Journal of Computational Physics **152,** 164–191 (1999)
Article ID jcph.1999.6240, available online at http://www.idealibrary.com on IDEAL®

A Multigrid Newton–Krylov Method for Multimaterial Equilibrium Radiation Diffusion[1]

William J. Rider,* Dana A. Knoll,* and Gordon L. Olson†

**Hydrodynamic Methods Group, Applied Theoretical and Computational Physics Division, Los Alamos National Laboratory, MS D413, Los Alamos, New Mexico 87545; and †Transport Methods Group, Applied Theoretical and Computational Physics Division, Los Alamos National Laboratory, MS D409, Los Alamos, New Mexico 87545*
E-mail: wjr@lanl.gov, nol@lanl.gov, glo@lanl.gov

Received June 29, 1998; revised January 26, 1999

We focus on a fully implicit, nonlinearly converged, solution of multimaterial equilibrium radiation diffusion problems. The nonlinear method of solution is a Newton–Krylov (generalized minimum residual, GMRES) method preconditioned by a multigrid method. The multigrid iteration matrix results from a Picard-type linearization of the governing equations. The governing equation is highly nonlinear with the principal forms of nonlinearity found in the fourth-order dependence of the radiation energy on temperature, the temperature dependence of the opacity, and flux limiting. The efficiency of both the linear and nonlinear iterative techniques is investigated. With the realistic time step control the solution of the linear system does not scale linearly with multigrid as might be expected from theory. In contrast, we find that the use of multigrid to precondition a Newton–Krylov (GMRES) method provides a robust, scalable solution for the nonlinear system. Also only through converging the nonlinearities within a time step does the solution method achieve its design accuracy.

Key Words: equilibrium radiation diffusion; multigrid; Newton–Krylov.

1. INTRODUCTION

In a large number of applications radiation transport plays a key role. This includes a variety of astrophysical phenomena, inertially confined fusion, combustion, and hypersonic flow. Often a diffusion approximation is made assuming isotropy and a small

[1] This work was performed under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory under Contract W-7405-ENG-36.

0021-9991/99

collisional mean free path of the transport. Integrating over all radiation energies (frequencies, i.e., a gray approximation) and assuming that the radiation energy is in equilibrium with the medium an equilibrium radiation description is found. We consider the integration of nonequilibrium radiation diffusion in [11].

Radiation diffusion is a highly nonlinear phenomenon. Despite this, the integration of the governing equation numerically is frequently accomplished with linearized PDEs where no attempt is made to converge the nonlinearities [2]. In addition to the simplicity of this approach there is a perception that effectively dealing with the nonlinearities with Newton's method is nearly intractable. Even with methods which deal with some of the nonlinearities, flux limiters are still used in a linearized fashion (i.e., the flux limiters are evaluated using explicit data) [2, 21]. Graziani and colleagues [1] have studied a variety of linear solvers, including a conjugate gradient preconditioned by multigrid on realistic applications.

We should note that other authors have integrated the equations of radiation diffusion converging nonlinearities [5, 4]. D'Amico [5] defines several algorithms, including a successive substitution method and a standard implementation of Newton's method. D'Amico's work is further distinguished by its focus on nonequilibrium radiation transport with flux limiting, but is only one-dimensional. Dai and Woodward [4] use a successive substitution method in conjunction with a multigrid solver and a second-order temporal differencing.

Our method converges the nonlinearities with a Newton-based method, but avoids the formation of the Jacobian matrix. Below we will introduce the physical problem with its governing equation and associated constitutive relations. Next, we discuss our nonlinear integration technique and the general properties of the numerical linear algebra used to solve radiation diffusion problems. This work is based on our earlier work combining multilevel preconditioners with the Newton–Krylov method [10]. Here and in our previous paper, we use the GMRES (generalized minimum residual method) [19, 20] as our Krylov method. Finally, we show results that indicate that multimaterial problems, substantial amounts of nonlinearity, and flux-limited diffusion pose no significant limitations to this methodology.[2]

2. DESCRIPTION OF THE PHYSICAL PROBLEM

Radiation diffusion can be posed in many forms. For instance, both the material temperature and the radiation energy density can be considered unknowns. It is instructive to start with the equations of nonequilibrium radiation diffusion where the energy equation is

$$\frac{\partial E}{\partial t} = \nabla \cdot \left(\frac{c}{3\kappa} \nabla E \right) + c\kappa (aT^4 - E) \tag{1a}$$

and the material temperature equation is

$$\frac{\partial C_v T}{\partial t} = c\kappa (E - aT^4), \tag{1b}$$

where κ is the opacity, a is the Stefan–Boltzmann constant, and c is the speed of light. Here we focus on the simpler setting where the material temperature is in equilibrium with

[2] We note that portions of this work have been presented at the 3rd IMACS International Symposium on Iterative Methods in Scientific Computation, July 1997, in Jackson, Wyoming [16], and the 5th Copper Mountain Conference on Iterative Methods, April 1998, Copper Mountain, Colorado [17].

the radiation energy density. If we assume that the temperature is in equilibrium with the radiation energy, $E = aT^4$, and sum (1a) and (1b), the result is

$$\frac{\partial(aT^4 + C_v T)}{\partial t} = \nabla \cdot \left(\frac{c}{3\kappa} \nabla aT^4 \right),$$

or the form we will use hereafter

$$\frac{\partial\left(\alpha + (1-\alpha)C_v E^{-3/4}\right)E}{\partial t} = \nabla \cdot (D(E)\nabla E). \tag{2}$$

Here, the dependent variable can be viewed as having two important limits where the energy of the system is dominated by either the material energy or the radiation energy. The parameter α is introduced to allow easy switching between limiting cases for the dependent variables. For simplicity we have chosen a set of units where $C_v = c = a = 1$. Next we will define the nonlinearities in the diffusion coefficient, $D = c/3\kappa$, as a nonlinear function of E [22, 14].

The energy dependence of D is found through the dependence of the opacity of the medium, κ, as a function of temperature. A common form for the temperature dependence of the opacity is $\kappa \propto 1/T^3 \rightarrow D \propto T^3$. Further complications are imposed by multimaterial problems where the opacity is a function of the atomic (Z) number of the medium (we choose Z^{-3} for all examples here).

Flux-limited diffusion is introduced to prevent transport faster than the maximum speed in the medium (the speed of light here). With flux-limited diffusion coefficients, the functional form of D will include the gradient of the energy density. The earliest form is due to Wilson [2] and contains the correct asymptotic behavior. As the gradients become small, the diffusion approximation (parabolic type of PDE) is recovered, while steep gradients recover a transport (hyperbolic type of PDE) form of the equation. Wilson's form is

$$\mathbf{D}_L = \frac{1}{1/\mathbf{D}(T) + |\nabla E|/E}. \tag{3}$$

The boundary conditions for the radiation diffusion are of a mixed form. Aside from symmetry conditions, we will also apply Milne or Robin (mixed) boundary conditions of the form

$$F_{\text{inc}} = \tfrac{1}{2} D(T)\nabla E + \tfrac{1}{4} E,$$

where F_{inc} is a prescribed flux and each quantity is evaluated on the boundary. It is notable that this boundary condition is a nonlinear function of the dependent variables.

Frequently, solutions to radiation diffusion problems involve sharp fronts known as Marshak waves [22]. While the equations are posed as a nonlinear diffusion equation, their solutions can be thought of as a wave problem. Numerical stability conditions and accuracy conditions may not be strongly related for this situation. As will be illustrated this condition manifests itself in the solutions and the numerical methods introduced below.

3. METHODS OF SOLUTION

3.1. *Linearized Methods*

For ease of presentation we will set $\alpha = 1$ in the following discussion. An explicit scheme can be defined by the forward Euler's method

$$E^{n+1} = E^n + \Delta t \nabla \cdot D(T^n) \nabla E^n,$$

with a (2-D) stability condition,

$$\frac{4D\Delta t}{h^2} < 1, \tag{4}$$

where $4D\Delta t / h^2$ is the Fourier number (Fo) and h is the mesh spacing (assuming $h = \Delta x = \Delta y$). Of course, one can go on to define an explicit algorithm of any temporal order of accuracy where stability is some multiple of the above condition. Unfortunately the stability-based time step is extremely restrictive, making an explicit integration impractical. Furthermore, as discussed earlier this stability condition may be poorly related to accuracy requirements for the solution of the radiation diffusion equations.

A frequently used approach to solve (1) is a semi-implicit method. The semi-implicit method is defined by a simple linearization of (1) using old time variables and backward Euler's method,

$$\delta E - \Delta t \nabla \cdot D(T^n) \nabla \delta E = \Delta t \nabla \cdot D(T^n) \nabla E^n, \tag{5a}$$

$$E^{n+1} = E^n + \delta E. \tag{5b}$$

This method is unconditionally stable (linearly). While it is stable, it is not nonlinearly converged, which can lead to inaccuracy for large time steps, $\Delta t \gg \Delta t_{\text{explicit}}$. We note that this form (5) is preferable to

$$E^{n+1} - \Delta t \nabla \cdot D(T^n) \nabla E^{n+1} = E^n, \tag{6}$$

because of error propagation characteristics of nonlinear problems. Equation (6) is equivalent to (5) if the equations are solved exactly. If one forms an effective equation for the error, the nonlinear residual forms the right-hand side. With the form based on δE the nonlinear residual goes to zero as the solution approaches a steady state while this is not true for the later differencing form. One can also define a second-order (Crank–Nicolson/implicit midpoint rule) version of this scheme by replacing (5a) with

$$\delta E - \tfrac{1}{2} \Delta t \nabla \cdot D(T^n) \nabla \delta E = \Delta t \nabla \cdot D(T^n) \nabla E^n.$$

As discussed later in the paper, this method does not yield the expected results in terms of accuracy, but also does not exhibit spurious solutions that are evident when integrating the equation with a large time step size (much larger than the explicit stability limit). Explanations for each of these effects will be given.

Another important aspect of the integration of these equations is time step control. Here, we employ a typical mechanism to accomplish this. The time step is estimated to provide a solution that evolves within a prescribed bound for relative energy change. The choice of

an upper bound for this ratio is chosen and the time step is dynamically adjusted to meet this as a requirement. Functionally, the form of the measure is

$$\eta = \frac{|E^{n+1} - E^n|}{E^{n+1} + E_{\text{floor}}},$$

where E_{floor} is a prescribed constant usually set equal to a multiple of the lower bound for the energy in a given problem. This quantity is evaluated everywhere on a grid and the largest value found is used for time step control. A typical target value for η might be 0.10–0.20.

The discrete spatial operator used here is a five-point Laplacian,

$$\nabla \cdot D(T)\nabla E \approx \frac{1}{\Delta x}\left[D\left(T_{i+\frac{1}{2},j}\right)\frac{E_{i+1,j} - E_{i,j}}{\Delta x} - D\left(T_{i-\frac{1}{2},j}\right)\frac{E_{i,j} - E_{i-1,j}}{\Delta x}\right] + \frac{1}{\Delta y}\left[D\left(T_{i,j+\frac{1}{2}}\right)\frac{E_{i,j+1} - E_{i,j}}{\Delta y} - D\left(T_{i,j-\frac{1}{2}}\right)\frac{E_{i,j} - E_{i,j-1}}{\Delta y}\right]; \quad (7)$$

here we take Δx and Δy as constant on a given grid.

The evaluation of the diffusion coefficient has three components: the material-dependent terms, the temperature-dependent term, and the flux limiting. Note, this division is dependent upon the simplifications used here and is not general. The material-dependent term is evaluated using a harmonic mean,

$$D_{i+\frac{1}{2},j} = \frac{2D_{i,j}D_{i+1,j}}{D_{i,j} + D_{i+1,j}}.$$

The dependence on the material present is taken so that $D \propto Z^{-3}$. We can evoke flux limiting with the form

$$D_L\left[E_{i,j}, E_{i+1,j}, D_{i+\frac{1}{2},j}\right]_{i+\frac{1}{2},j} = \frac{1}{\frac{1}{D_{i+\frac{1}{2},j}} + \frac{|E_{i+1,j} - E_{i,j}|}{\frac{1}{2}(E_{i,j} + E_{i+1,j})}}.$$

Other flux limiters can be applied here, most notably the form given by Levermore and Pomraning [12], without changing the basic linear or nonlinear algorithms as demonstrated in [11].

The form of the operator and the diffusion coefficient can have a profound impact on the linear algebra. Irrespective of the form of the diffusion coefficient, the linearized form of $\nabla \cdot D(T)\nabla E$ is symmetric positive definite. On the other hand, the Jacobian, $\partial(\nabla \cdot D(T)\nabla E)/\partial E$, is nonsymmetric and as the temperature front becomes sharper (near a Marshak wave), can become indefinite.

3.2. *Nonlinear Iteration Methods*

The starting point for the development of our nonlinearly convergent methods is the standard linearized solution introduced in the previous section. It is this typical linearization which, if applied iteratively to the same time step, constitutes a Picard-type (or successive substitution) nonlinear solver. Again, we note that several other works have introduced this method [5, 4]. Later, we show that this linearized solver forms the basis of the nonlinear preconditioning.

For the Picard-type nonlinear solver the backward Euler time differencing is

$$\delta E^{k+1} - \Delta t \nabla \cdot D(T^{n+1,k}) \nabla \delta E^{k+1} = (E^n - E^{n+1,k}) + \Delta t \nabla \cdot D(T^{n+1,k}) \nabla E^{n+1,k}, \quad (8a)$$

$$E^{n+1,k+1} = E^{n+1,k} + \xi \delta E^{k+1}, \quad (8b)$$

where k is the iteration index. For second-order time differencing using an implicit midpoint rule (rather than Crank–Nicolson), the finite difference form is

$$\delta E^{k+1} - \frac{1}{2}\Delta t \nabla \cdot D\left(\frac{T^n + T^{n+1,k}}{2}\right) \nabla \delta E^{k+1}$$
$$= (E^n - E^{n+1,k}) + \Delta t \nabla \cdot D\left(\frac{T^n + T^{n+1,k}}{2}\right) \nabla \left(\frac{E^n + E^{n+1,k}}{2}\right), \quad (9a)$$

$$E^{n+1,k+1} = E^{n+1,k} + \xi \delta E^{k+1}. \quad (9b)$$

In solving this equation we will refer to a nonlinear function that must be satisfied to some tolerence as

$$F(E^{n+1}) = (E^n - E^{n+1}) + \Delta t \nabla \cdot D\left(\frac{T^n + T^{n+1}}{2}\right) \nabla \left(\frac{E^n + E^{n+1}}{2}\right).$$

We employ an under-relaxation factor ξ defined by $\xi = \min(1, 1/\|\delta E/E\|)$ to robustly deal with convergence difficulties often encountered during the early stages of a nonlinear iteration. Both of the nonlinear iteration methods considered are inexact [6]; we use 10^{-2} times the current nonlinear residual to define the linear convergence tolerance. This limits the amount of work which is used to produce solutions that poorly approximate the nonlinear solution. Convergence within a time step is determined by the norm, $\|F(E)\|_2$, dropping below 10^{-6}.

As we will see preconditioning is the heart of the problem, and the Picard solver shown first (Algorithm 1) only differs from the Newton solver (Algorithm 2) in the matrix–vector multiply (Step 2c).

ALGORITHM 1 (Multigrid Picard-Type Nonlinear Solver).

1. Start the nonlinear iteration, $k = 0$.
2. Compute the nonlinear residual, $\mathbf{r} = -F(E)$.
   (a) Start the Krylov iteration to solve $A\delta E = \mathbf{r}$, $n = 0$. Initialize the Krylov vector with $\mathbf{v}_n = \mathbf{r}_n$.
   (b) Compute the preconditioned Krylov vector, $A\tilde{M}^{-1}\mathbf{v}$, using a multigrid V-cycle to approximate the solution to $A\mathbf{y}_n = \mathbf{v}_n$. $\tilde{M}^{-1}$ is the approximate inverse of A.
   (c) Perform the matrix–vector multiply through the operation $\mathbf{w}_n = A\mathbf{y}_n$.
   (d) Complete the Krylov iteration (constructing a new Krylov vector, $\mathbf{v}_{n+1}$) and compute the Krylov convergence—if converged, exit; otherwise $n := n + 1$ and go to (b).
3. Compute the (damped) update to the full nonlinear problem.
4. Check for nonlinear convergence—if converged, exit; otherwise, $k := k + 1$ and go to 2.

End Algorithm 1

This algorithm forms the foundation of our more sophisticated algorithm. In other words, a convergent nonlinear Picard-type iteration preconditions Newton's method. As will be seen shortly, the only difference between the two algorithms is the form of the matrix–vector multiply used in the Krylov algorithm. Both methods use a multigrid preconditioned Krylov method as an inner iteration with different connections to the full nonlinear problem. In developing our method we build upon our earlier efforts to combine multigrid as a preconditioner for Newton–Krylov methods [10].

First, we define the nonlinear functions that are being solved,

$$F(E^{n+1}) = \frac{E^{n+1} - E^n}{\Delta t} - \nabla \cdot D(T^{n+1}) \nabla E^{n+1}$$

for the first-order method and

$$F(E^{n+1}) = \frac{E^{n+1} - E^n}{\Delta t} - \nabla \cdot D\left(\frac{T^n + T^{n+1}}{2}\right) \nabla \left(\frac{E^n + E^{n+1}}{2}\right)$$

for the second-order implicit midpoint rule. Our goal is to execute an inexact Newton iteration within a time step. The updates to the dependent variables are found by approximately solving

$$J(E^{n+1,k}) \delta E^{k+1} = -F(E^{n+1,k}), \tag{10}$$

where k is the iteration index, and

$$E^{n+1,k+1} = E^{n+1,k} + \xi \delta E^{k+1} \tag{11}$$

to solve $F(E^{n+1}) = 0$. J is the Jacobian of $F(E)$ whose elements are defined by $J_{i,j} = \partial F(E_i)/\partial E_j$. To implement a Krylov method we only need to represent the matrix–vector product rather than explicitly represent the matrix. This allows the definition of the matrix-free (Jacobian-free) algorithm [3] with an approximation,

$$J\mathbf{v} \approx \frac{F(\mathbf{E} + \epsilon \mathbf{v}) - F(\mathbf{E})}{\epsilon}, \tag{12}$$

where $\mathbf{v}$ is a Krylov vector and $\epsilon = \rho(1 + \|\mathbf{E}\|)$ and $\rho = 10^{-8}$ here.

In order for this algorithm to be effective, a preconditioner must be employed. In this case we need to approximate $J\tilde{M}^{-1}\mathbf{v}$ which is done in two steps:

1. Approximately solve the linear system $M\mathbf{y} = \mathbf{v}$, where we choose M as the linear system A from the Picard-type iteration with an approximate solution computed with a single multigrid V-cycle.
2. Approximate the Jacobian via

$$J\tilde{M}^{-1}\mathbf{v} = J\mathbf{y} \approx \frac{F(\mathbf{E} + \epsilon \mathbf{y}) - F(\mathbf{E})}{\epsilon}.$$

Here $\mathbf{y}$ is referred to as a preconditioned Krylov vector. Symbolically, this can be compactly represented as $J\tilde{M}^{-1}\mathbf{v}$ with $\tilde{M}^{-1}$ referring to the approximate inversion accomplished with the multigrid V-cycle. The overall Newton–Krylov iteration takes the symbolic form $(J\tilde{M}^{-1})(\tilde{M}\delta E) = -F(E)$, which is known as right preconditioning.

The chief advantage of this method is that the actual Jacobian is never formed. The element that is necessary for this approach to be successful is good preconditioning. This process should not be confused with the process of numerically approximating the elements of the Jacobian via numerically evaluated (Frechet) derivatives.

To summarize, we apply the Picard-type linearization of the governing equation as the preconditioner. This is simultaneously the most important and subtle aspect of our method. Despite the asymmetry and potential indefiniteness of the nonlinear system, the symmetric positive definite preconditioner is used. In using this approximation, the only presence of the true Jacobian is found in the matrix-free matrix–vector product in the Krylov iteration.

Symbolically this algorithm can be stated in the following way:

ALGORITHM 2 (Newton–Krylov with Picard-Type Multigrid Preconditioning).

1. Start the nonlinear iteration, $k = 0$.
2. Compute the nonlinear residual, $\mathbf{r} = -F(E)$.
   (a) Start the Krylov iteration to solve $J\delta E = \mathbf{r}$, $n = 0$. Initialize the Krylov vector with $\mathbf{v}_n = \mathbf{r}_n$.
   (b) Compute the preconditioned Krylov vector, $J\tilde{M}^{-1}\mathbf{v}$, using a multigrid V-cycle to approximate the solution to $A\mathbf{y}_n = \mathbf{v}_n$.
   (c) Perform the matrix–vector multiply through the operation $\mathbf{w}_n = [F(\mathbf{E} + \epsilon\mathbf{y}_n) - F(\mathbf{E})]/\epsilon$.
   (d) Complete the Krylov iteration (constructing a new Krylov vector, $\mathbf{v}_{n+1}$) and compute convergence—if converged, exit; otherwise $n := n + 1$ and go to (b).[3]
3. Compute the (damped) update to the full nonlinear problem.
4. Check for nonlinear convergence—if converged, exit; otherwise, $k := k + 1$ and go to 2.

End Algorithm 2

The only difference between the Picard-type and Newton iteration is the matrix-free implementation of the GMRES algorithm in Newton's method. Viewed in this light, the matrix-free Newton's method can be viewed as accelerating the convergence of the simpler Picard iteration. The convergence tolerance of the linear problem is adaptive on each nonlinear step. We make note that we favor the use of GMRES because of its superior stability properties [9].

We motivate the use of nonlinear solvers over the linearized solvers with a simple example. In a one-dimensional domain we apply a flux to the boundaries at $x = 0$ and $x = 1$. At $x = 0$ the boundary condition is

$$\tfrac{1}{4}E + \tfrac{1}{2}D_0\frac{\partial E}{\partial x} = \tfrac{1}{4} \times 10^4.$$

At $x = 1$ the boundary condition is

$$\tfrac{1}{4}E + \tfrac{1}{2}D_1\frac{\partial E}{\partial x} = \tfrac{1}{4}.$$

The diffusion coefficient depends on the temperature cubicly, $D(T) = T^3$, it is flux-limited, and $\alpha = 0$. The radiation energy is initially set equal to 1.

[3] This is the step that distiquished Algorithm 2 from Algorithm 1 and is the heart of the matrix-free Krylov method.

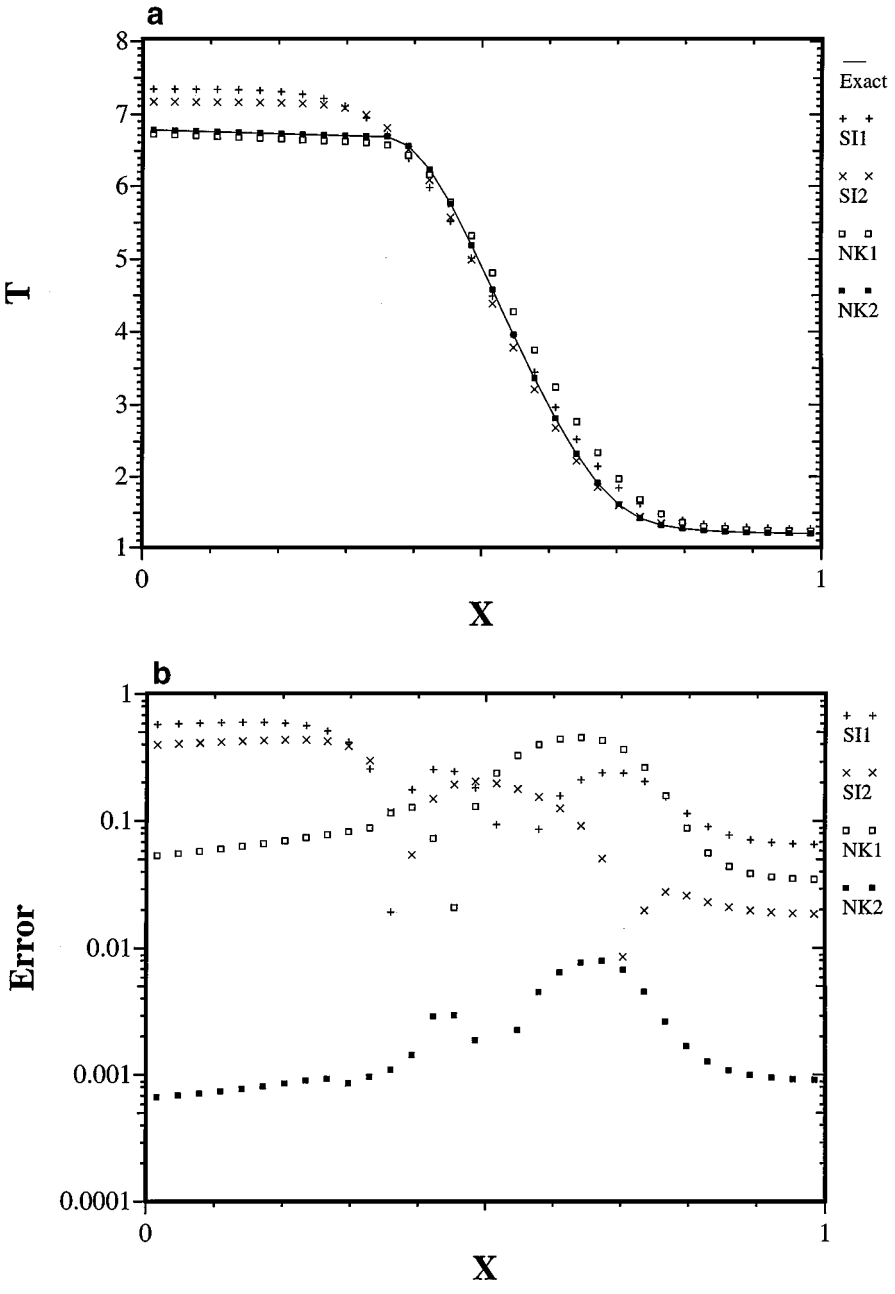

FIG. 1. The one-dimensional Marshak wave problem used to demonstrate the accuracy of various nonlinear iterative techniques. (a) Solutions at $t = 0.5$ with $\Delta t = 0.005$. (b) Absolute error $t = 0.5$ with $\Delta t = 0.005$.

Four methods are compared here: SI1, backward Euler semi-implicit; SI2, Crank–Nicolson semi-implicit; NK1, backward Euler Newton–Krylov; and NK2, implicit midpoint Newton–Krylov. Solutions are computed using a fixed time step (following an initial ramp from $\Delta t = 1 \times 10^{-6}$; Δt can be seen in the figures) and compared at $t = 0.5$. The ramp increases the time step by $10^{\frac{1}{16}}$ a time step up to the final time step size. Figure 1 shows a comparison of the results using the largest time step size, and the L_2 norm of the error

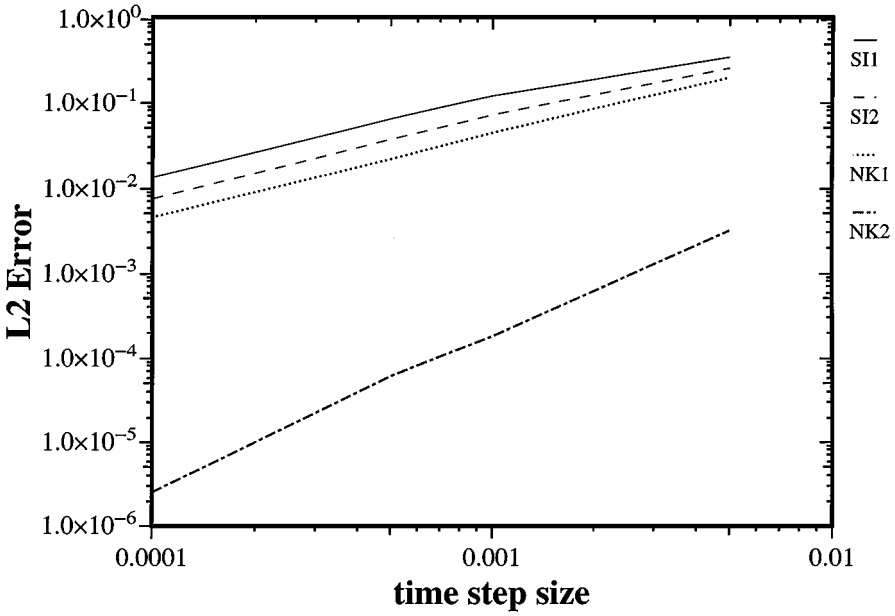

FIG. 2. Convergence of the solutions to the one-dimensional Marshak wave problem under time step size refinement.

(compared with a highly resolved solution using the second-order Newton–Krylov method with $\Delta t = 1 \times 10^{-7}$ and a tighter nonlinear convergence tolerance of 1×10^{-8}). The convergence rates for each of the four methods is given in Fig. 2. Clearly the nonlinearly converged Newton's method is superior when considering the L_2 error.

Perhaps more striking is the order of accuracy. Nonlinearly converged methods achieve something close to design accuracy. This is to say that the second-order Newton's method achieves nearly second-order convergence, while the same discrete form chosen for the semi-implicit method is only first-order accurate (marginally). Additionally, the second-order solution methods do not manifest oscillatory behavior often present when time steps are significantly larger than the explicit stability limit (for $\Delta t = 0.005$ the time step size gives $\mathrm{Fo} > 4000$). Stronger conclusions can be drawn from nonequilibrium radiation diffusion calculations [11] where the presence of the stiff source terms precludes even first-order convergence from methods not enforcing nonlinear convergence. Based on these results we will use the SI1 method for the semi-implicit method and the NK2 method for the nonlinear solution.

3.3. *Numerical Linear Algebra*

The implicit simulation of radiation diffusion problems requires the solution of large systems of linear equations. To be practical, these solutions must be of an iterative nature. The standard solution to this problem has been the incomplete Cholesky conjugate gradient (ICCG) algorithm introduced by Kershaw [8]. When introduced, ICCG was a significant improvement over earlier numerical linear algebra algorithms (such as point relaxation or ADI). Nevertheless, ICCG does not provide a scalable algorithm. As a grid is refined the asymptotic cost of solution scales as $N^{3/2}$ with ICCG for very poorly conditioned problems (N is the number of degrees of freedom being solved for; later we will use the notation N^s,

where s is the scaling exponent for the cost of the solution). For an introduction to Krylov methods we recommend Saad's book [19] and for Krylov methods and nonlinear methods see Kelly's book [7].

We will show results for three traditional Krylov subspace methods for linear problems. These three methods are categorized by their preconditioning. The first is the above-described ICCG, the second is the simpler diagonal scaled Jacobi preconditioning (SJCG), and the third used four passes of a composite Jacobi iteration (CJCG). This composite Jacobi iteration uses two different weights which provide optimal (idealized) smoothing for the two highest wavenumber error modes of the diffusion operator. These weights are successively one-half (damping the highest wavenumber on the grid) followed by one (damping the second highest wavenumber) with this combination of weights comprising one iteration. We have found this method to be as effective, and more computationally efficient than, ICCG.

When the system of linear equations is nonsymmetric as in the case of the Jacobian of the governing equation, the conjugate gradient method must be replaced by a more general method. The general class of conjugate gradient-like methods is known as Krylov–subspace methods. Among this class of methods the generalized minimum residual method (GMRES) is among the most robust. The properties of GMRES make it advantageous for use as the Krylov method here (conversely the properties of other methods such as CGS, BiCGStab, and other similar methods are problematic). The differences between GMRES and other Krylov methods are amplified with the use of matrix-free Newton methods [9]. This is caused by the enforcement of orthonormality within GMRES and its finite termination property (manifesting itself with improved computational stability). Additionally, GMRES has the property of finite termination and is more robust as a consequence. This is offset to some degree by the increased storage and work requirements imposed by GMRES. As noted before, preconditioning the linear problem is essential for efficiency.

To overcome the less than optimal scaling, multigrid algorithms are employed since they theoretically scale linearly. Unfortunately, multigrid algorithms are often less robust than Krylov methods. A way of overcoming this lack of robustness while still achieving scaling is to use multigrid as a preconditioner. For a multigrid preconditioned conjugate gradient method we use the acronym MGCG.

Our multigrid method was developed to be both simple and robust for multimaterial problems [18, 15]. In keeping with these principles, we use simple piecewise constant interlevel transfer operators and pointwise relaxation such as Jacobi or Gauss–Seidel iterations. Coarse grid equations are found through using control volume concepts to compute effective coarse grid diffusion coefficients from the previous fine grid (the process is similar to that found in [13]). While this multigrid is simple, its saving grace is that it is used to precondition a Krylov method. Previously, we have highlighted the degree to which the Krylov method returns this method to suitable robustness and scalability in severe circumstances [18, 15].

The chief issues regarding the numerical linear algebra are its properties of efficiency in relative terms as well as the scalability. In terms of work and run time, the rough equivalence of the different methods was determined (empirically determined, on an UltraSparc 2-200, SunOS 5.1, f77 compiler). For SJCG one iteration is normalized to one work unit. One CJCG iteration costs $\approx$4.5 SJCG iterations. One MGCG iteration costs $\approx$8.5 SJCG iterations. For ICCG the cost is more complex because the algorithm has two distinct steps, and because of the cost of the preparation for the preconditioning the algorithm is not competitive until the number of iterations becomes large. ICCG will not be considered further in this paper. Next, we will examine the efficiency, robustness, and scalability of the algorithms described above.

4. RESULTS

In this section we will produce quantitative performance results for the methods described above. First, using a semi-implicit method as a driver, the pure numerical linear algebra scaling efficiency is examined for several problems incorporating different levels of difficulty and nature of time step size control. Next, the nonlinear iterative methods will be examined on the same set of problems used to interrogate the numerical linear algebra. The cumulative result of these investigations will show that multigrid preconditioned Newton–Krylov methods provide a route to highly accurate efficient numerical solutions of multimaterial equilibrium radiation diffusion problems. Our goal is to develop methods for nonequilibrium radiation transport, but these simple models provide an adequate test bed for issues related to the solution of multidimensional nonlinear equations. Many of the principal issues present in the more complex models are represented here in the form of the nonlinearity of the opacity, flux limiting, and realistic time step control.

4.1. *Problem Descriptions*

To test the methods described above on multimaterial radiation physics we use a test problem exhibiting several important features. We use a set of regions with differing material compositions, with a rectangle for $x \leq 0.5$ and $y \leq 0.5$, $Z = 20.0$, another rectangle for $x \geq 0.75$ and $y \leq 0.25$, $Z = 100.0$, and a circle for $\sqrt{(x-0.75)^2 + (y-0.75)^2} \leq 0.15$, $Z = 50.0$ and $Z = 10.0$ everywhere else (the overall domain is $0 \leq x \leq 1$ and $0 \leq y \leq 1$). The material geometry is shown in Fig. 3. The initial condition for each problem is $E = 1$ everywhere.

Flux boundary conditions are applied to the left- and right-hand boundaries with the upper and lower boundaries being symmetric. On the right boundary $F_{\text{inc}} = 0.25$, and at the

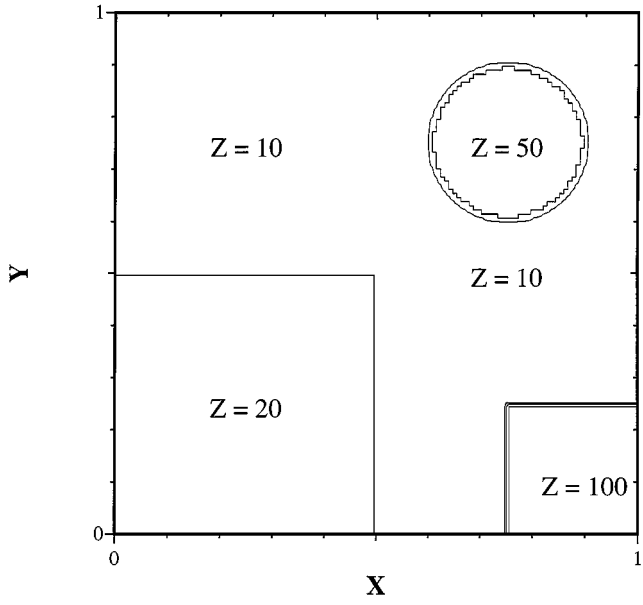

FIG. 3. A plot of the material topology for the radiation diffusion problems used in this paper. The atomic mass numbers, Z, are shown for each region.

left-hand boundary $F_{\text{inc}} = 2.5 \times 10^3$. Again, the initial condition is $E = 1$. Asymptotically, these fluxes are chosen so that the energies at the boundary approach 1×10^4 and 1 on the left and right boundaries respectively.

To provide some breadth for our results we will work with three physical models and integrate the equations using two different time step controls. The multimaterial geometry of the problem provides a three orders of magnitude jump in diffusion coefficient without considering the dynamics of the temperature evolution because of the cubic dependence of the diffusion coefficient on Z. This geometry is common to each of the problems described next.

1. Model 1 (M1) has the smallest amount of nonlinearity using the energy model ($\alpha = 1$), and the diffusion coefficient depends on the temperature linearly, $D(E) = E^{1/4} = T$. The temperature differences here provide an additional order of magnitude in jump in the diffusion coefficient.
2. Model 2 (M2) has a large degree of nonlinearity using the temperature model ($\alpha = 0$ and $C_v = 1$) with a cubic dependence of the diffusion coefficient on temperature, $D(E) = E^{3/4} = T^3$. With the fourth-order dependence of the flux on the dependent variable and the cubic dependence of the diffusion coefficient, another three orders of magnitude in potential jump are provided.
3. Model 3 (M3) uses the energy model ($\alpha = 1$), and the cubic temperature dependence for the diffusion coefficient, $D(E) = E^{3/4} = T^3$, but the diffusion coefficient is also flux limited using Wilson's limiter. The diffusion coefficient can exhibit three orders of magnitude jumps with respect to its temperature dependence, and the flux limiting can change the type of PDE locally from parabolic to hyperbolic.

Time step control uses two values $\eta = 0.10$ and $\eta = 0.50$ ($\eta = 0.50$ is larger than is typically applied in practice). Time steps are adjusted dynamically to attempt to achieve this change in energy over a time step with the proviso that the time step not grow more than 10% over any one time step. E_{floor} is set to 1 for all problems. In the range we explore in this paper the time step size is roughly linear with respect to η. Based on our experience the first-order accurate semi-implicit method will be five times more accurate at $\eta = 0.10$ than at $\eta = 0.50$. The second-order Newton–Krylov method will have errors two orders of magnitude smaller than the semi-implicit method at $\eta = 0.50$ and those errors will be reduced by a factor of 25 at $\eta = 0.10$. Two example calculations are shown in Fig. 4. M1's low relative nonlinearity is contrasted with the large amount of nonlinearity in M2. The larger amount of nonlinearity results in steeper fronts and generally more challenging numerical computations.

4.2. *Numerical Linear Algebra Performance*

First, we will examine the efficiency of the numerical linear algebra in idealized and more practical circumstances. Chief among our interests is the scaling of the work required to solve our problems as a function of grid resolution. As a measure of the scaling we will display the number of linear iterations used in solving the problems defined above with the semi-implicit algorithm.

The linear system that is solved arises from the discretization of a diffusion equation and is symmetric positive definite. If on a sequence of grids the diffusion equation is solved with a fixed Fourier number rather than a fixed η, the linear systems are (roughly) identical

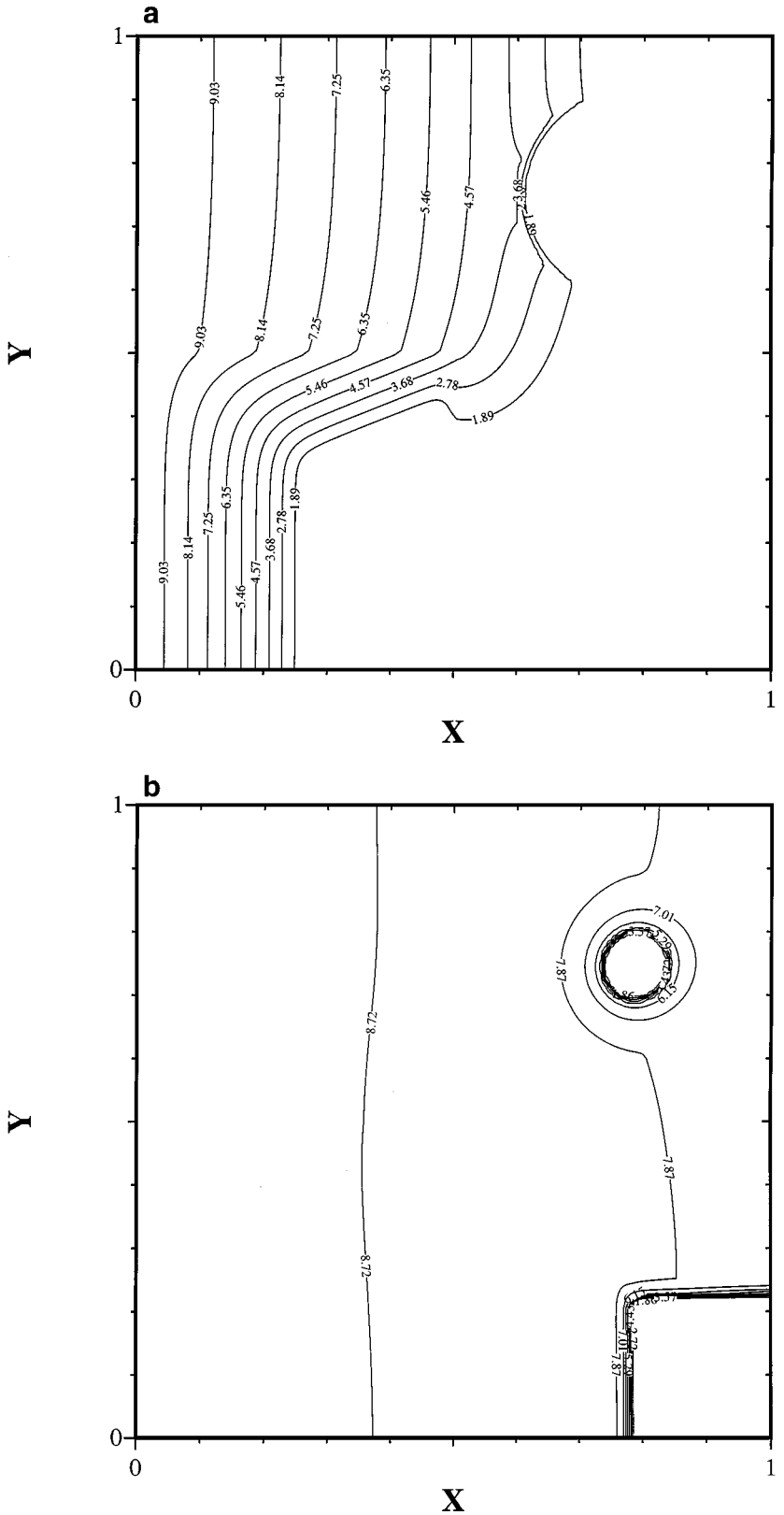

FIG. 4. The end of simulation time temperature solution for M1 and M2. Temperature ($E^{0.25}$) is plotted with the temperatures ranging from 1 to 10. (a) M1, $\eta = 0.50$, $t = 5.0$. (b) M2, $\eta = 0.50$, $t = 0.005$.

TABLE I
The Average Number of Iterations for the CJCG Method with a Fixed Fourier Number of 100

Grid	Average iterations
32×32	12.1
64×64	19.3
128×128	23.1
Scaling exponent	1.233

except for the mesh spacing. In these circumstances, we expect multigrid to behave linearly with Krylov–subspace methods behaving super-linearly.

Notably, the time step control offered by a fixed η does not yield equal Fourier numbers on sequences of refined grids. With this formof time step control, the time step size becomes a roughly linear function of the mesh spacing. For a fixed Fourier number the time step size is a quadratic function of the mesh spacing. Thus, for a fixed η, one would expect the Fourier number to follow $1/\Delta x$. In the results that follow, we report results in terms of a scaling coefficient, s, where s is determined by the power law fit of computational work compared with the degrees of freedom (N^s is proportional to the amount of work required to solve this system of linear equations).

Below, we show the number of linear iterations used for M1 and a time step control given a fixed Fourier number. As expected, the multigrid algorithm scales linearly ($s = 1.014$) while the Krylov–subspace methods scale super-linearly ($s = 1.233$) for a Fourier number of 100 (100 times the explicit stability limit). For a small Fourier number the super-linear scaling is weak although for the larger Fourier numbers the super-linear scaling approaches the worst-case result $N^{3/2}$. These results are given in Tables I and II. Under these conditions the multigrid method provides a significant advantage over the conjugate gradient method. As we will see this advantage is somewhat muted by applying a more realistic time step control to the problem.

Next, the scaling of the diagonally scaled Jacobi preconditioned conjugate gradient iteration is examined. For M3, the results are displayed in Fig. 5. For the other models SJCG behaves similarly as the problem evolves although M3 exhibits the worst scaling behavior. In each case, the time step control, $\eta = 0.10$, gives a super-linear scaling of $s = 1.290$–1.391 (see Table III). In the case where the looser time step control, $\eta = 0.50$,

TABLE II
The Average Number of Iterations for the MGCG Method with a Fixed Fourier Number of 100

Grid	Average iterations
32×32	5.60
64×64	5.73
128×128	5.82
Scaling exponent	1.014

Note. The scaling coefficient shows rough correspondence with the expected linear scaling.

TABLE III

The Average Number of Iterations for the SJCG Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	3.83	7.01	5.95
64×64	5.48	10.5	10.2
128×128	8.57	16.2	17.6
Scaling exponent	1.290	1.302	1.391

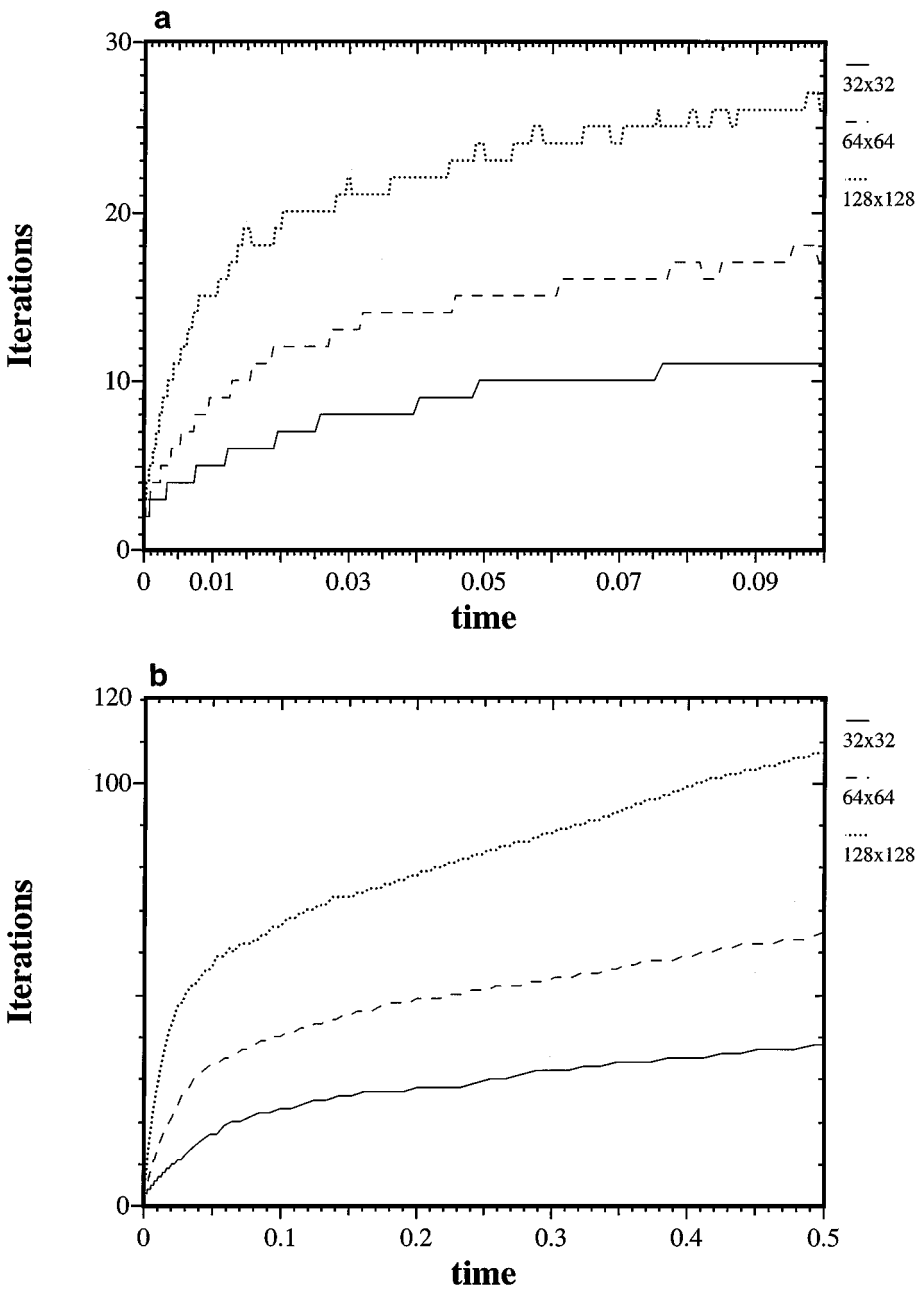

FIG. 5. Diagonal scaled Jacobi conjugate gradient's iteration count on M3. In general, the scaling of this linear algebra solver is not favorable in an idealized or practical case. The large iteration count is evidence of the weakness of the preconditioning. (a) $\eta = 0.10$. (b) $\eta = 0.50$.

TABLE IV
The Average Number of Iterations for the SJCG Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	6.77	26.5	12.5
64×64	12.3	46.8	27.3
128×128	24.1	74.9	55.2
Scaling exponent	1.458	1.375	1.536

is used the scaling exponentincreases to $s = 1.375$–1.536 (see Table IV). Thus, as the grid is refined one can expect to pay a continually higher price for the linear algebra solution.

For the composite Jacobi preconditioning conjugate gradient, the scaling is slightly better than that found for the diagonal Jacobi preconditioner. The temporal behavior of this solution method is nearly identical to the diagonal Jacobi preconditioned method with much lower iteration counts. In each case, the time step control, $\eta = 0.10$, gives a super-linear scaling for the method with an exponent $s = 1.242$–1.365 (see Table V). In the case where the looser time step control, $\eta = 0.50$, is used the scaling exponent increases to $s = 1.361$–1.505 (see Table VI). These scalings are nearly the same as the diagonal scaled Jacobi and thus both of these methods will provide increasingly greater costs to the linear algebra as the grid is refined. For M3 the costs for large grid sizes (or 3-D) may become prohibitive. Based on work/CPU time considerations, the CJCG method is superior for M2 while being slightly inferior for M1 and M3.

For a multigrid preconditioned conjugate gradient similar results can be given although the scaling exponents are slightly smaller. Figure 6 shows the behavior of this method on the problems used here in the worst light. The evolutionary behavior of the iteration count is better behaved for M1 and M2. Nonetheless, the $\eta = 0.50$ case with M3 is most efficiently solved with MGCG. For $\eta = 0.10$, the exponent $s = 1.142$–1.248 and for $\eta = 0.50$, $s = 1.224$–1.393 (see Tables VII and VIII). For the cases given here, the multigrid only results in greater economy than the other methods for the loose time step control on M2 and M3 although it is competitive for all problems. Certainly, should the scaling hold under further grid refinement, the multigrid method will be superior.

Because the linear algebra problem being solved changes character as the grids are refined, the linear scaling characteristic of the multigrid is not found here. The conclusion is that for easier problems with a tight time step control, the multigrid can provide a significantly more scalable solution for the linear algebra problem. This can be seen in the light of the relative cost of each iteration. Nonetheless, the multigrid solver does not provide a scalable solution

TABLE V
The Average Number of Iterations for the CJCG Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	1.34	1.96	1.79
64×64	1.76	2.96	2.83
128×128	2.62	4.60	4.88
Scaling exponent	1.242	1.308	1.365

TABLE VI
The Average Number of Iterations for the CJCG Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	2.11	3.62	6.74
64×64	3.65	7.55	14.8
128×128	7.25	12.4	19.7
Scaling exponent	1.419	1.361	1.505

when applied under these conditions. Ultimately, these costs will become exorbitant as grids are refined.

As noted earlier, in any case, the linear problems are becoming consistently different (and more difficult) as grids are refined with an energy-based time step control. A factor of 2 linear refinement provides an effective Fourier number that is twice that on the coarser grid. Thus, the linear algebra problems approach the worst-case scaling in the limit $\Delta x \to 0$.

The general scaling we see for the linear solvers is somewhat troubling, but as we will see there is hope for a more optimistic scaling. We would like to provide a prelude to the behavior we see with the nonlinear solvers. We note that the multigrid provides a significantly improved residual reduction in its early iterations when compared to the Krylov methods. The Krylov methods provide improved residual reduction late in their iterative sequence. A comparison between the solvers is shown in Fig. 7a (for the 32×32 grid) and the scaling for multigrid is shown in Fig. 7b. For nonlinear solvers the linear systems are solved less stringently early in the nonlinear iteration. For the efficiency of the nonlinear solver (inexact Newton or successive substitution) the early time iterative behavior is more important (note the 1×10^{-2} relative tolerance used in the course of the nonlinear solution algorithm).

4.3. *Nonlinear Solvers*

In this section, we compare the performance of the Picard and Newton–Krylov algorithms on the same set of radiation diffusion problems. We will compare the linear and nonlinear iterations as a function of time as well as the average number of iterations to provide a measure of algorithmic efficiency.

The chief conclusion from the results given below are that as grids are refined, the Picard iteration does not scale (where the Newton–Krylov method does) and this lack of scaling leads to a rapid expansion of relative computational effort per grid point as the grid is refined. Furthermore, the Picard iteration is not reliable, functionally failing

TABLE VII
The Average Number of Iterations for the MGCG Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	1.35	1.56	2.00
64×64	1.84	2.22	2.73
128×128	1.7	2.29	3.38
Scaling exponent	1.142	1.143	1.248

TABLE VIII

The Average Number of Iterations for the MGCG Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	1.81	2.76	4.60
64×64	4.70	6.59	8.74
128×128	2.86	4.93	8.50
Scaling exponent	1.336	1.224	1.393

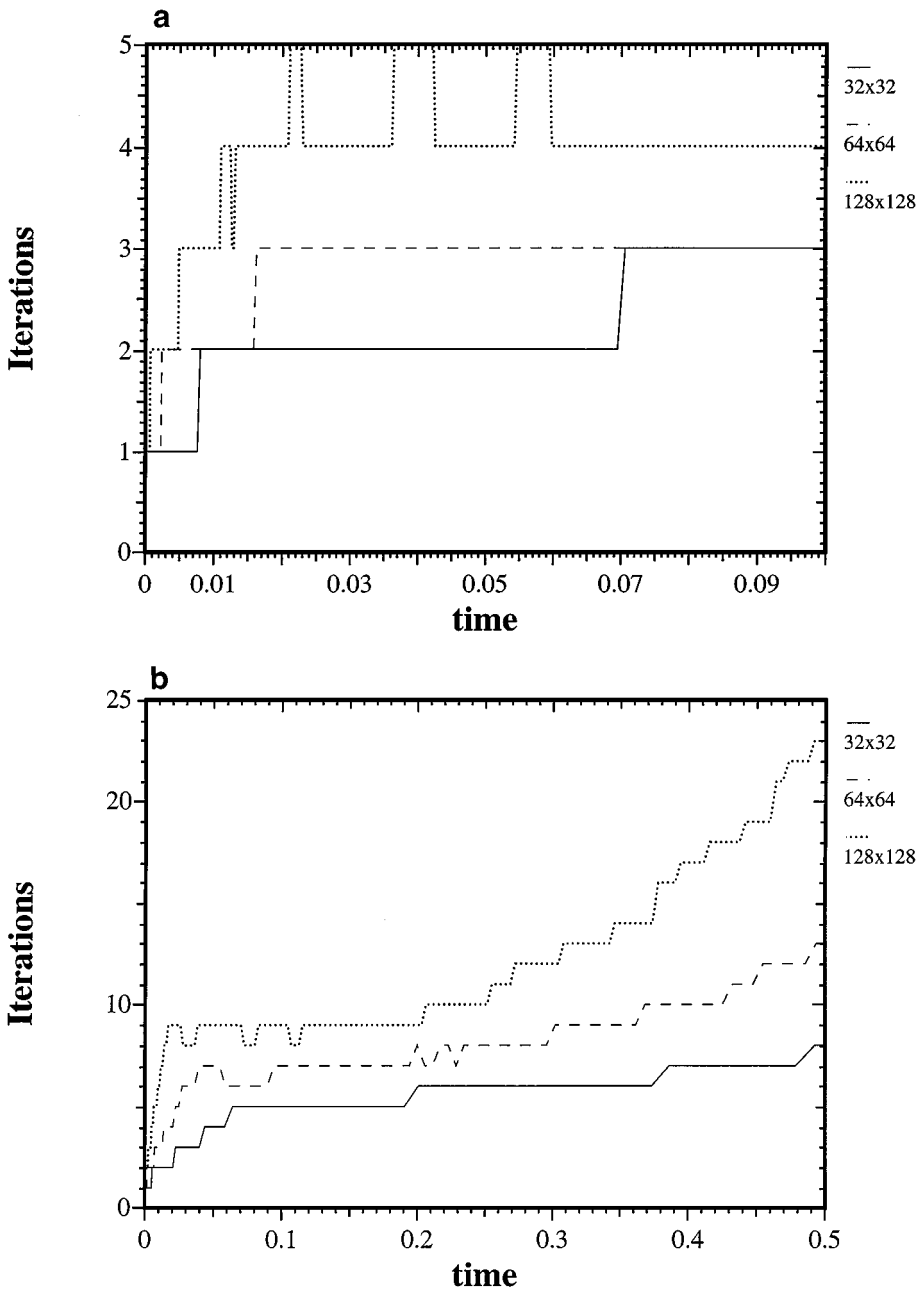

FIG. 6. Multigrid conjugate gradient's iteration count on M3. The low relative iteration count is evidence of the strong preconditioning provided by the multigrid. The linear scaling that might be expect here is not seen due to the impact of the time step control. (a) $\eta = 0.10$. (b) $\eta = 0.50$.

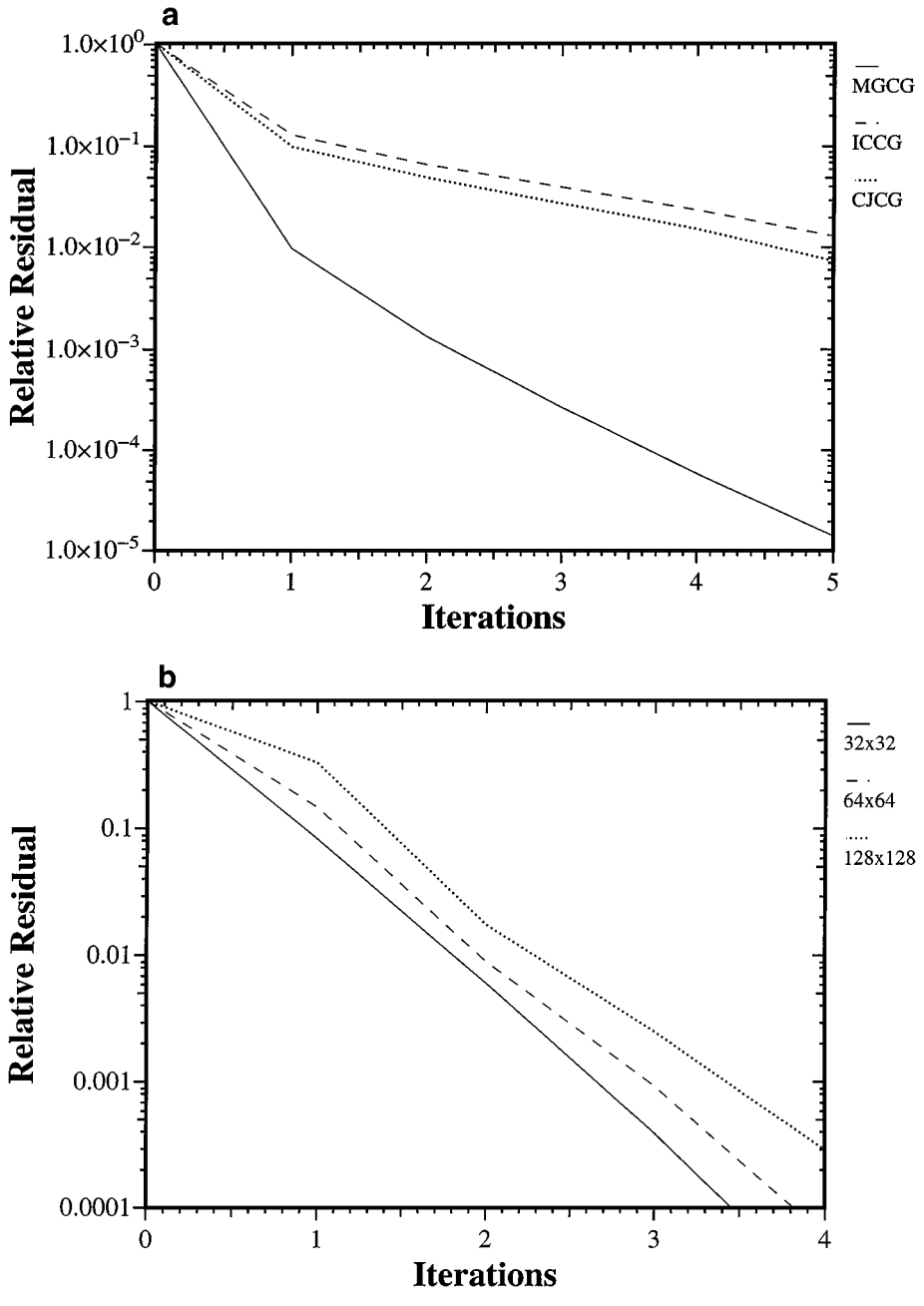

FIG. 7. The behavior of the linear solvers early in the iteration shows that the multigrid solver provides significantly higher convergence rates in the earlier iterations and scales well. (a) Comparing ICCG, CJCG, and MGCG convergence. (b) The scaling of multigrid convergence with grid size.

for more difficult physical problems. Neither of these failings make this iteration a poor preconditioner for the Newton–Krylov solver. We note that we might not expect grid scaling these circumstances as demonstrated in the previous section, but as we shall see the Newton–Krylov method recovers much of the linear scaling seemingly lost for the semi-implicit method's linear solution using the same machinery.

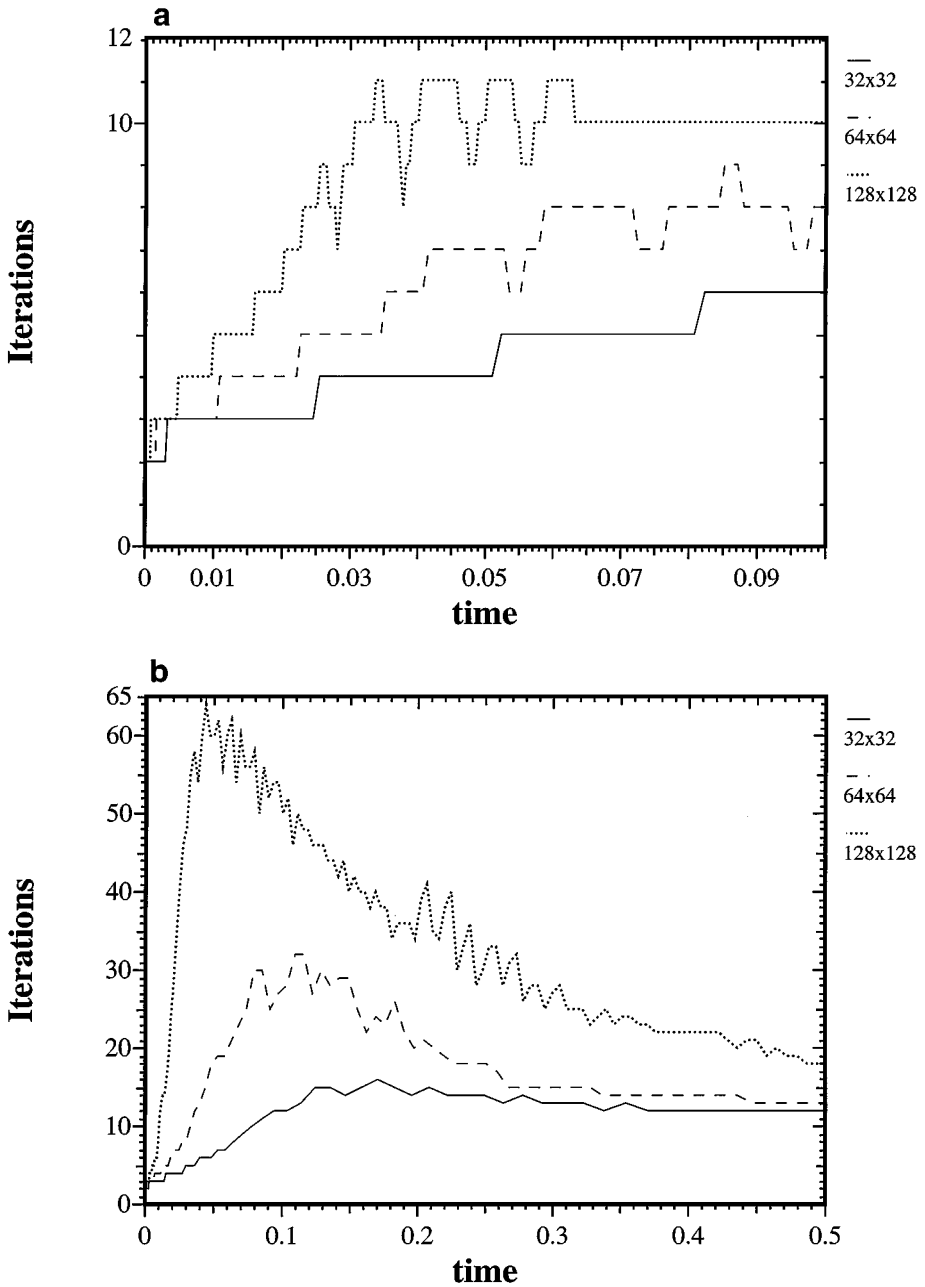

FIG. 8. Linear iterations with the Picard iteration on M3. (a) $\eta = 0.10$. (b) $\eta = 0.50$.

The lack of robustness mentioned above relates to the solution of M2 where the Picard iteration failed (or failed to converge in less than 50 nonlinear iterations). We present the results in terms of linear and nonlinear iterations as a function of time for M3 in Figs. 8 and 9. Most notably, the linear and nonlinear iteration counts are strongly correlated (further demonstrated by their similar scaling). M3 provides the method significant challenges and a good deal of temporal character where the problem is significantly more difficult near $t = 0.1$ for the $\eta = 0.50$ case. Each of these results has the following character: the number

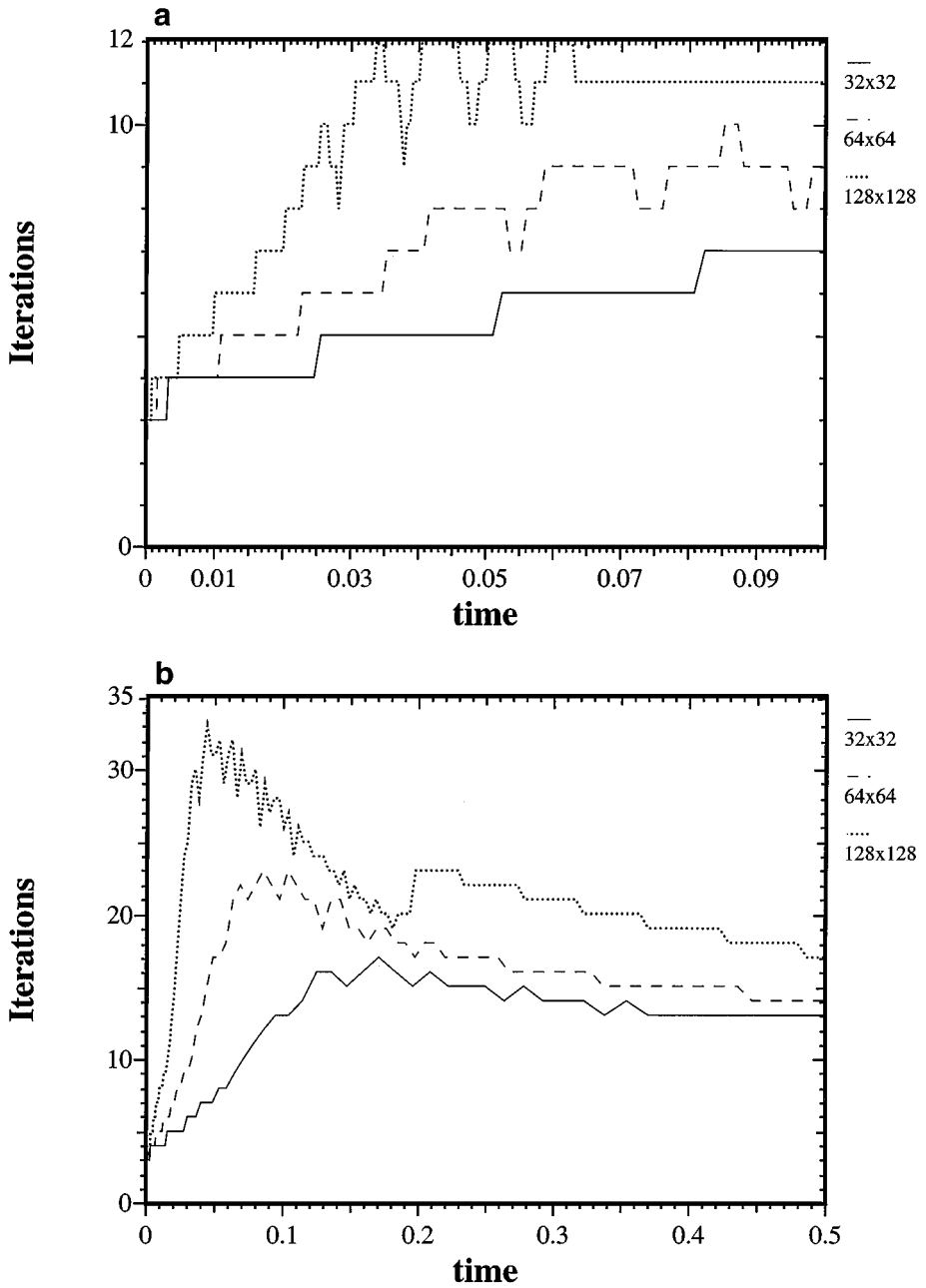

FIG. 9. Nonlinear iterations with the Picard iteration on M3. (a) $\eta = 0.10$. (b) $\eta = 0.50$.

of nonlinear iterations scales with an exponent of $s \approx 1.23$ with the linear iterations closely following this trend. This scaling exponent rises to $s \approx 1.30$ for $\eta = 0.50$. Note that the M1 results are much more favorable toward the Picard-type method as shown in Tables IX through XII. This is likely due to the lower level of nonlinearity intrinsic in this model. Nonetheless, the failure on M2 and the erratic behavior with M3 undermine the utilityof the Picard-type nonlinear solver.

The nonlinear solver's performance depends most critically on the choice of nonlinear solver rather than the linear iterative performance. The linear solver's performance in the

TABLE IX
The Average Number of Linear Iterations for the Picard-Type Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	3.05	—	3.26
64×64	3.38	—	4.09
128×128	3.66	—	7.06
Scaling exponent	1.0668	—	1.278

Note. For M2 the method does not converge in under 50 nonlinear iterations.

TABLE X
The Average Number of Nonlinear Iterations for the Picard-Type Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	4.05	—	4.26
64×64	4.38	—	5.69
128×128	4.66	—	8.05
Scaling exponent	1.051	—	1.230

Note. For M2 the method does not converge in under 50 nonlinear iterations.

TABLE XI
The Average Number of Linear Iterations for the Picard-Type Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	3.61	—	5.43
64×64	4.70	—	9.75
128×128	6.81	—	22.7
Scaling exponent	1.229	—	1.516

Note. For M2 the method does not converge in under 50 nonlinear iterations.

TABLE XII
The Average Number of Nonlinear Iterations for the Picard-Type Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	4.61	—	6.43
64×64	5.34	—	9.49
128×128	6.12	—	14.7
Scaling exponent	1.103	—	1.298

Note. For M2 the method does not converge in under 50 nonlinear iterations.

area of scalability is secondary, in that it determines the amount of work to solution. One must remember that the raw number of linear iterations must be minimized because the number of nonlinear function evaluations increases linearly with the number of linear iterations. As such we seek methods which minimize the linear iteration count (i.e., multigrid; one cycle is preferable and we want only enough work to meet our relative residual reduction criteria).

Having established the unreliability of Picard iterations in terms of solution robustness and scalability, we now examine the Newton–Krylov iteration using this same Picard iteration as a preconditioner. This is shown in Figs. 10 and 11 and Tables XIII through XVI. First,

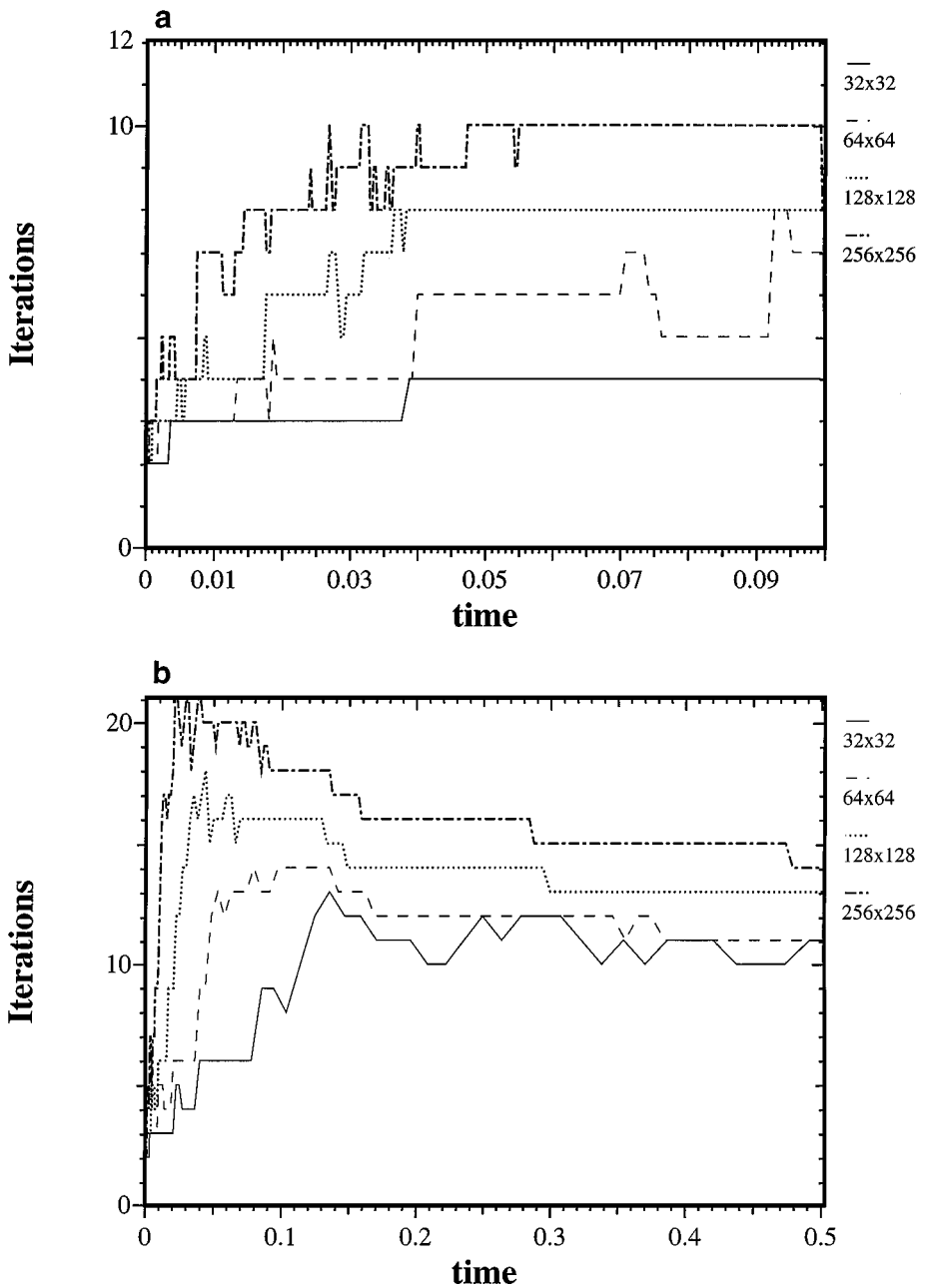

FIG. 10. Linear iterations with the Newton–Krylov iteration on M3. (a) $\eta = 0.10$. (b) $\eta = 0.50$.

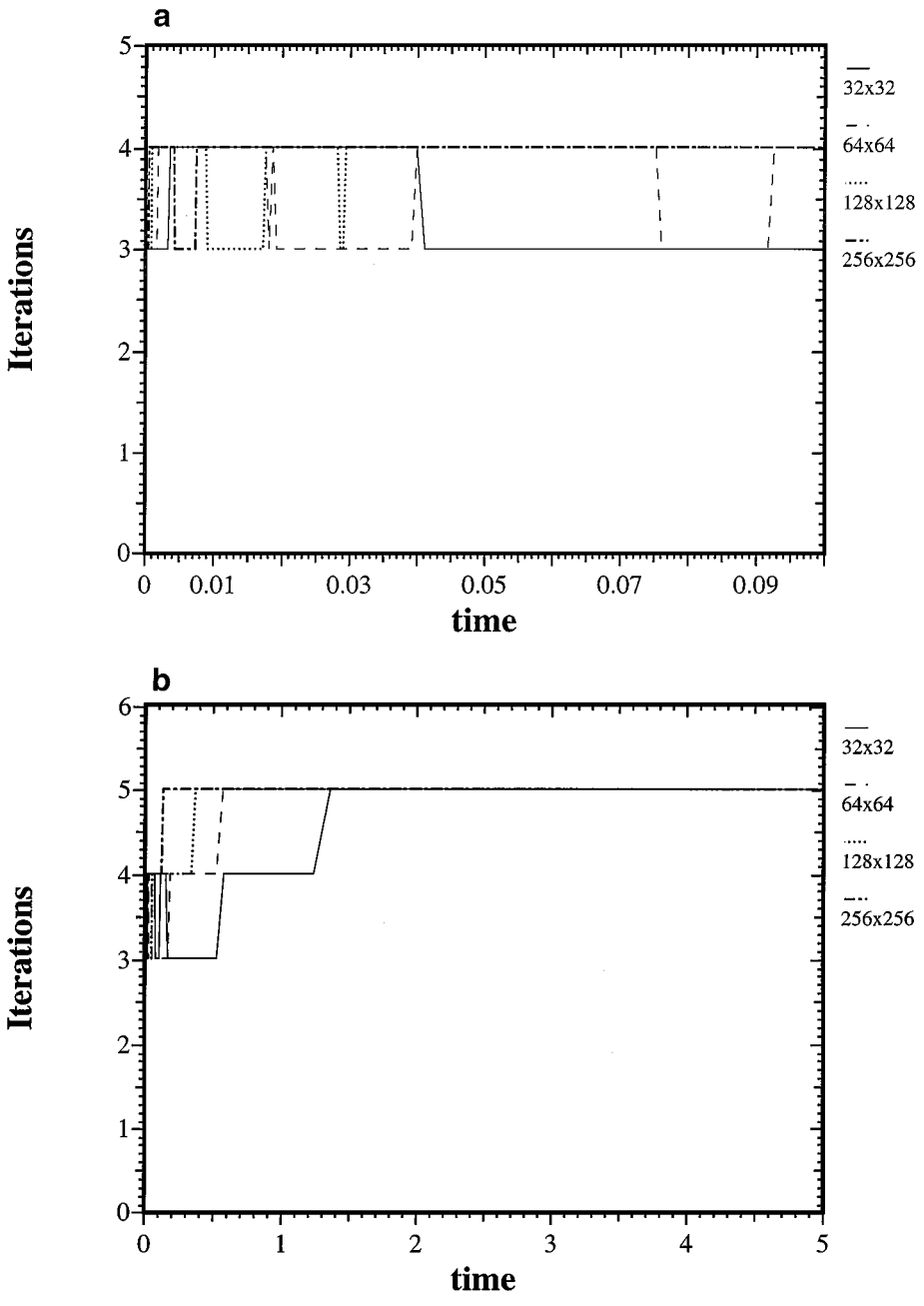

FIG. 11. Nonlinear iterations with the Newton–Krylov iteration on M3. (a) $\eta = 0.10$. (b) $\eta = 0.50$.

we note that the Newton–Krylov method successfully solves each problem (where Picard fails outright on M2). More impressively, the Newton–Krylov iteration shows almost no dependence on grid size for the nonlinear iteration count (the worst scaling is $s = 1.051$). Nevertheless the overall scaling is lower than that seen for the linearized problem. One obvious caveat is the larger overall number of iterations although if the scalings hold, the iteration count for the nonlinear case may win as the grid is further refined. Additionally, the accuracy of the resulting solution using the second-order Newton–Krylov method has

TABLE XIII
The Average Number of Linear Iterations for the Newton–Krylov Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	2.57	7.72	2.88
64×64	3.02	9.01	3.97
128×128	3.40	10.4	5.70
256×256	3.39	11.0	7.74
Scaling exponent	1.0668	1.087	1.240

TABLE XIV
The Average Number of Nonlinear Iterations for the Newton–Krylov Method with $\eta = 0.10$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	3.42	4.53	3.32
64×64	3.57	4.62	3.49
128×128	3.71	4.66	3.74
256×256	3.61	4.62	3.86
Scaling exponent	1.014	1.005	1.038

TABLE XV
The Average Number of Linear Iterations for the Newton–Krylov Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	3.35	14.3	4.71
64×64	3.96	15.2	6.61
128×128	4.59	16.2	9.37
256×256	5.34	17.5	12.8
Scaling exponent	1.111	1.048	1.242

TABLE XVI
The Average Number of Nonlinear Iterations for the Newton–Krylov Method with $\eta = 0.50$

Grid	Average iterations M1	Average iterations M2	Average iterations M3
32×32	3.42	5.29	3.66
64×64	3.63	5.37	3.97
128×128	3.79	5.48	4.27
256×256	3.99	5.66	4.52
Scaling exponent	1.036	1.016	1.051

been shown to be at least two orders of magnitude more accurate than the semi-implicit method (found using a comparable value of η).

The results show that the Newton–Krylov method does not exhibit a strong correlation between the linear and nonlinear iteration count. Thus, one can expect the number of nonlinear iterations to be well behaved without regard to the linear solver's iteration count. This independence may account for much of Newton's method's relative robustness. In the Picard iteration the linear and nonlinear iterations are closely correlated. Perhaps the most important difference can be seen by noting both the similarities between Figs. 8 and 10 showing linear iterations, while the nonlinear iterations shown in Figs. 9 and 11 are different.

Perhaps this is attributable to the previously identified problem of using a time step control that changed the effective Fourier number with grid size. We note that the energy ratio-based time step control provides a time step that is based on the physical character of the solution in a heuristic sense. We further conjecture that this provides for a nonlinear problem on the sequence of grids that is intrinsically similar leading to the similar algorithmic behavior. For the purely linear problem, the linear system will more closely follow the character of a heat conduction problem, thus providing for the stronger dependence of the performance on the effective Fourier number.

The bottom line is that the combination of multigrid preconditioning applied to the Picard linearization and the Newton–Krylov method provides a scalable solution to this class of radiation diffusion problems using realistic time step control mechanisms. Furthermore, methods that converge on the intrinsic nonlinearities in the physics are capable of providing number solutions with the naive theoretically expected rates of convergence. Just as importantly, the nonlinearly convergent methods allow one to achieve a design level of accuracy temporally, thus opening the door for greater than first-order time accuracy.

5. CONCLUSIONS

In summary, multigrid Newton–Krylov methods appear to be attractive for nonlinear initial value problems. The multigrid algorithm is critical to the efficient solution and using some sort of Krylov acceleration improves the robustness of the multigrid so that it can be used for this type of problem. Newton's method is significantly more efficient than a Picard iteration in providing accurate nonlinear solutions for this problem. This difference is particularly acute when flux-limited diffusion is employed or the nonlinearity is of high order.

It is particularly notable that we can use a Picard linearization as the preconditioner for our nonlinear algorithm. This frees one from having to form the actual Jacobian of the governing equations at any time. A Picard-type linearization is the typically employed discrete system solved in standard implementations. The only difference between a Picard nonlinear solver and our Newton's method is the presence of the matrix-free (and Jacobian-free) Krylov algorithm. We feel that this flexibility allows a substantially simpler path to the construction of a Newton's method for fairly general problems.

REFERENCES

1. C. Baldwin, P. N. Brown, R. Falgout, J. Jones, and F. Graziani, *Iterative Linear Solvers in a 2D Radiation-Hydrodynamics Code: Methods and Performance*, Lawrence Livermore National Laboratory Report, UCRL-JC-130933. [Also, submitted for publication]

2. R. L. Bowers and J. R. Wilson, *Numerical Modeling in Applied Physics and Astrophysics* (Jones & Bartlett, Boston, 1991).
3. P. N. Brown and A. C. Hindmarsh, Matrix-free methods for stiff systems of ODE's, *SIAM J. Numer. Anal.* **23**, 610 (1986).
4. W. L. Dai and P. R. Woodward, Numerical simulations for nonlinear heat transfer in a system of multimaterials, *J. Comput. Phys.* **139**, 158 (1998).
5. M. D'Amico, A Newton–Raphson approach for nonlinear diffusion equations in radiation hydrodynamics, *J. Quant. Spectrosc. Radiat. Transfer* **54**, 655 (1995).
6. R. Dembo *et al.*, Inexact Newton methods, *SIAM J. Numer. Anal.* **19**, 400 (1982).
7. C. T. Kelly, *Iterative Methods for Linear and Nonlinear Equations*, SIAM Frontiers in Applied Mathematics (SIAM, Philadelphia, 1995).
8. D. S. Kershaw, Incomplete Cholesky-conjugate gradient method for iterative solution of systems of linear equations, *J. Comput. Phys.* **26**, 43 (1978).
9. D. A. Knoll, P. R. McHugh, and V. A. Mousseau, Newton–Krylov–Schwarz methods applied to the tokamak edge plasma fluid equations, in *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering, Minneapolis, Minnesota, April 1994*, edited by D. E. Keyes, Y. A. Saad, and D. G. Truhlar (SIAM, Philadelphia, 1984).
10. D. A. Knoll and W. J. Rider, A multigrid preconditioned Newton–Krylov method, *SIAM J. Sci. Comput.*, in press (1999). [Also see LANL Report LA-UR-97-4013]
11. D. A. Knoll, W. J. Rider, and G. L. Olson, An efficient nonlinear solution method for nonequilibrium radiation diffusion, *J. Quant. Spectrosc. Radiat. Transfer*, in press. [Also see LANL Report LA-UR-98-2154]
12. C. D. Levermore and G. C. Pomraning, A flux-limited diffusion theory, *Astrophys. J.* **248**, 321 (1981).
13. C. Liu, Z. Liu, and S. McCormick, An efficient multigrid scheme for elliptic equations with discontinuous coefficients, *Commun. Appl. Numer. Methods* **8**, 621 (1992).
14. D. Mihalas and B. W. Mihalas, *Foundations of Radiation Hydrodynamics* (Oxford Univ. Press, London, 1984).
15. E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider, A second-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* **130**, 269 (1997).
16. W. J. Rider and D. A. Knoll, Solving nonlinear heat conduction problems with multigrid preconditioned Newton–Krylov methods, in *Iterative Methods in Scientific Computations, Jackson, Wyoming, July 1997*, edited by Junping Wang *et al.* (Elsevier, Amsterdam/New York, 1997).
17. W. J. Rider, D. A. Knoll, and G. L. Olson, A multigrid Newton–Krylov method for flux limited radiation diffusion, in *5th Copper Mountain Conference on Iterative Methods, Copper Mountain, Colorado, March 30–April 4, 1998.*
18. 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, Washington, DC, 1995). [Presented at the 33rd Aerospace Sciences Meeting and Exhibit]
19. Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS, Boston, 1996).
20. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
21. J. M. Stone, D. Mihalas, and M. L. Norman, ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. III. The radiation hydrodynamic algorithms and tests, *Astrophys. J. Suppl. Ser.* **80**, 819 (1992).
22. Y. B. Zel'dovich and Y. P. Raizer, *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena* (Academic Press, San Diego, 1966).