 - (r_i)^3\right], \tag{9}$$

$$\kappa^s_{i+(1/2),j+(1/2),k+(1/2)} \equiv \kappa^s\left(r_{i+(1/2)}, \varepsilon_{j+(1/2)}, \varepsilon_{k+(1/2)}\right), \tag{10}$$

and where we have made use of the notation $\Delta\varepsilon_{k+(1/2)} = \varepsilon_{k+1} - \varepsilon_k$ and $\Delta t = t^{n+1} - t^n$. The flux is differenced as

$$F^{n+1}_{i,k+(1/2)} = D^n_{i,k+(1/2)} \frac{E^{n+1}_{i+(1/2),k+(1/2)} - E^{n+1}_{i-(1/2),k+(1/2)}}{r_{i+(1/2)} - r_{i-(1/2)}}. \tag{11}$$

Note that following Turner & Stone (2001) we hold the diffusion coefficient constant throughout the time step, i.e., $D^n_{i,k+(1/2)}$ is defined at time t^n.

By substituting equation (11) into equation (8) we can algebraically manipulate the resulting equation into the form

$$A_{i+(1/2)}\eta_{i-(1/2)} + B_{i+(1/2)}\eta_{i+(1/2)} + C_{i+(1/2)}\eta_{i+(3/2)} = R_{i+(1/2)}, \tag{12}$$

where

$$\eta^T_{i+(1/2)} \equiv \left(E^{n+1}_{i+(1/2),(3/2)}, E^{n+1}_{i+(1/2),(5/2)}, \ldots, E^{n+1}_{i+(1/2),N_g+(1/2)}\right), \tag{13}$$

and

$$R^T_{i+(1/2)} \equiv \left(E^n_{i+(1/2),(3/2)}, E^n_{i+(1/2),(5/2)}, \ldots, E^n_{i+(1/2),N_g+(1/2)}\right). \tag{14}$$

The superscript T in equations (13) and (14) indicates the transpose. $A_{i+(1/2)}$, $B_{i+(1/2)}$, and $C_{i+(1/2)}$ are $N_g \times N_g$ matrices given by

$$A_{i+(1/2)} = \begin{pmatrix} \alpha_{i+(1/2),(3/2)} & 0 & \cdots & 0 \\ 0 & \alpha_{i+(1/2),(5/2)} & \cdots & 0 \\ \vdots & & & \\ 0 & \cdots & 0 & \alpha_{i+(1/2),N_g+(1/2)} \end{pmatrix}, \tag{15}$$

$$B_{i+(1/2)} = \begin{pmatrix} \beta_{i+(1/2),(3/2),(3/2)} & \beta_{i+(1/2),(3/2),(5/2)} & \cdots & \beta_{i+(1/2),(3/2),N_g+(1/2)} \\ \beta_{i+(1/2),(5/2),(3/2)} & \beta_{i+(1/2),(5/2),(5/2)} & \cdots & \beta_{i+(1/2),(5/2),N_g+(1/2)} \\ \vdots & & & \\ \beta_{i+(1/2),N_g+(1/2),(3/2)} & \beta_{i+(1/2),N_g+(1/2),(5/2)} & \cdots & \beta_{i+(1/2),N_g+(1/2),N_g+(1/2)} \end{pmatrix}, \tag{16}$$

and

$$C_{i+(1/2)} = \begin{pmatrix} \gamma_{i+(1/2),(3/2)} & 0 & \cdots & 0 \\ 0 & \gamma_{i+(1/2),(5/2)} & \cdots & 0 \\ \vdots & & & \\ 0 & \cdots & 0 & \gamma_{i+(1/2),N_g+(1/2)} \end{pmatrix}, \tag{17}$$

where

$$\alpha_{i+(1/2),k+(1/2)} = \frac{(r_i)^2 D^n_{i,k+(1/2)} \Delta t}{(\Delta V)_{i+(1/2)} \left(r_{i+(1/2)} - r_{i-(1/2)}\right)}, \tag{18}$$

$$\gamma_{i+(1/2),k+(1/2)} = \frac{(r_{i+1})^2 D^n_{i+1,k+(1/2)} \Delta t}{(\Delta V)_{i+(1/2)} \left(r_{i+(3/2)} - r_{i+(1/2)}\right)}, \tag{19}$$

and

$$\begin{aligned} \beta_{i+(1/2),k+(1/2),j+(1/2)} &= -\kappa^s_{i+(1/2),j+(1/2),k+(1/2)} \Delta\varepsilon_{j+(1/2)} c \Delta t \\ &+ \delta_{k+(1/2),j+(1/2)} \left(1 - \alpha_{i+(1/2),k+(1/2)} - \gamma_{i+(1/2),k+(1/2)}\right) \\ &+ \delta_{k+(1/2),j+(1/2)} \left(\sum_{\ell=1}^{\ell=N_g} \kappa^s_{i+(1/2),k+(1/2),\ell+(1/2)} \Delta\varepsilon_{\ell+(1/2)} c \Delta t \right). \end{aligned} \tag{20}$$

In equation (20) we have made use of the Kronecker delta function:

$$\delta_{j+(1/2),k+(1/2)} = \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Equation (12) is a block tridiagonal linear system of $N_r N_g$ equations:

$$\begin{pmatrix} B_{(1/2)} & C_{(1/2)} & 0 & 0 & \cdots & 0 \\ A_{(3/2)} & B_{(3/2)} & C_{(3/2)} & 0 & \cdots & 0 \\ \vdots & & & & & \\ 0 & \cdots & & 0 & A_{N_r+(1/2)} & B_{N_r+(1/2)} \end{pmatrix} \times \begin{pmatrix} \eta_{(1/2)} \\ \eta_{(3/2)} \\ \vdots \\ \eta_{N_r+(1/2)} \end{pmatrix} = \begin{pmatrix} R_{(1/2)} \\ R_{(3/2)} \\ \vdots \\ R_{N_r+(1/2)} \end{pmatrix}. \tag{22}$$

We shall refer to the matrix in equation (22) as the *coefficient matrix*, and we will denote this matrix symbolically by $\mathcal{A}$. The sub- and super-diagonal blocks of the coefficient matrix are themselves diagonal. In the absence of a nonconservative scattering opacity κ^s, the system is symmetric. The sparsity pattern of the coefficient matrix $\mathcal{A}$ for the simple case of $N_r = 4$ and $N_g = 20$ is illustrated in Figure 2.

Many radiation transport problems are solved in conjunction with explicit hydrodynamic simulations where the time step Δt is set by the *Courant-Friedrichs-Lewy* (CFL) stability limit for the hydrodynamics. Alternatively, radiation transport simulations performed without hydrodynamic evolution are often performed at a fixed transport CFL number. In either case the time step is proportional to the zone size Δr. We adopt a fixed CFL criterion in this paper even though we are not performing hydrodynamic evolutions. This implies that the lower resolution simulations will have larger time steps. For this paper the time steps are computed by

$$\Delta t = \tau \left(\frac{256}{N_r} \right), \tag{23}$$

where $\tau = 4.36$ μs and the 256 is chosen to make the scaling factor unity for our smallest simulations. Such a time step is typical in astrophysical simulations of neutron stars.

3. CHARACTERISTICS OF THE LINEAR SYSTEMS

Several of the test problems that we consider in this paper describe important physical limits of the MGFLD equation. These limiting problems also yield linear systems with widely differing characteristics. The variety of the mathematical characteristics of these systems is an important aspect of our testing of the preconditioner/iterative methods. Some of the physical limits of the MGFLD equation yield linear systems that demand more from the preconditioners.

In all cases the linear systems arising from one-dimensional MGFLD are block tridiagonal. The "*A*" and "*C*" blocks as described by equations (15) and (17) will always be diagonal although the magnitude of the diagonal elements of these blocks may vary. Nevertheless, we identify three important limiting cases for the MGFLD problem. The first is the *optically thick limit when dominated by conservative scattering.* The second is the *optically thick limit dominated by nonconservative scattering.* The third is the *optically thin limit*. Each of these limits produces linear systems with different characteristics. We now describe these cases in turn.

3.1. *Optically Thick Limit, Conservative Scattering*

The simplest limit is the optically thick limit, sometimes known as the diffusive limit, where the opacity is high and conservative scattering is dominant while nonconservative scattering is negligible. When nonconservative scattering processes are negligible, each integral term in equation (1) is effectively zero. In the limit where the conservative scattering opacity becomes large, the diffusion coefficient given by equation (5) is small and the flux of radiation given by

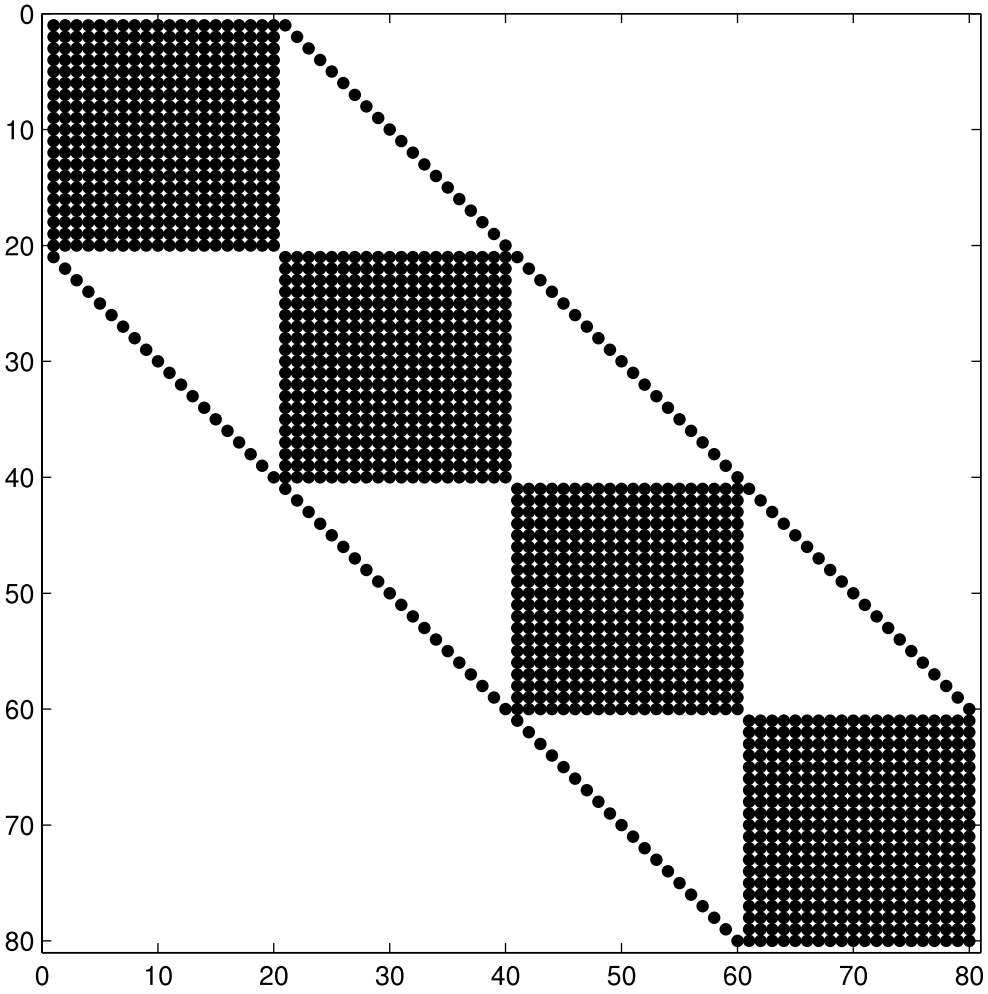

FIG. 2.—Sparsity pattern of the coefficient matrix $\mathcal{A}$.

equation (4) is essentially zero. Under these circumstances we find that

$$A_{i+(1/2)} \approx 0, \tag{24}$$

$$B_{i+(1/2)} \approx \begin{pmatrix} \beta_{i+(1/2),(3/2),(3/2)} & 0 & \cdots & 0 \\ 0 & \beta_{i+(1/2),(5/2),(5/2)} & \cdots & 0 \\ \vdots & & & \\ 0 & \cdots & 0 & \beta_{i+(1/2),N_g+(1/2),N_g+(1/2)} \end{pmatrix}, \tag{25}$$

and

$$C_{i+(1/2)} \approx 0. \tag{26}$$

The coefficient matrix is approximately diagonal, and the solution of this system is trivial. The time evolution of the radiation density is almost entirely due to the presence of any absorption opacity in $\beta_{i+(1/2),j+(1/2),k+(1/2)}$ or the presence of source terms on the right-hand side of the system.

3.2. *Optically Thick Limit, Nonconservative Scattering*

The second important limiting case that can arise is in the optically thick limit, where nonconservative scattering is significant. In this case the integral terms of equation (1) remain present and the system takes the form

$$A_{i+(1/2)} \approx 0, \tag{27}$$

$$B_{i+(1/2)} = \begin{pmatrix} \beta_{i+(1/2),(3/2),(3/2)} & \beta_{i+(1/2),(3/2),(5/2)} & \cdots & \beta_{i+(1/2),(3/2),N_g+(1/2)} \\ \beta_{i+(1/2),(5/2),(3/2)} & \beta_{i+(1/2),(5/2),(5/2)} & \cdots & \beta_{i+(1/2),(5/2),N_g+(1/2)} \\ \vdots & & & \\ \beta_{i+(1/2),N_g+(1/2),(3/2)} & \beta_{i+(1/2),N_g+(1/2),(5/2)} & \cdots & \beta_{i+(1/2),N_g+(1/2),N_g+(1/2)} \end{pmatrix}, \tag{28}$$

and

$$C_{i+(1/2)} \approx 0. \tag{29}$$

In this circumstance all of the elements of the center block $\beta_{i+(1/2),j+(1/2),k+(1/2)}$ may be large. However, the inverse of this system may be obtained by computing the inverse of $B_{i+(1/2)}$. If the number of energy groups is large the cost of this computation may be prohibitive.

3.3. *Optically Thin Limit*

The third limit occurs in the optically thin case, where both conservative and nonconservative scattering are negligible. In this limit one can show algebraically that the diffusion coefficient becomes

$$D^n_{i,k+(1/2)} \approx \frac{cE^n_{i,k+(1/2)}}{(\nabla E)^n_{i,k+(1/2)}}. \tag{30}$$

The effect of this limit on the linear systems may most effectively be seen in the plane-parallel case of the spherical coordinate system, where r is large and where

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\ldots\right)\approx\frac{\partial^2}{\partial r^2}(\ldots). \tag{31}$$

In this limit the α and γ coefficients both become proportional to the CFL number σ corresponding to the speed of light c, where

$$\sigma\equiv\frac{c\Delta t}{\Delta r}. \tag{32}$$

For most applications $\sigma\gg 1$. It is this fact that necessitates the use of implicit numerical methods for equation (1). We can then write the α and γ coefficients as

$$\alpha_{i+(1/2),k+(1/2)}\approx\frac{D^n_{i,k+(1/2)}\Delta t}{(\Delta r)^2}=\sigma\left[\frac{E^n_{i,k+(1/2)}}{(\nabla E)^n_{i,k+(1/2)}\Delta r}\right] \tag{33}$$

and

$$\gamma_{i+(1/2),k+(1/2)}\approx\frac{D^n_{i+1,k+(1/2)}\Delta t}{(\Delta r)^2}=\sigma\left[\frac{E^n_{i+1,k+(1/2)}}{(\nabla E)^n_{i+1,k+(1/2)}\Delta r}\right]. \tag{34}$$

Equations (33) and (34) reveal that the diagonal elements of each block become dominant as the time step Δt and the CFL number σ increase. Additionally, we can see that another stressful case arises when the radiation distribution $E^n_{i,k+(1/2)}$ becomes uniform, i.e., $E^n_{i,k+(1/2)}\approx E$. In this situation the gradient of the radiation density ∇E goes to zero and the α and γ coefficients blow up. This is usually dealt with numerically by modifying the gradient to be

$$(\nabla E)^n_{i,k+(1/2)}=\frac{E^n_{i+(1/2),k+(1/2)}-E^n_{i-(1/2),k+(1/2)}}{\Delta r}+\delta, \tag{35}$$

where δ is a small positive number chosen to prevent division by zero in equations (33) and (34). In practice, this additional term only has a significant effect when adjacent spatial zones have virtually identical energy densities in the same energy group. The inclusion of this term is a practical necessity to prevent a division by zero in such a circumstance where the radiation field is "flat." In this optically thin limit with a flat radiation field the linear system takes on the form

$$A_{i+(1/2)}\approx\begin{pmatrix}\frac{\sigma E}{\delta} & 0 & \ldots & 0\\ 0 & \frac{\sigma E}{\delta} & \ldots & 0\\ \vdots & & & \\ 0 & \ldots & 0 & \frac{\sigma E}{\delta}\end{pmatrix}, \tag{36}$$

$$B_{i+(1/2)}\approx\begin{pmatrix}1-\frac{2\sigma E}{\delta} & 0 & \ldots & 0\\ 0 & 1-\frac{2\sigma E}{\delta} & \ldots & 0\\ \vdots & & & \\ 0 & \ldots & 0 & 1-\frac{2\sigma E}{\delta}\end{pmatrix}, \tag{37}$$

and

$$C_{i+(1/2)}\approx\begin{pmatrix}\frac{\sigma E}{\delta} & 0 & \ldots & 0\\ 0 & \frac{\sigma E}{\delta} & \ldots & 0\\ \vdots & & & \\ 0 & \ldots & 0 & \frac{\sigma E}{\delta}\end{pmatrix}. \tag{38}$$

Since the scattering opacity, κ^s, is negligible, there is no coupling between the energy groups via the integral terms of equation (1). Thus, the linear system can be separated into N_g uncoupled tridiagonal linear systems of the form

$$\begin{pmatrix}1-\frac{2\sigma E}{\delta} & \frac{\sigma E}{\delta} & 0 & 0 & \ldots & 0\\ \frac{\sigma E}{\delta} & 1-\frac{2\sigma E}{\delta} & \frac{\sigma E}{\delta} & 0 & \ldots & 0\\ \vdots & & & & & \\ 0 & \ldots & 0 & \frac{\sigma E}{\delta} & 1-\frac{2\sigma E}{\delta} & \frac{\sigma E}{\delta}\\ 0 & \ldots & 0 & 0 & \frac{\sigma E}{\delta} & 1-\frac{2\sigma E}{\delta}\end{pmatrix}\times\begin{pmatrix}E^{n+1}_{(1/2),k+(1/2)}\\ E^{n+1}_{(3/2),k+(1/2)}\\ \vdots\\ E^{n+1}_{N_r-(3/2),k+(1/2)}\\ E^{n+1}_{N_r-(1/2),k+(1/2)}\end{pmatrix}=\begin{pmatrix}E^{n}_{(1/2),k+(1/2)}\\ E^{n}_{(3/2),k+(1/2)}\\ \vdots\\ E^{n}_{N_r-(3/2),k+(1/2)}\\ E^{n}_{N_r-(1/2),k+(1/2)}\end{pmatrix}. \tag{39}$$

When $\sigma E/\delta$ is large, we can factor out the constant and the matrix in equation (39) becomes

$$\begin{pmatrix}+2 & -1 & 0 & 0 & \ldots & 0\\ -1 & +2 & -1 & 0 & \ldots & 0\\ \vdots & & & & & \\ 0 & \ldots & 0 & -1 & +2 & -1\\ 0 & \ldots & 0 & 0 & -1 & +2\end{pmatrix}. \tag{40}$$

This limiting form corresponds to the second-order finite-difference representation of the Laplacian in Cartesian coordinates. The Laplacian possesses a dense inverse and the condition number scales with the number of equations N_r. The condition number of a matrix cond(A) is defined to be $\| A \| \| A^{-1} \|$. If cond(A) is small, A is said to be well conditioned. If cond(A) is large, A is said to be ill conditioned and solving a system with A as the coefficient matrix may be problematic, even if the system is preconditioned (Trefethen & Bau 1997). In this asymptotic limit (optically thin and flat) the condition number of the MGFLD coefficient matrix scales with the number of equations N_r and the linear system corresponding to each group becomes ill conditioned as N_r becomes large. Furthermore, the overall linear system formed by combining the N_g systems of size $N_r\times N_r$ may be even more ill conditioned.

4. TEST PROBLEMS

In order to offer a robust test for preconditioner/iterative method combinations we have constructed a set of simplified test problems that models many of the behavioral features of a variety of physical radiation transport applications.

These problems are based on the physical problem of proto–neutron star cooling, which requires the modeling of multigroup neutrino transport within the core of a collapsed star and its surrounding environs. The problem must treat neutrino transport over the entire range of behavior from the optically thick to optically thin limits. Additionally, the problem involves neutrino-electron scattering, which serves to couple strongly the individual energy groups in the problem. The problem involves simulating the flow of neutrinos out of a spherically symmetric star. The spectrum is continuous, thus eliminating the need to deal with spectral lines. The density is sufficiently large in the center of the star that the neutrinos there are diffusive, while in the medium around the star the matter is either translucent or transparent to the neutrinos.

Rather than involve the complications of neutrino microphysics, we have simplified the essential behavioral features of this problem into a less complex problem with simple parameterized opacities. By varying the parameters in these opacities we obtain a suite of linear systems that are used to test preconditioner/iterative method combinations. We assume that matter is in a fixed configuration to avoid the complications of radiation-hydrodynamic effects. The domain of computation is a sphere with an outer radius of 10^7 cm. The "star" is given a fixed density profile described by

$$\rho(r) = \max[\,\rho_c \exp(-r/r_0), \rho_0], \tag{41}$$

where $\rho_c = 10^{14}$ g cm^{-3}, $r_0 = 5\times10^4$ cm, and $\rho_0 = 10^8$ g cm^{-3}. The central density of this object is typical of the core collapse of a massive star midway through its collapse. The spatial domain is discretized into a uniform number of radial zones of width $\Delta r = 10^7/N_r$ cm. The spectrum is discretized into a series of groups with geometrically increasing values for the group center: $\varepsilon_{k+(3/2)} = \zeta\varepsilon_{k+(1/2)}$ with $\varepsilon_{(3/2)} = 2.5$ MeV and $\zeta = 1.5$. The group edges are defined by the geometrical mean of adjacent group centers: $\varepsilon_k = \left(\varepsilon_{k+(1/2)}\varepsilon_{k-(1/2)}\right)^{1/2}$. Such a grouping scheme is typical of that used in neutrino transport models. For the models in this paper we employ $N_g = 20$ groups. Typically, in a supernova or proto–neutron star model (Myra et al. 1987; Swesty et al. 1994) one would employ a geometrical factor of $\zeta = 1.3$ for the discretization of the spectrum, but the larger factor used here provides a more stressful test problem by increasing the amount of scattering between groups. The initial values for the radiation number densities are set according to

$$E_{i+(1/2),k+(1/2)} = \frac{\rho_{i+(1/2)}}{m_b}\left(\frac{m_1}{m_{i+1}}\right)^2\left(\frac{\varepsilon_{(3/2)}}{\varepsilon_{k+(1/2)}}\right)^3, \tag{42}$$

where m_i is the total mass contained interior to r_i, and $m_b = 1.66056\times10^{-24}$ g is the baryon mass. This initial radiation density is monotonically decreasing outward and provides for a higher radiation density to be "trapped" in the diffusive interior and assures an outward flux of radiation.

The model for the microphysics has been considerably simplified. The conservative scattering opacity is modeled as

$$\kappa^c_{i+(1/2),k+(1/2)} = \kappa^c\left(\rho_{i+(1/2)}, \varepsilon_{k+(1/2)}\right) = S_c C_c \rho_{i+(1/2)}\left(\varepsilon_{k+(1/2)}\right)^2, \tag{43}$$

where $C_c = 1.02\times10^{-20}$ cm^2 MeV^2 g^{-1}. This conservative scattering opacity is based on the real neutrino-nucleon scattering opacity. The S_c factor is an arbitrary factor that we have inserted into the opacity to change the optical depth of the model. In this paper we will consider multiple values of S_c. The nonconservative scattering opacity is loosely based on the physical process of neutrino-electron scattering. It has the effect of changing the energy of the neutrinos causing them to scatter into another energy group. Compton scattering would play a similar role for photons. Our model for the nonconservative scattering opacity is

$$\kappa^s_{i+(1/2),j+(1/2),k+(1/2)} = \frac{S_s}{\varepsilon_{j+(1/2)}}\kappa^c_{i+(1/2),j+(1/2)} \times \exp\left(-\left|\varepsilon_{j+(1/2)} - \varepsilon_{k+(1/2)}\right|/\Delta\varepsilon_{j+(1/2)}\right), \tag{44}$$

where S_s is a parameter that we will typically take to be 10^{-2}. This model for the opacity is chosen to ensure that the scattering is predominantly into adjacent groups and so that the overall nonconservative opacity is a few percent of the conservative opacity. Additionally, the factor of $1/\varepsilon$ ensures that the nonconservative opacity increases linearly with energy. This simplified opacity model avoids the tremendous complication of the true neutrino-electron opacity (Bruenn 1985) while ensuring the same qualitative behavior. The choice of an exponential form ensures that the scattering amplitude falls off for more distant groups, a generic feature for many applications.

As mentioned before, we will assume that the absorption opacity $\kappa^a(\varepsilon)$ and the number emissivity $S(\varepsilon)$ are both zero in this paper, avoiding the complications of picking a realistic form for these functions while at the same time making for a more stressful test problem. A nonzero absorption opacity would simply increase the value of the diagonal elements of the corresponding linear systems moving the matrix toward diagonal dominance, a situation for which preconditioning is trivial. By ignoring the absorption opacity we move the system away from diagonal dominance toward a more difficult preconditioning situation. The emissivity only enters into the linear system in the right-hand side and thus has no effect on how well conditioned the system is.

For the remainder of this paper we consider a suite of six linear systems problems in which we choose the opacity parameters to emphasize transport in the optically translucent and optically transparent limits. The test problems were established with several objectives in mind. First, we wanted to understand how the numerical methods scaled with the number of equations. Therefore, we created a series of three problems employing a varying number of radial zones, i.e., $N_r = 256$, 512, and 1024. In this case we kept $S_c = 1$. A second objective is to understand how the methods varied as the optical depth of the problem varied. Thus, we consider two additional problems with $S_c = 10^2$ and $S_c = 10^{-2}$ while keeping the number of radial zones fixed at $N_r = 1024$. Finally, we considered one additional problem where nonconservative scattering is absent ($S_s = 0$). This problem resulted in a symmetric system where each of the tridiagonal blocks is itself diagonal. This inclusion of this test problem allowed us to examine how the methods perform on a system that is symmetric.

The complete set of problem parameters is delineated in Table 1.

TABLE 1
PROBLEM PARAMETERS

Problem	N_r	S_c	S_s
1	256	1	10^{-2}
2	512	1	10^{-2}
3	1024	1	10^{-2}
4	1024	10^{-2}	10^{-2}
5	1024	10^{2}	10^{-2}
6	1024	10^{-2}	0

5. ITERATIVE SOLVERS

As previously mentioned, we have considered four iterative methods in our study of parallelizable methods for solving the multi-group flux-limited diffusion equations. For the sake of brevity we will not include the details of each of these methods in this paper. Instead we refer the reader to Barrett et al. (1994), Greenbaum (1995), and Saad (1996) for a thorough discussion of these algorithms.

Three of the methods we consider (GMRES, BiCGSTAB, and CGS) are easily implemented in a parallel context where the parallelism is derived from a logically Cartesian spatial domain decomposition. In the case of the GMRES algorithm (Saad & Schultz 1986) we have only considered a short restart length of five because of the large storage requirement associated with longer restart values. Hereafter, we denote this particular configuration of the GMRES algorithm by GMRES(5). This short restart length yields a minimal storage requirement comparable to the other methods.

The fourth method that we consider is based on Chebyshev iteration (Elman et al. 1985) and is somewhat more complicated. This method has been implemented as CHEBYCODE (Ashby 1985). The algorithm consists of two major parts: determining iteration parameters based on an estimation of the convex hull surrounding the eigenvalues in the complex plane and the Chebyshev iteration itself. The former (parameter construction) step is based on the adaptive procedure of Manteuffel (1978) and consists of a series of rather complex procedures for finding the convex hull surrounding the eigenvalues. The latter (iterative) step is rather simple and is described in Barrett et al. (1994). The parameter construction step requires some estimate of the eigenvalues of the preconditioned system, which are obtained by applying the GMRES algorithm for the first four iterations (Elman et al. 1985). Using the residuals from the first four iterations one can then estimate eigenvalues by the PM4 variant (J. Castor 1988, private communication) of the power method (Golub & van Loan 1989), which requires a linear least-squares fit accomplished via a QR factorization or Householder transformation. The adaptive procedure of Manteuffel can then be applied to estimate the convex hull and the iterative parameters. The major restriction to this method is that all the eigenvalues of the (preconditioned) system must lie entirely within either the right or the left half-plane. In practice we have always found such a situation in the radiation transport problems that we have examined. The only inner products needed in the parameter construction step are contained in the four GMRES iterations and the QR (or Householder) steps.

The parameter construction step can be reapplied periodically to reestimate the iterative parameters. The initial parameter estimation takes place after the first four GMRES iterations. For our test problems we have reestimated parameters every 10 iterations after the initial estimation but a posteriori examination of the residuals revealed no significant gain in the reduction of the residual from frequent reestimation. In practice we believe parameter reestimation is unnecessary for the problems in this paper. Such a strategy would minimize the number of inner products needed. We also wish to note that the iteration step requires only one matrix-vector multiply (MATVEC) per iteration. If an inner product were required it would be for a possible stopping criterion only.

Two of our algorithms, CGS (Sonneveld 1989) and BiCGSTAB (van der Vorst 1992), are variants of the BiCG algorithm, which is based on the Lanczos algorithm (Saad 1996, p. 210; Kelley 1995, p. 11). However, we have not considered the original BiCG algorithm because of the need for the transpose of the coefficient matrix. The transpose version of a MATVEC poses an additional communication requirement in a parallel setting, and in practice we have found BiCG to be less effective than the other algorithms that we are considering. The BiCGSTAB, GMRES, and CGS algorithms are all well described by Barrett et al. (1994), and we have implemented them exactly as described. We refer the reader to Barrett et al. (1994) for further details. We note that the BiCGSTAB and CGS algorithms require two MATVECs per iteration, which dominate the computational cost of the algorithms. For GMRES(5) we choose to count each pass of the middle loop of the algorithm as one iteration, with a single MATVEC per iteration.

The other factor to weigh when comparing these algorithms is the number of inner products required per iteration. For problems employing fine-grained parallelism the communication cost of these inner products can form a bottleneck to achieving good scalability. The CGS algorithm requires two inner products per iteration in order to estimate the iterative parameters, while BiCGSTAB requires four inner products per iteration. The GMRES(n) algorithm requires $n(n+3)/2$ inner products for every pass through the outermost loop (n iterations), thus yielding on average $(n+3)/2$ inner products per iteration. By choosing a restart length of five, we find on average that we have four inner products per iteration. An additional inner product is needed in each of these algorithms if one wishes to base the stopping criterion on the norm of residual. For the problems in this paper we halt the iteration when the norm of the residual divided by the norm of the right-hand side becomes less than 5.12×10^{-5}. This stopping criterion was determined by experiment to yield adequate accuracy with this particular discretization scheme applied to a variety of neutrino transport problems.

6. PRECONDITIONERS

To speed up the convergence of an iterative method, one may use a *preconditioning matrix* M^{-1} to transform the original linear system, $\mathcal{A}x = b$, into the system $M^{-1}\mathcal{A}x = M^{-1}b$, where M^{-1} is chosen so that $M^{-1}\mathcal{A}$ is a good approximation of the identity matrix (Ashby et al. 1989, 1992). The preconditioning matrix, M^{-1}, is often referred to simply as "the preconditioner." More generally, we may have two preconditioning matrices, M_L^{-1} and M_R^{-1}, that yield the preconditioned matrix $M_L^{-1}\mathcal{A}M_R^{-1}$. There are many possible preconditioning strategies, and a complete review is beyond the scope of this work. We refer the reader to Benzi (2002) for a comprehensive review of the subject.

For two- and three-dimensional problems the large size of the linear systems resulting from the implicit discretization of the MGFLD equations makes the use of parallel computing

architectures a practical necessity. In this paper we consider only one-dimensional problems for simplicity. However, we focus on preconditioning strategies that are easy to implement on massively parallel architectures. This focus on easily parallelizable preconditioners precludes consideration of some commonly used preconditioners, one obvious example being incomplete LU factorization (Greenbaum 1995). We choose to avoid the use of incomplete factorizations because the forward and backward solutions they require form a scaling bottleneck on parallel computers. There has been recent work on improving the scalability of incomplete factorizations, and we refer the reader to the discussion of this work by Benzi (2002). In this paper we chose to investigate an alternative preconditioning method (SPAI preconditioning) that has a high degree of inherent parallelism.

In addition to the SPAI preconditioners, we have investigated three other preconditioners that were based on the linear systems' behavior as they approach asymptotic physical limits. Preconditioner 1, which relies on the Thomas algorithm (Thomas 1949), derives from the optically thin limit where the coefficient matrix becomes diagonal in each of the three blocks. Under these circumstances the Thomas algorithm can be applied to each group individually to obtain the direct solution of the system. Even when the system does not obtain this asymptotic form, the Thomas algorithm can still be applied group-by-group as a preconditioner. In our case the tridiagonal system is formed by taking only the diagonal elements of equations (15), (16), and (17). The resulting system is employed as a preconditioner, which, as we will see, can be remarkably effective even in situations where it is not obvious that it should. The major drawback to this preconditioner lies in the recursive nature of the Thomas algorithm, which makes it difficult to parallelize effectively.

Another preconditioner (preconditioner 2) that we have considered is diagonal preconditioning, which renders an exact solution in the limit of pure diffusion in the absence of nonconservative scattering. The parallelization of this preconditioner is trivial. The third preconditioner (preconditioner 3) considered is block Jacobi preconditioning (Barrett et al. 1994), which yields an exact solution to the problem in the limit of diffusion with nonconservative scattering. This preconditioner is embarrassingly parallel when employed with a spatial domain decomposition. However, because this method requires the solution of the dense linear system involving the B block, the cost of this preconditioner may be prohibitive when the number of energy groups, and hence the block size, becomes large. Nevertheless, this preconditioner may be effective in certain circumstances.

An alternative, and more scalable, approach to preconditioning is to make use of SPAI preconditioners of a predetermined bandwidth (Benzi et al. 1996; Chow & Saad 1997, 1998; Gould & Scott 1998). This is the approach taken to compute preconditioners 4–7 described below. We presuppose that the coefficient matrix and its inverse would both be diagonally dominant and that creating an approximate inverse of appropriate bandwidth would produce an efficient preconditioner. The approximate inverse M^{-1} is obtained by minimizing (over all M^{-1} with predetermined entries)

$$\min \| M^{-1}\mathcal{A} - I\|_F^2 \equiv \sum_{j=1}^{n} \min \| \hat{m}_j^T \mathcal{A} - e_j^T\|_2^2, \quad (45)$$

where e_j is the unit n-vector in the jth direction, the Frobenius norm is defined by

$$\|G\|_F^2 \equiv \sum_{i=1}^{m}\sum_{j=1}^{n} |G_{ij}|^2, \quad (46)$$

where G is an arbitrary $m \times n$ matrix, and the L_2 norm is defined by

$$\|x\|_2^2 \equiv \sum_{i=1}^{n} |x_i|^2, \quad (47)$$

where x is an arbitrary n-vector.

In equation (45) the right-hand minimization is over all vectors $\hat{m}_j$ with entries in positions corresponding to those in the corresponding row of M^{-1}. An advantage of the SPAI preconditioners is the ability to parallelize the process of determining the components of the preconditioner. For a specific row of the approximate inverse the minimization that determines the elements of that row is independent of the minimization of all other rows. The minimization takes the form of a linear least-squares problem that can be solved by means of a QR decomposition. The size of the linear least-squares problem is determined by the number and position of the nonzero elements in each row of the approximate inverse. As a practical consideration, one desires to keep the number of elements in each row of the approximate inverse as small as possible in order that the linear least-squares problem remain small.

An important issue with the use of SPAI preconditioners is the choice of the positions of the elements in each row of the approximate inverse. A number of adaptive strategies have been suggested for choosing the sparsity pattern of the approximate inverse (Grote & Huckle 1997; Tang 1999; Benzi et al. 2001). These algorithms attempt to improve the efficiency of the preconditioners by adaptively augmenting the sparsity pattern of the preconditioner in order to minimize equation (45) further. However, this strategy can possibly incur a communication cost on a parallel architecture by requiring non–nearest-neighbor communications between processes as the bandwidth of the approximate inverse is increased. While this communication cost may be tolerable, the implementation is nontrivial (Barnard et al. 1999).

For this reason we have chosen the simplified strategy of considering a fixed sparsity pattern for the approximate inverse that requires only nearest-neighbor communication (assuming a one-dimensional process topology derived from a simple one-dimensional domain decomposition; Gropp et al. 1994). Our tests made use of several different fixed sparsity patterns motivated by relatively simple physical and mathematical considerations. Our choice of sparsity patterns was motivated by looking at absolute values of the elements of the coefficient matrix and its inverse for a small problem where the true inverse can be computed by brute force. This approach is best illustrated by Figures 3 and 4, which present false color images of the upper left corner of the coefficient matrix $\mathcal{A}$ (Fig. 3) and the upper left corner of the inverse $\mathcal{A}^{-1}$ (Fig. 4). The bright colored elements in the image of the inverse $\mathcal{A}^{-1}$ clearly reveal that large elements are present in bands with the same spacing as the bands that are present in $\mathcal{A}$. The location of these bands reflects the coupling between the energy density E at spatially adjacent points. Note that the size of the elements diminishes with the distance away from the diagonal. The behavior of the true inverse serves to motivate the placement of elements of a sparse approximate inverse.

Preconditioners 4–9 in our study are SPAI preconditioners that try six relatively simple sparsity patterns based on the aforementioned physical and mathematical motivations.

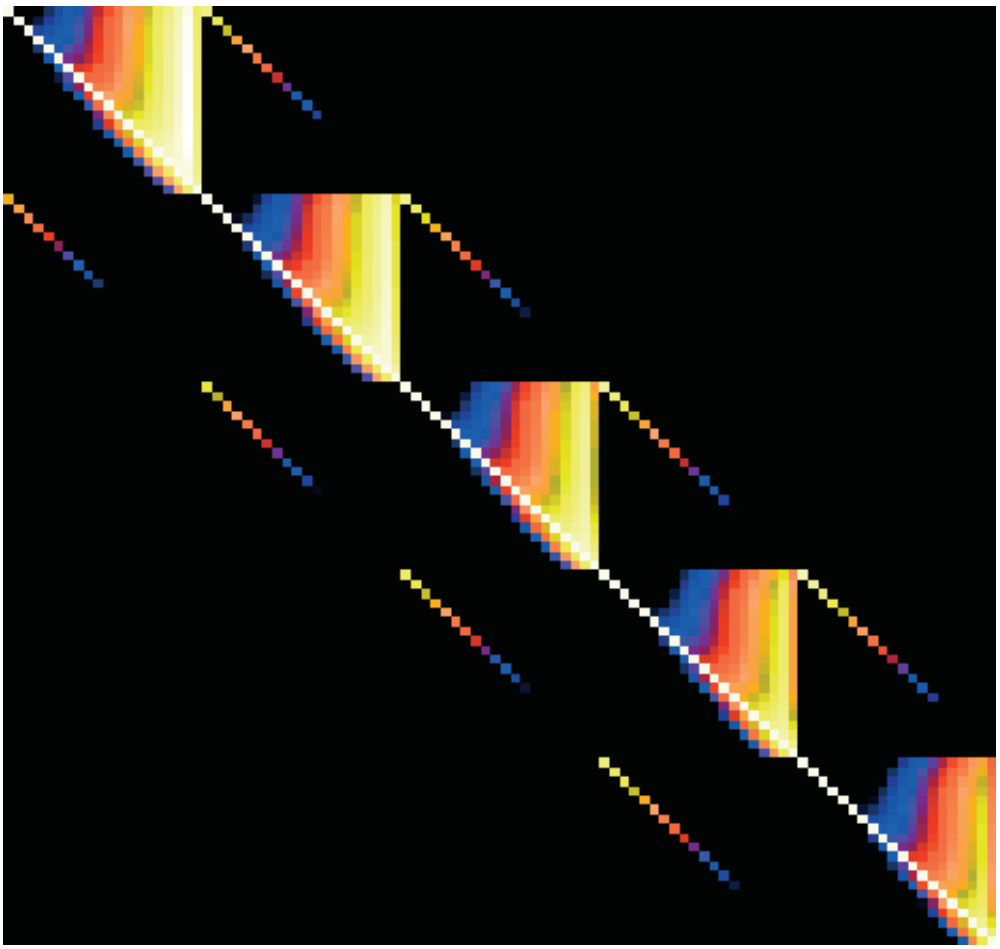

FIG. 3.—False color image of the absolute value of the upper left (100×100) corner of the coefficient matrix $\mathcal{A}$ for test problem 1. Brighter colors indicate larger elements.

The simplest of the SPAI preconditioners we consider is preconditioner 4 (SPAI 1), which uses a sparsity pattern consisting of a single element located on the diagonal of the approximate inverse. The value of this element is determined by a least-squares fit (Benzi et al. 1996; Gould & Scott 1995). Although the resulting matrix resembles preconditioner 2, the diagonal preconditioner, the computation is different in that the method used to determine the entries of the approximate inverse is a least-squares solution involving all the row elements of the coefficient matrix rather than merely the diagonal as in preconditioner 2.

Preconditioner 5 (SPAI 3) employs a tridiagonal sparsity structure, which could account for strong coupling between groups when nonconservative scattering is dominant. The values of the elements in this preconditioner are obtained by a linear least-squares fit as described in equation (45).

Preconditioner 6 (SPAI 3C) also uses the same three element approximate inverse. However, in this instance, the least-squares fit made use of only the nonzero elements in the center blocks of the coefficient matrix, ignoring other nonzero elements in the coefficient matrix. The motivation for this choice was to reduce the amount of computation required for the least-squares fit.

For preconditioner 7 (SPAI 5C) we employ a pentadiagonal sparsity pattern for the approximate inverse. The motivation for this choice was to study how the increase in bandwidth of the approximate inverse within the center block would improve the efficiency of the preconditioner. As with preconditioner 6, this matrix makes use of only the nonzero elements in the center blocks of the coefficient matrix, ignoring other nonzero elements in the coefficient matrix.

The sparsity patterns for preconditioners 8 and 9 are chosen for a different reason. A comparison of the SPAI sparsity pattern for preconditioners 8 and 9 (Figs. 5 and 6) with the true inverse dominant value pattern (Fig. 4) illustrates the motivation for the placement of the bands in preconditioners 8 and 9. Preconditioner 8 (SPAI 3T) is a three-banded sparse approximate inverse with two nonzero single-element bands spaced at a distance equal to the block size, i.e., $N_g = 20$, to either side of the diagonal (see Fig. 5 for an illustration of the sparsity pattern of this approximate inverse). This sparsity pattern allows coupling between nearest spatial zones by placing elements in the same bands in which they are found in the true inverse. The values of the elements are determined by least-squares fitting as described by equation (45). Similarly, preconditioner 9 (SPAI 5T) has five nonzero bands, with the four nondiagonal bands spaced at a distance of N_g and $2N_g$ from the diagonal (see Fig. 6 for an illustration). This pattern allows influence from the two nearest spatial zones located to either side of a specified zone. Because we are considering only sparsity patterns that allow the nearest spatial zones to influence a given zone, these two preconditioners are parallelizable under a logically Cartesian spatial domain decomposition with only nearest-neighbor communication.

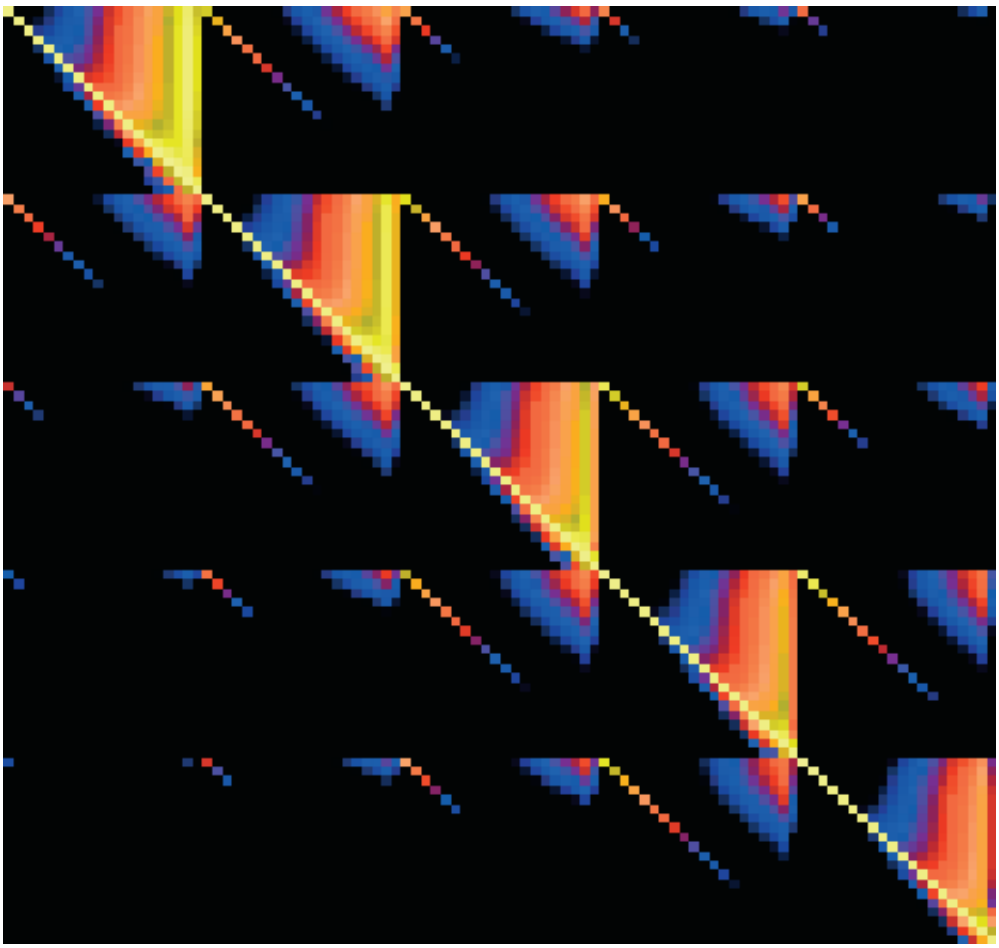

FIG. 4.—False color image of the absolute value of the upper left (100×100) corner of the inverse of the coefficient matrix $\mathcal{A}^{-1}$ for test problem 1. Brighter colors indicate larger elements.

7. RESULTS

We now examine the application of the iterative methods and preconditioners previously described. Before applying these methods to the linear systems we scale the systems so that the diagonal elements are unity. This is especially necessary in multigroup radiation transport applications as the resultant matrix elements can vary widely from row to row because of the energy dependence of the opacities. By scaling we eliminate the variation among rows, which helps to minimize problems associated with round-off error.

7.1. *Effects of Initial Scaling*

We begin with a comment on scaling the rows or columns of a matrix $\mathcal{A}$. Let $\mathcal{D}$ be the diagonal of $\mathcal{A}$. By *row scaling*, we mean multiplying on the left by $\mathcal{D}^{-1}$ (assuming that the inverse exists), whereas by *column scaling* we mean multiplying on the right by $\mathcal{D}^{-1}$. These scalings may be viewed as preconditioning operations. To explain, the general form of preconditioning is $M_L^{-1}\mathcal{A}M_R^{-1}$. Thus, row scaling means that $M_L = \mathcal{D},\ M_R = I$, whereas column scaling means that $M_R = \mathcal{D},\ M_L = I$.

Let $\mathcal{D}^{1/2}$ be a diagonal matrix, the nonzero elements of which are the square roots of the corresponding diagonal elements (assumed positive) of $\mathcal{D}$. If we take $M_R = \mathcal{D}^{1/2} = M_L$, then (since $M_R^T = M_R$) we have $M_L^{-1}\mathcal{A}M_R^{-1} = M_R^{-1}\mathcal{A}M_R^{-T}$, where M_R^{-T} means the transpose of the inverse of M_R. We call this *symmetric scaling*. Symmetric scaling preserves the symmetry of the symmetric part of the coefficient matrix. For the test problems in this paper we consider both row and symmetric scaling.

The effects of symmetric and row scaling are shown in Tables 2, 3, and 4, where we compare the results of the various preconditioner/algorithm combinations on problems 1, 2, and 3. In these tables we list both the number of iterations, N_i, and the number of MATVECs, N_m, required for convergence. The latter quantity is a machine-independent approximate measure of the total work required for convergence. We prefer this approximate measure to actual code timings, which can vary among platforms. A meaningful comparison of code timings would require the code for each individual preconditioner and iterative method to be tuned for a specific architecture. Furthermore, inevitable uncertainty as to whether the code has been optimally tuned leads to an uncertainty in the comparisons among preconditioners and algorithms. Counting MATVECs for each method avoids this uncertainty while still providing a rough metric for the total work. We have timed the setup phase for the preconditioner setup (and in the case of CHEBYCODE the parameter estimation) we have found that for most cases the setup time was on the order of the cost of one or two matvecs. However, our preconditioner code was written for maximum flexibility in investigating preconditioners as opposed to speed, while the MATVEC has been optimized for computational efficiency. Optimization of code for a specific

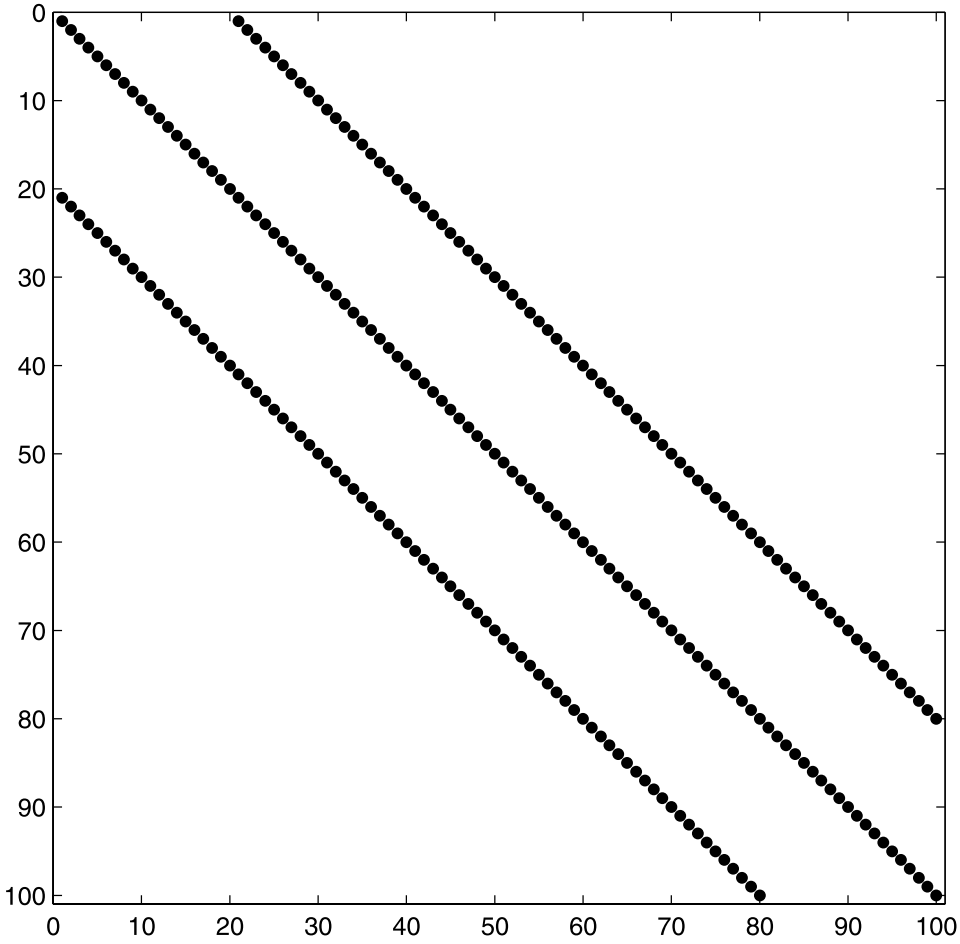

FIG. 5.—Sparsity pattern of preconditioner 8.

preconditioner is likely to yield a substantially lower time for the setup of that preconditioner. For this reason we view this rough estimate as an upper bound on the cost of the preconditioner setup.

An "NC" entry in the tables indicates nonconvergence of the preconditioner/algorithm combination. We consider failure to achieve convergence in 200 iterations as a nonconvergent result. We have found those cases in which convergence is not achieved in 200 iterations either evidence stagnation of the residual, or show such a slow decline in the residual, that the iterative-method/preconditioner combination is impractical for use.

The set of test problems 1–3 reveals some surprising results. First we note a clear difference between the symmetric and row scaling cases. In general, row scaling yields a superior rate of convergence in the linear system. Many of the combinations that fail to converge with symmetric scaling achieve convergence with row scaling. While we are at a loss for an explanation of this behavior, it is manifest in almost all of our test problems. The sole exception to this trend is the symmetric linear system in problem 6. In that case both symmetric and row scaling yield nearly identical results (within one iteration), which is expected since the two methods are the same in that special case.

If we consider only the row scaling case, which yields faster convergence of the iterative methods, we can see that the best results are obtained with preconditioners 1, 8, and 9. However, recall that preconditioner 1 uses the Thomas algorithm, which is recursive and difficult to parallelize. Comparable performance is obtained with preconditioners 8 and 9, which do not suffer from this drawback. In fact, because they require only nearest-neighbor communication the latter two preconditioners are relatively easy to parallelize under a spatial domain decomposition.

Problem 1 also reveals this fact about the performance of our four Krylov subspace methods: when employed with preconditioners 8 and 9 CHEBYCODE and BiCGSTAB both converge with comparable numbers of MATVECs. Note that CHEBYCODE requires only one MATVEC per iteration, while BiCGSTAB and CGS require two. Thus, the five iterations required by BiCGSTAB and CGS when used in conjunction with preconditioner 9 require roughly the same amount of work, as CHEBYCODE does with 11 iterations.

7.2. *Changing the Size of the System*

We now turn to the effects of changing the size of the linear system. This could be accomplished in two ways. The first approach increases the spatial resolution of the simulation by decreasing the zone size while keeping the spatial domain fixed in size. The second approach is to enlarge the spatial domain of the simulation while keeping the zone size fixed. We choose the former approach since many long-timescale astrophysical radiation transport simulations involve the simulation of an object of fixed size. Typically, the simulations are carried out with time steps determined by specifying the largest permissible CFL number, which is chosen so as to maintain sufficient

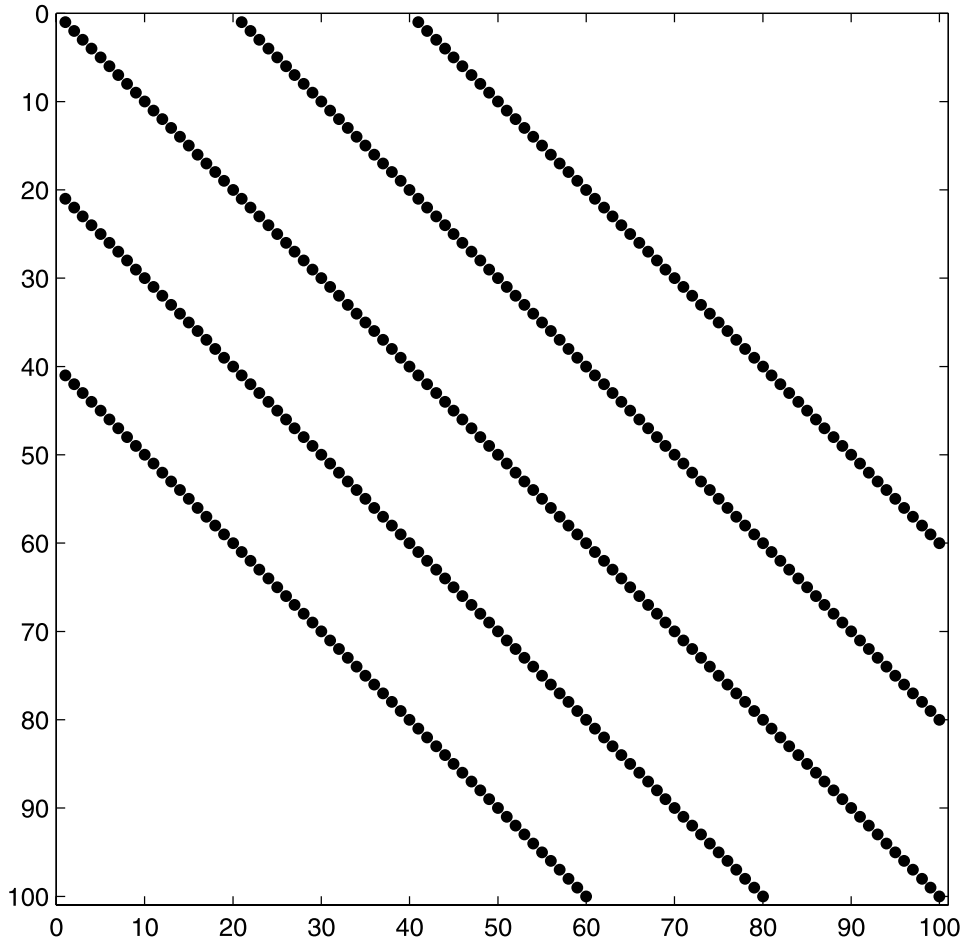

FIG. 6.—Sparsity pattern of preconditioner 9.

TABLE 2
PROBLEM 1: $N_r = 256$, $S_c = 1$, $S_s = 10^{-2}$

Preconditioner	CHEBYCODE N_i (N_m)	BiCGSTAB N_i (N_m)	CGS N_i (N_m)	GMRES(5) N_i (N_m)
Symmetric Scaling				
1 (Thomas)	19 (19)	7 (14)	6 (12)	15 (15)
2 (Diagonal)	41 (41)	10 (20)	11 (22)	20 (20)
3 (Block Jacobi)	29 (29)	8 (16)	13 (26)	15 (15)
4 (SPAI 1)	NC	53 (106)	174 (348)	195 (195)
5 (SPAI 3)	53 (53)	24 (48)	21 (42)	35 (35)
6 (SPAI 3C)	90 (90)	25 (50)	28 (56)	45 (45)
7 (SPAI 5C)	87 (87)	27 (54)	28 (56)	50 (50)
8 (SPAI 3T)	NC	56 (112)	103 (206)	120 (120)
9 (SPAI 5T)	NC	58 (116)	101 (202)	145 (145)
Row Scaling				
1 (Thomas)	15 (15)	5 (10)	5 (10)	10 (10)
2 (Diagonal)	27 (27)	9 (18)	10 (20)	20 (20)
3 (Block Jacobi)	25 (25)	8 (16)	13 (26)	15 (15)
4 (SPAI 1)	25 (25)	9 (18)	10 (20)	20 (20)
5 (SPAI 3)	21 (21)	8 (16)	10 (20)	15 (15)
6 (SPAI 3C)	28 (28)	9 (18)	9 (18)	15 (15)
7 (SPAI 5C)	26 (26)	9 (18)	9 (18)	15 (15)
8 (SPAI 3T)	11 (11)	5 (10)	7 (14)	15 (15)
9 (SPAI 5T)	11 (11)	5 (10)	5 (10)	10 (10)

NOTES.—The number of iterations, N_i, are shown followed by the number of MATVECs, N_m, in parentheses. An entry of "NC" indicates no convergence.

TABLE 3

PROBLEM 2: $N_r = 512$, $S_c = 1$, $S_s = 10^{-2}$

Preconditioner	CHEBYCODE N_i (N_m)	BiCGSTAB N_i (N_m)	CGS N_i (N_m)	GMRES(5) N_i (N_m)
Symmetric Scaling				
1 (Thomas)	18 (18)	6 (12)	6 (12)	15 (15)
2 (Diagonal)	40 (40)	11 (22)	10 (20)	20 (20)
3 (Block Jacobi)	31 (31)	8 (16)	13 (26)	15 (15)
4 (SPAI 1)	186 (186)	48 (96)	46 (92)	85 (85)
5 (SPAI 3)	40 (40)	19 (38)	18 (36)	30 (30)
6 (SPAI 3C)	80 (80)	20 (40)	21 (42)	30 (30)
7 (SPAI 5C)	87 (87)	30 (60)	23 (46)	35 (35)
8 (SPAI 3T)	NC	49 (98)	52 (104)	80 (80)
9 (SPAI 5T)	NC	50 (100)	51 (102)	85 (85)
Row Scaling				
1 (Thomas)	15 (15)	5 (10)	5 (10)	10 (10)
2 (Diagonal)	26 (26)	8 (16)	10 (20)	20 (20)
3 (Block Jacobi)	26 (26)	8 (16)	11 (22)	15 (15)
4 (SPAI 1)	23 (23)	9 (18)	10 (20)	20 (20)
5 (SPAI 3)	21 (21)	8 (16)	10 (20)	15 (15)
6 (SPAI 3C)	28 (28)	9 (18)	11 (22)	15 (15)
7 (SPAI 5C)	26 (26)	9 (18)	9 (18)	15 (15)
8 (SPAI 3T)	12 (12)	6 (12)	7 (14)	15 (15)
9 (SPAI 5T)	11 (11)	5 (10)	5 (10)	10 (10)

NOTES.—The number of iterations, N_i, are shown followed by the number of MATVECs, N_m, in parentheses. An entry of "NC" indicates no convergence.

TABLE 4

PROBLEM 3: $N_r = 1024$, $S_c = 1$, $S_s = 10^{-2}$

Preconditioner	CHEBYCODE N_i (N_m)	BiCGSTAB N_i (N_m)	CGS N_i (N_m)	GMRES(5) N_i (N_m)
Symmetric Scaling				
1 (Thomas)	17 (17)	6 (12)	6 (12)	15 (15)
2 (Diagonal)	38 (38)	11 (22)	10 (20)	20 (20)
3 (Block Jacobi)	29 (29)	8 (16)	13 (26)	15 (15)
4 (SPAI 1)	64 (64)	30 (60)	36 (72)	75 (75)
5 (SPAI 3)	27 (27)	15 (30)	13 (26)	20 (20)
6 (SPAI 3C)	64 (64)	21 (42)	20 (40)	25 (25)
7 (SPAI 5C)	59 (59)	20 (40)	18 (36)	25 (25)
8 (SPAI 3T)	NC	31 (62)	35 (70)	65 (65)
9 (SPAI 5T)	NC	33 (66)	34 (68)	45 (90)
Row Scaling				
1 (Thomas)	10 (10)	5 (10)	5 (10)	10 (10)
2 (Diagonal)	25 (25)	9 (18)	9 (18)	20 (20)
3 (Block Jacobi)	24 (24)	8 (16)	11 (22)	15 (15)
4 (SPAI 1)	23 (23)	9 (18)	13 (26)	20 (20)
5 (SPAI 3)	20 (20)	8 (16)	10 (20)	15 (15)
6 (SPAI 3C)	26 (26)	9 (18)	8 (16)	15 (15)
7 (SPAI 5C)	24 (24)	9 (18)	9 (18)	15 (15)
8 (SPAI 3T)	11 (11)	5 (10)	5 (10)	10 (10)
9 (SPAI 5T)	12 (12)	5 (10)	5 (10)	10 (10)

NOTES.—The number of iterations, N_i, are shown followed by the number of MATVECs, N_m, in parentheses. An entry of "NC" indicates no convergence.

accuracy. If the zone size is reduced, the time step must be reduced to maintain the specified CFL number. This decrease in the time step affects the linear system and, potentially, the convergence of the iterative method. As the time step is decreased, the off-diagonal elements decrease in size. Whether this decrease can offset the rise in condition number associated with an increased number of equations is the subject of test problems 2 and 3. We maintain the same opacity parameters in problem 2 and problem 3 while increasing the spatial resolution. In problem 2 the resolution is doubled to 512 spatial zones and in problem 3 it is quadrupled to 1024 spatial zones.

A comparison of the number of iterations among problems 1–3 for each of the numerical methods in the row scaling case shows no significant increase in the number of iterations as the size of the system is doubled and subsequently doubled again. We therefore conclude that these preconditioner/iterative-method combinations scale with the size of the system *when the CFL number is held constant*. This is a promising indication that these preconditioning strategies may scale to much larger numbers of equations arising out of three-dimensional MGFLD problems. However, there is clearly work that needs to be done to investigate further this possibility. We also note that the symmetric scaling case continues to provide convergence problems for all of the numerical methods.

Another major observation to draw from test problems 1, 2, and 3 is that preconditioners 8 and 9, which possess a sparsity pattern accounting for the influence of adjacent spatial zones, continue to perform well with all four Krylov subspace methods. In contrast, the sparse approximate inverses 4, 5, 6, and 7, which are achieved by fitting a sparsity pattern consisting of a few elements adjacent to the diagonal, yield inconsistent results, working well with some methods and less well with others. In problem 3, preconditioners 4–7 perform approximately as well as they do in problems 1 and 2. This can be understood in terms of the behavior of the elements of the B block given by equation (20). In an optically thick situation the elements of the entire B block can become large in magnitude. In contrast, the diagonal elements of the A and C blocks become smaller as the diffusion coefficient becomes smaller. An additional behavior that arises is that the time step becomes larger as the zone size increases with decreasing N_r. Consequently, the $c\Delta t$ terms in the equation for β become larger relative to the diagonal elements of the B block. Thus, the true inverse of the coefficient matrix becomes well approximated by a block-diagonal matrix consisting of the inverses of the B blocks. This is evidenced by the fact that both the diagonal preconditioner and the block-Jacobi preconditioners work even with symmetric scaling when the other preconditioners fail.

A third observation is that CHEBYCODE and BiCGSTAB, in conjunction with preconditioners 8 and 9, are competitive with respect to the number of MATVECs required to achieve convergence. In general, GMRES(5) and CGS seem to require only slightly more work to achieve convergence. Another aspect to this problem is the number of inner products that are required. However, in a parallel implementation CHEBYCODE may have an advantage over BiCGSTAB, and the other methods that we have considered, because of the relatively small number of global reduction operations required in a message passing context.

7.3. *Changing Optical Depth*

Another interesting aspect of the behavior of the methods is how the iterative performance varies as the optical depth of the problem changes. This is achieved by varying the parameter S_c in problems 3, 4, and 5. Problem 3 acts as the baseline problem. In problem 4 the opacity is decreased by 2 orders of magnitude relative to problem 3, while in problem 5 it is increased by 2 orders of magnitude relative to problem 3.

The results from this study are listed in Tables 4, 5, and 6. The symmetric scaling cases in problems 4 and 5 continue to manifest the same convergence difficulties, while row scaling continues to yield rapid convergence with most preconditioners. The SPAI preconditioners remain the best performers. Overall, preconditioners 8 and 9 continue to perform quite well. Preconditioners 5, 6, and 7 perform much better in the optically thick case (problem 5) than they do in the optically thin case (problem 4). This behavior is due to the increased importance of group-to-group scattering in the optically thick case, where the elements near the diagonal of the coefficient matrix obtain larger magnitudes. Thus, fitting an approximate inverse to these elements yields a better approximation of the true inverse. The importance of the group-to-group scattering in problem 5 also explains the improved performance of the block Jacobi preconditioner relative to problem 4. In general, the block Jacobi preconditioner is not competitive with the sparse approximate inverse preconditioners unless the problem is optically thick. For problems 3–5, CHEBYCODE, BiCGSTAB, and CGS offer comparable results for the same numbers of MATVECs. In a parallel context CHEBYCODE again should be the clear winner because of the small number of required global reduction operations. Furthermore, for preconditioners 8 and 9, the number of iterations to convergence does not seem to be a function of optical depth.

7.4. *The Symmetric Case*

The results from test problem 6 (Table 7) illustrate the performance of these methods on a symmetric problem in the optically thin limit. This problem can also be interpreted as a gray flux-limited diffusion problem because the resulting band structure of the matrix reflects only nearest-neighbor spatial coupling between zones. In this case the Thomas preconditioner becomes an excellent approximation to the true inverse of the coefficient matrix. This is manifested by the fact that most algorithms converge in a single iteration (five iterations for GMRES(5)) for preconditioner 1 with both row and symmetric scaling. Nevertheless, the recursive nature of preconditioner 1 makes it an undesirable choice on parallel platforms.

With the exception of preconditioner 1, the SPAI preconditioners 8 (SPAI 3T) and 9 (SPAI 5T) again offer the best performance. This is illustrated in Figure 7, where we compare the convergence of preconditioners 8 and 9 with diagonal preconditioning for the BiCGSTAB method with row scaling. In Figure 7 we have employed a tighter convergence tolerance than we would normally employ to illustrate that the SPAI preconditioners outperform diagonal preconditioning even as the residual is reduced over many orders of magnitude. While we have depicted the performance for the BiCGSTAB method, examination of Table 7 reveals the fact that each of the four iterative methods, when used in combination with the best preconditioners, require approximately the same work to obtain convergence. The results from approximate inverse preconditioners SPAI3T and SPAI5T indicate that SPAI preconditioners, which account for the spatial coupling between adjacent zones, are potentially useful for gray flux-limited diffusion problems. We also wish to point out that this test problem does not manifest the widely disparate behavior between symmetric scaling

TABLE 5

PROBLEM 4: $N_r = 1024$, $S_c = 10^{-2}$, $S_s = 10^{-2}$

Preconditioner	CHEBYCODE N_i (N_m)	BiCGSTAB N_i (N_m)	CGS N_i (N_m)	GMRES(5) N_i (N_m)
		Symmetric Scaling		
1 (Thomas)	10 (10)	4 (8)	4 (8)	10 (10)
2 (Diagonal)	31 (31)	8 (16)	9 (18)	15 (15)
3 (Block Jacobi)	27 (27)	8 (16)	13 (26)	15 (15)
4 (SPAI 1)	22 (22)	8 (16)	10 (20)	20 (20)
5 (SPAI 3)	20 (20)	9 (18)	10 (20)	15 (15)
6 (SPAI 3C)	27 (27)	9 (18)	10 (20)	15 (15)
7 (SPAI 5C)	27 (27)	9 (18)	10 (20)	15 (15)
8 (SPAI 3T)	15 (15)	5 (10)	5 (10)	10 (10)
9 (SPAI 5T)	11 (11)	5 (10)	6 (12)	10 (10)
		Row Scaling		
1 (Thomas)	10 (10)	4 (8)	4 (8)	10 (10)
2 (Diagonal)	26 (26)	8 (16)	8 (16)	15 (15)
3 (Block Jacobi)	25 (25)	8 (16)	11 (22)	15 (15)
4 (SPAI 1)	22 (22)	9 (18)	9 (18)	15 (15)
5 (SPAI 3)	20 (20)	8 (16)	10 (20)	20 (20)
6 (SPAI 3C)	26 (26)	9 (18)	10 (20)	15 (15)
7 (SPAI 5C)	26 (26)	9 (18)	10 (20)	15 (15)
8 (SPAI 3T)	11 (11)	5 (10)	7 (14)	10 (10)
9 (SPAI 5T)	10 (10)	4 (8)	5 (10)	10 (10)

NOTES.—The number of iterations, N_i, are shown followed by the number of MATVECs, N_m, in parentheses. An entry of "NC" indicates no convergence.

TABLE 6

PROBLEM 5: $N_r = 1024$, $S_c = 10^2$, $S_s = 10^{-2}$

Preconditioner	CHEBYCODE N_i (N_m)	BiCGSTAB N_i (N_m)	CGS N_i (N_m)	GMRES(5) N_i (N_m)
		Symmetric Scaling		
1 (Thomas)	22 (22)	9 (18)	8 (16)	20 (20)
2 (Diagonal)	22 (22)	9 (18)	8 (16)	20 (20)
3 (Block Jacobi)	7 (7)	3 (6)	3 (6)	5 (5)
4 (SPAI 1)	NC	NC	NC	NC
5 (SPAI 3)	NC	42 (84)	63 (126)	60 (60)
6 (SPAI 3C)	NC	41 (82)	117 (234)	80 (80)
7 (SPAI 5C)	NC	64 (128)	168 (336)	170 (170)
8 (SPAI 3T)	NC	NC	NC	NC
9 (SPAI 5T)	NC	NC	NC	NC
		Row Scaling		
1 (Thomas)	14 (14)	5 (10)	5 (10)	10 (10)
2 (Diagonal)	14 (14)	5 (10)	5 (10)	10 (10)
3 (Block Jacobi)	5 (5)	3 (6)	3 (6)	5 (5)
4 (SPAI 1)	9 (9)	5 (10)	6 (12)	10 (10)
5 (SPAI 3)	6 (6)	3 (6)	3 (6)	5 (5)
6 (SPAI 3C)	6 (6)	3 (6)	3 (6)	5 (5)
7 (SPAI 5C)	6 (6)	3 (6)	3 (6)	5 (5)
8 (SPAI 3T)	9 (9)	5 (10)	5 (10)	10 (10)
9 (SPAI 5T)	9 (9)	5 (10)	5 (10)	10 (10)

NOTES.—The number of iterations, N_i, are shown followed by the number of MATVECs, N_m, in parentheses. An entry of "NC" indicates no convergence.

TABLE 7

PROBLEM 6: $N_r = 1024$, $S_c = 10^{-2}$, $S_s = 0$

Preconditioner	CHEBYCODE N_i (N_m)	BiCGSTAB N_i (N_m)	CGS N_i (N_m)	GMRES(5) N_i (N_m)
		Symmetric Scaling		
1 (Thomas)	1 (1)	1 (2)	1 (2)	5 (5)
2 (Diagonal)	24 (24)	8 (16)	14 (28)	15 (15)
3 (Block Jacobi)	24 (24)	8 (16)	14 (28)	15 (15)
4 (SPAI 1)	20 (20)	9 (18)	12 (24)	15 (15)
5 (SPAI 3)	20 (20)	9 (18)	12 (24)	15 (15)
6 (SPAI 3C)	24 (24)	8 (16)	14 (28)	15 (15)
7 (SPAI 5C)	24 (24)	8 (16)	14 (28)	15 (15)
8 (SPAI 3T)	12 (12)	5 (10)	5 (10)	10 (10)
9 (SPAI 5T)	9 (9)	3 (6)	4 (8)	10 (10)
		Row Scaling		
1 (Thomas)	1 (1)	1 (2)	1 (2)	5 (5)
2 (Diagonal)	23 (23)	8 (16)	10 (20)	15 (15)
3 (Block Jacobi)	23 (23)	8 (16)	10 (20)	15 (15)
4 (SPAI 1)	19 (19)	8 (16)	9 (18)	15 (15)
5 (SPAI 3)	19 (19)	8 (16)	9 (18)	15 (15)
6 (SPAI 3C)	23 (23)	8 (16)	10 (20)	15 (15)
7 (SPAI 5C)	23 (23)	8 (16)	10 (20)	15 (15)
8 (SPAI 3T)	11 (11)	5 (10)	5 (10)	10 (10)
9 (SPAI 5T)	9 (9)	3 (6)	3 (6)	10 (10)

NOTES.—The number of iterations, N_i, are shown followed by the number of MATVECs, N_m, in parentheses. An entry of "NC" indicates no convergence.

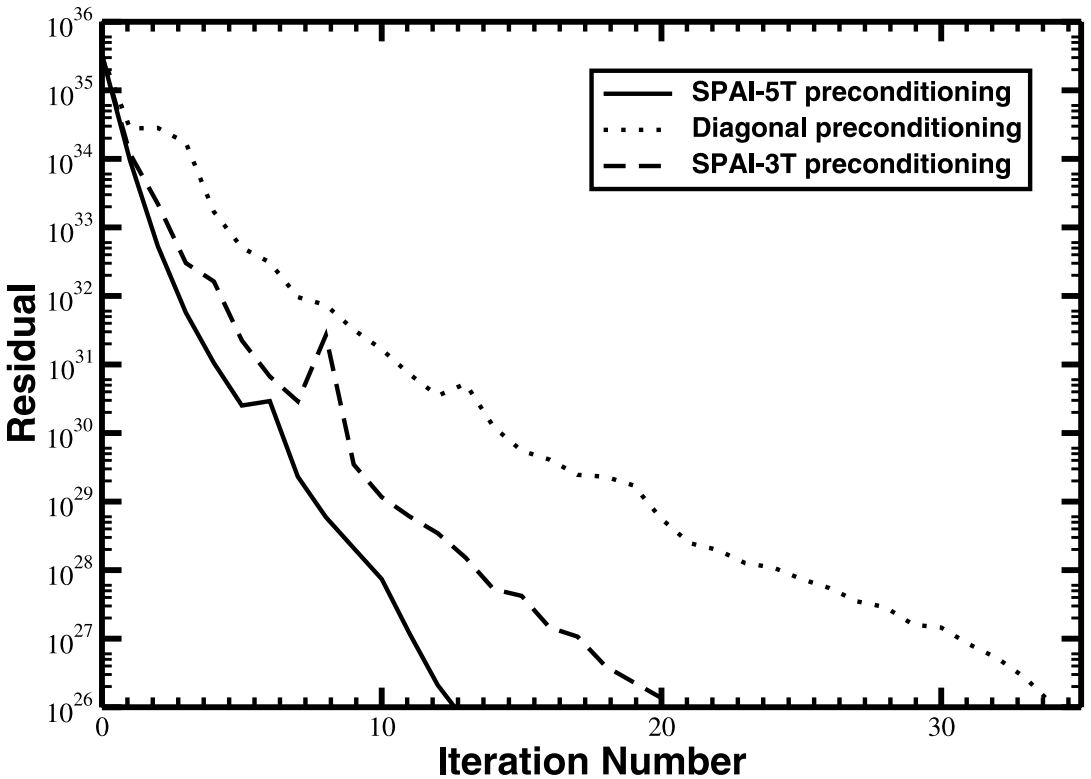

FIG. 7.—Convergence of BiCGSTAB method with row scaling on test problem 6.

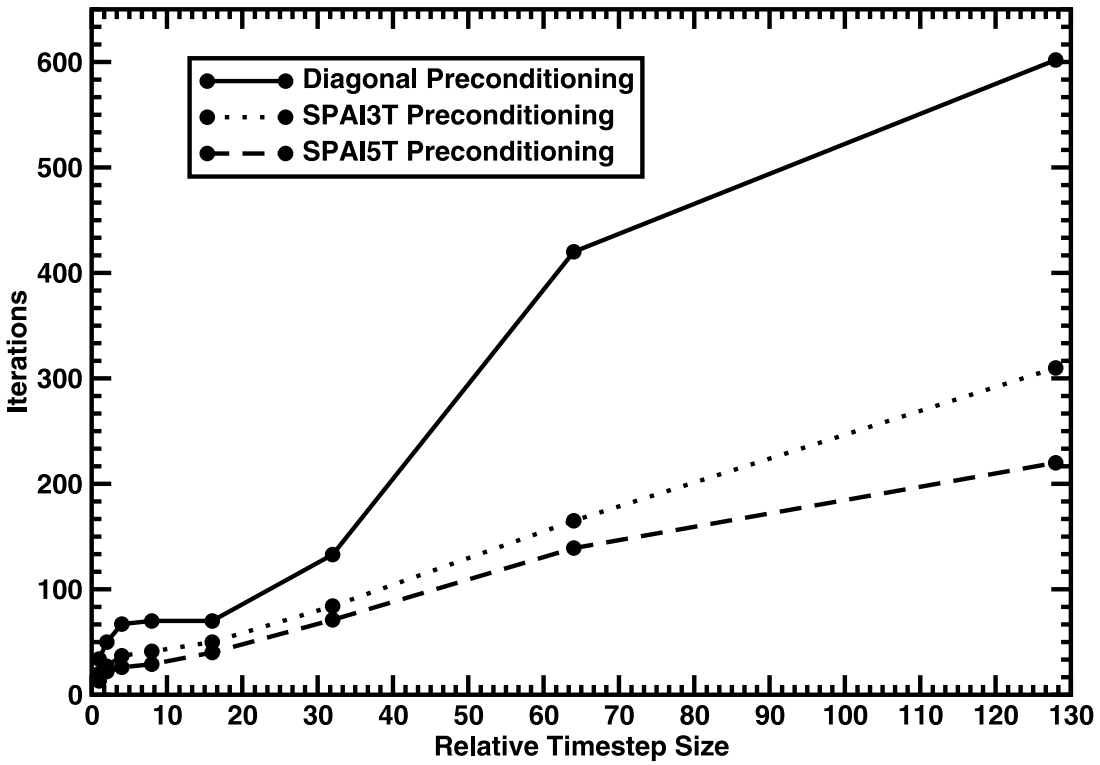

FIG. 8.—Iteration count for the BiCGSTB algorithm vs. relative time step size for the parameters specified in problem 6.

and row scaling. The performance of these two scaling methods is much closer than is seen in the other test problems. This is merely a consequence of the symmetry of the underlying problem.

7.5. *Changing the Size of the Time Step*

A final effect that we have considered is the influence of the time step size on the effectiveness of the preconditioners. For this representative study we have included only the BiCGSTAB algorithm with diagonal preconditioning and our two most robust SPAI preconditioners: SPAI3T and SPAI5T. We have scaled the time step relative to the original time step prescribed by equation (23). In Figure 8 the number of iterations to achieve convergence is plotted against the relative time step size (given by ratio of the new time step to the original time step) for problem 6. This study has considered a time step increased by factors of 2–128. We have also increased the maximum number of iterations allowed for convergence in order to allow for the increase in condition number of the system that usually accompanies an increasing time step size. The increase in the iteration count with time step size for all preconditioners is immediately apparent. However, the gap in iterative performance between the preconditioners increases steadily with increasing time step size. The merits of the SPAI preconditioners compared with diagonal preconditioning are obvious.

8. COMMENTS AND CONCLUSIONS

The test problems that we have considered in this paper were designed to span the range of behavior found in typical one-dimensional MGFLD problems. By examining the performance of preconditioner/iterative method combinations on these problems, we are able to offer some general conclusions about the efficacy of these combinations for use in MGFLD problems.

One of the most important results we have discovered is that, as a preconditioning step, row scaling seems to provide substantially better performance than symmetric scaling. This behavior is puzzling, and we have been unable to find an explanation for it. Nevertheless, the results clearly indicate that symmetric scaling slows convergence or, in many cases, prevents convergence in conjunction with the methods we have investigated. Further work is needed in order to obtain some mathematical insight into this behavior.

Second, we have found that SPAI preconditioners offer competitive iterative performance. In particular, preconditioners 8 and 9, which take into account the influence from nearby spatial zones, have yielded the best rate of convergence. These preconditioners can be computed with a minimum of work and could also be easily parallelized with only nearest-neighbor communication. The preconditioners with sparsity patterns that did not account for the spatial influence of adjacent zones did not, in general, offer the best convergence. The sole exception to this was problem 5, where the strong coupling between energy groups, due to scattering, gave preconditioners 4, 5, and 6 a slight advantage. These sparsity patterns could be combined with those of preconditioners 8 and 9 for problems that had both optically thick and thin regions. Since preconditioner 9 only slightly outperformed preconditioner 8 on some problems, the additional computational cost of solving a larger least-squares problem to compute the approximate inverse may not yield a significant speedup. This trade-off could also be problem-size and machine dependent and would require detailed comparisons on a particular platform.

Third, we have found that when the problem size is scaled up while keeping the CFL number fixed the number of iterations

seems to remain approximately constant for a specific problem. This is a consequence of the decrease in size of the off-diagonal elements of the linear system that accompanies the decrease in time step. The increasing diagonal dominance that results offsets the increasing condition number arising from the growing size of the linear system. It is unlikely that the number of iterations would remain constant for either time-independent problems or for problems where the size of the computational domain is increased while holding the spatial resolution fixed. The increase in iteration count with time step size is seen in Figure 8. However, our time step study does reveal that, at least, some of the SPAI preconditioners continue to be effective at larger time step sizes.

Finally, we found that similar numbers of MATVECs were required in order to achieve convergence for the CHEBYCODE and BiCGSTAB algorithms when these methods were used in conjunction with the best preconditioners. We therefore conclude that these algorithms have a comparable cost when implemented sequentially. The GMRES(5) and CGS algorithms seem to offer similar performance in most cases. However, when the potential costs of inner products on a distributed memory architecture are factored in, the CHEBYCODE algorithm may offer an advantage because of the minimal number of global reduction operations. The scalability of this algorithm is the focus of our continued investigations.

We would like to acknowledge financial support for this work under NASA CAN S5-153, DOE SciDAC cooperative agreement DE-FC02-01ER41185, and NASA grant NAG5-3099. Additional support from Academic Strategic Alliance Program and DOE under grant number B341494. We would also like to thank NERSC and NCSA for supercomputing support. We would also like to thank the referee for numerous helpful comments about the manuscript.

REFERENCES

Alme, M. L., & Wilson, J. R. 1974, ApJ, 194, 147
Ashby, S. F. 1985, CHEBYCODE: A FORTRAN Implementation of Manteuffel's Adaptive Chebyshev Algorithm, Research Rep. UIUCDCS-R-85-1203 (Urbana-Champaign: Dept. of Computer Science, Univ. Illinois)
Ashby, S. F., Manteuffel, T. A., & Otto, J. S. 1992, SIAM J. Sci. Comput., 13, 1
Ashby, S. F., Manteuffel, T. A., & Saylor, P. E. 1989, Bol. Inst. Tonantzintla, 29, 583
Baldwin, C., Brown, P. N., Falgout, R., Graziani, F., & Jones, J. 1999, J. Comput. Phys., 154, 1
Barnard, S. T., Bernardo, L. M., & Simon, H. D. 1999, Int. J. High Perf. Comput. Appl., 13, 107
Barrett, R., et al. 1994, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (Philadelphia: SIAM)
Benzi, M. 2002, J. Comput. Phys., 182, 418
Benzi, M., Cullum, J. K., & Tůma, M. 2001, SIAM J. Sci. Comput., 22, 1318
Benzi, M., Meyer, C. D., & Tůma, M. 1996, SIAM J. Sci. Comput., 17, 1135
Bruenn, S. W. 1985, ApJS, 58, 771
Bruenn, S. W., Buchler, J. R., & Yueh, W. 1978, Ap&SS, 59, 261
Cernohorsky, J., & Bludman, S. A. 1994, ApJ, 433, 205
Chow, E., & Saad, Y. 1997, SIAM J. Sci. Comput., 18, 1657
———. 1998, SIAM J. Sci. Comput., 19, 995
Elman, H. C., Saad, Y., & Saylor, P. 1985, SIAM J. Sci. Comput., 7, 840
Golub, G. H., & van Loan, C. F. 1989, Matrix Computations (3rd ed.; Baltimore: Johns Hopkins Univ. Press)
Gould, N. I. M., & Scott, J. A. 1995, On Approximate-inverse Preconditioners, Rep. RAL 95-026 (Oxfordshire: Rutherford Appleton Lab.)
———. 1998, SIAM J. Sci. Comput., 19, 602
Greenbaum, A. 1995, Iterative Methods for Solving Linear Systems (Philadelphia: SIAM)
Gropp, W., Lusk, E., & Skjellum, A. 1994, Using MPI: Portable Parallel Programming with the Message-Passing Interface (Cambridge: MIT Press)
Grote, M. J., & Huckle, T. 1997, SIAM J. Sci. Comput., 18, 838
Kelley, C. T. 1995, Iterative Methods for Linear and Nonlinear Equations (Philadelphia: SIAM)
Kittel, C., & Kroemer, H. 1980, Thermal Physics (New York: Freeman)
Levermore, C. D. 1984, J. Quant. Spectrosc. Radiat. Transfer, 31, 149
Levermore, C. D., & Pomraning, G. C. 1981, ApJ, 248, 321 (LP)
Manteuffel, T. A. 1978, Numer. Math., 31, 183
Mihalas, D., & Weibel-Mihalas, B. 1984, Foundations of Radiation Hydrodynamics (Dover: Mineola)
Minerbo, G. N. 1978, J. Quant. Spectrosc. Radiat. Transfer, 20, 541
Myra, E. S., Bludman, S. A., Lichtenstadt, I., Sack, N., & van Riper, K. A. 1987, ApJ, 318, 744
Saad, Y. 1996, Iterative Methods for Sparse Linear Systems (Boston: PWS Publ.)
Saad, Y., & Schultz, M. H. 1986, SIAM J. Matrix Anal. Appl., 7, 856
Sonneveld, P. 1989, SIAM J. Sci. Stat. Comput., 10, 36
Stone, J. M., Mihalas, D., & Norman, M. L. 1992, ApJS, 80, 819
Stone, J. M., & Norman, M. L. 1992a, ApJS, 80, 753
———. 1992b, ApJS, 80, 791
Swesty, F. D., Lattimer, J. M., & Myra, E. S. 1994, ApJ, 425, 195
Tang, W.-P. 1999, SIAM J. Matrix Anal. Appl., 20, 970
Thomas, L. H. 1949, Elliptic Problems in Linear Difference Equations over a Network, Watson Sci. Comput. Lab. Rep. (New York: Columbia Univ.)
Trefethen, L. N., & Bau III, D. 1997, Numerical Linear Algebra (Philadelphia: SIAM)
Turner, N. J., & Stone, J. M. 2001, ApJ, 135, 95
van der Vorst, H. 1992, SIAM J. Sci. Stat. Comput., 13, 631