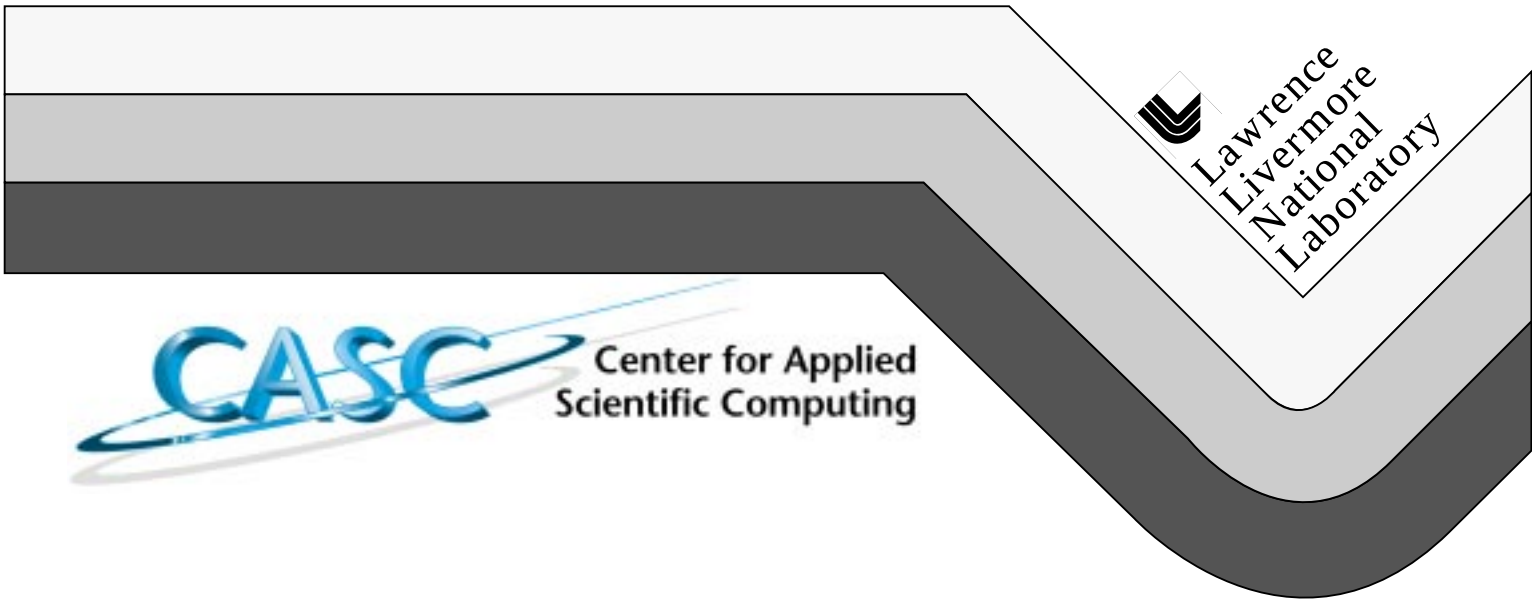
Implicit Solution of Large-Scale Radiation–Material Energy Transfer Problems

**Peter N. Brown, Britton Chang,
Frank Graziani, and Carol S. Woodward**



UCRL-JC-132831
January 1999

This work was performed under the auspices of the U.S. Dept. of Energy at LLNL under contract no. W-7405-Eng-48.

Implicit Solution of Large-Scale Radiation-Material Energy Transfer Problems

Peter N. Brown Britton Chang Frank Graziani
Carol S. Woodward

DEDICATION

Dedicated to David Young on the occasion of his 75th birthday.

Abstract

Modeling of radiation-diffusion processes has traditionally been accomplished through simulations based on decoupling and linearizing the basic physics equations. By applying these techniques, physicists have simplified their model enough that problems of moderate sizes could be solved. However, new applications demand the simulation of larger problems for which the inaccuracies and nonscalability of current algorithms prevent solution. Recent work in iterative methods has provided computational scientists with new tools for solving these problems. In this paper, we present an algorithm for the implicit solution of the multigroup diffusion approximation coupled to an electron temperature equation. This algorithm uses a stiff ODE solver coupled with Newton's method for solving the implicit equations arising at each time step. The Jacobian systems are solved by applying GMRES preconditioned with a semicoarsening multigrid algorithm. By combining the nonlinear Newton iteration with a multigrid preconditioner, we take advantage of the fast, robust nonlinear convergence of Newton's method and the scalability of the linear multigrid method. Numerical results show that the method is accurate and scalable.

Keywords: multigroup diffusion, multigrid, Newton-Krylov

1 Introduction

In this paper, we present a new numerical approach to the solution of very large-scale radiation-diffusion problems where energy can be transferred to a material through coupling terms in the radiation equation and a material temperature equation. These problems are important in modeling photon energy progression through an optically thick regime, a situation common in some laser and stellar fusion applications. Traditionally, solutions for these problems have been developed using operator split and time-lag techniques to reduce the coupled system of nonlinear equations to the solution of a series of linear problems. These solution techniques, however, lead to requirements of unacceptably small time steps. Furthermore, as computers

Iterative Methods in Scientific Computation II
David R. Kincaid et. al. (eds.), pp. 1–14.

ISBN 0–123–45678–9

have become faster, researchers have attempted to simulate larger problems, despite existing solution methods that did not scale well for increased numbers of unknowns.

For these reasons, we have developed a solution method for solving radiation-diffusion problems formulated in a fully implicit manner. The fully implicit formulation allows larger time steps to be taken without sacrificing accuracy. Furthermore, recent work in iterative methods has provided computational scientists with new tools for solving these problems — tools that scale well to large numbers of unknowns. In order to solve this fully implicit formulation, we employ ODE time integration techniques which then require an implicit solve for the solution at each time step. We use an inexact Newton method to solve the discrete nonlinear systems at each time step, with a preconditioned Krylov method for solving the linear Jacobian systems that arise within the Newton iterations. The Newton method provides fast nonlinear convergence, and the Krylov method gives a robust linear solver. Our preconditioner employs a robust, semicoarsening multigrid algorithm developed by Steve Schaffer [13, 4] which allows for significant differences in problem parameters from one spatial zone to the next.

Implicit solutions methods for radiation-diffusion problems have recently become active areas of research. In [10], a fully implicit formulation was examined for an integrated-in-energy form of the equations in one dimension. In this case, it was shown that the fully implicit form gave greater accuracy in shorter times than did traditional methods. The same authors also considered a fully implicit form of the integrated-in-energy equations in which an equilibrium condition was assumed and the resulting equation solved in a domain with many materials [11]. Similar advantages were seen for this fully implicit formulation. In this paper, we consider large-scale problems in parallel using the multigroup diffusion form of radiation-diffusion where we have an equation for each energy group, as well as one for material temperature. We show an effective, fully implicit solution strategy for these problems.

The rest of this paper is organized as follows. In the next section, we present the equations we are solving and the spatial and temporal discretization techniques used. In Section 3 we detail the iterative solvers used for implicit solutions at each time step, and in Section 4 we give some numerical results showing algorithm performance in one dimension and in three dimensions. Section 5 provides some concluding remarks.

2 Problem Formulation and Discretization

Our model for radiation-diffusion in optically thick regimes comes from assuming isotropic radiation and Fick's Law of diffusion in the Boltzmann equation to give the diffusion model [1],

$$\frac{\partial \varepsilon(\nu, \mathbf{x})}{\partial t} = \nabla \cdot \left(\frac{1}{3\sigma_T} \nabla \varepsilon(\nu, \mathbf{x}) \right) + \sigma_a \left(\frac{4\pi}{c} B(T, h\nu) - \varepsilon(\nu, \mathbf{x}) \right), \tag{1}$$

coupled to the material temperature equation,

$$\frac{\partial (C_v \rho T(\mathbf{x}))}{\partial t} = -\int_0^\infty \sigma_a \left(\frac{4\pi}{c} B(T, h\nu) - \varepsilon(\nu, \mathbf{x}) \right) d(h\nu), \tag{2}$$

where ε is the energy density at frequency ν and position $\mathbf{x} = (x, y, z)$, T is the material temperature at position $\mathbf{x}$, t is time, c is the speed of light, $B(T, h\nu)$ is the Planck function at frequency ν and temperature T, ρ is the material density, C_v is the material specific heat, σ_T is the total opacity, σ_a is the absorption opacity and h is Planck's constant. Note that we are

not allowing for scattering effects at this point (although we plan to add this physics later); therefore, the total opacity and absorption opacity are equal.

The Planck function is highly nonlinear, varying over many orders of magnitude. Figure 1 shows a typical plot of the Planck function over a range of energy values for a moderate value of temperature. In general, the Planck function peaks at the value $2.8kT$ where $k = 8.62 \times 10^{-5}$eV/K is the Boltzmann constant. For lower temperature values, the curve peaks earlier and declines faster than that shown, and for higher values of T, the peak occurs later.

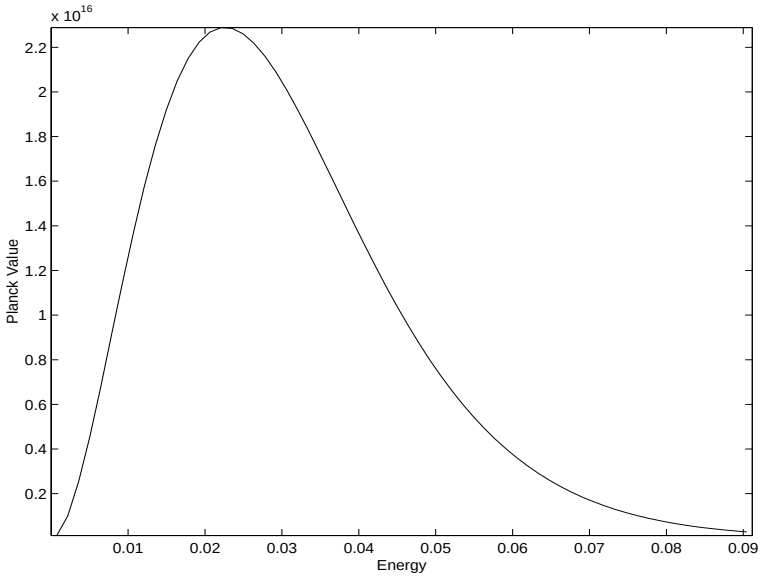


Figure 1: Planck function for temperature $= 10^{-3}$.

We divide frequency space into G cells and let ε_g be the approximate energy density at the center of cell g. We employ a cell-centered discretization in frequency space to give a series of equations for energy density at each discrete frequency and use the midpoint rule to approximate the integral over frequencies in (2).

For spatial discretization, we use a tensor product grid with N_x, N_y and N_z cells in the x, y and z directions, respectively. We employ cell-centered finite differences over this mesh and write our discrete equations in terms of a discrete diffusion operator,

$$\begin{aligned} L_{ijk}(\varepsilon_g) &= \left[\frac{1}{3\sigma_{T,i+1/2jk}} \frac{\varepsilon_{g,i+1jk} - \varepsilon_{g,ijk}}{\Delta x_{i+1/2jk}} - \frac{1}{3\sigma_{T,i-1/2jk}} \frac{\varepsilon_{g,ijk} - \varepsilon_{g,i-1jk}}{\Delta x_{i-1/2jk}} \right] / \Delta x_i \\ &\quad + \text{ y and z terms}, \end{aligned} \tag{3}$$

and a discrete source operator,

$$S_{ijk}(\varepsilon_g, T) = \sigma_{a,ijk} \left(\frac{4\pi}{c} B(T_{ijk}, h\nu_g) - \varepsilon_{g,ijk} \right). \tag{4}$$

Thus, our discrete scheme is to find $\varepsilon_{g,ijk}$ and T_{ijk} for each $g = 1, \ldots, G$ and $i = 1, \ldots, N_x; j = 1, \ldots, N_y; k = 1, \ldots, N_z$ such that,

$$\frac{\partial \varepsilon_{g,ijk}}{\partial t} = L_{ijk}(\varepsilon_g) + S_{ijk}(\varepsilon_g, T), \tag{5}$$

$$\frac{\partial(C_v \rho T_{ijk})}{\partial t} = -\sum_{g=1}^{G} S_{ijk}(\varepsilon_g, T)\Delta h \nu_g. \tag{6}$$

The system of equations in (5)–(6) is a system of nonlinear, coupled, ordinary differential equations. In order to allow for accurate time-stepping as well as larger steps than what traditional methods allow, we use an ODE time integrator to handle the temporal discretization. In particular, we use the PVODE package [7] developed at Lawrence Livermore National Laboratory. This package employs the fixed leading coefficient variant of the Backward Differentiation Formula (BDF) method [3, 9] and allows for variation in the order of the time discretization as well as in the time step size. Time step sizes are chosen to minimize the local truncation error, and thus give a solution that obeys a user-specified accuracy bound.

The general form of the BDF method for solving the system of ODEs, $\dot{y} = f(t, y)$, is given by,

$$y_n = \sum_{j=1}^{q} \alpha_j y_{n-j} + \Delta t_n \beta_0 f(t_n, y_n), \tag{7}$$

where q is the order of the method, α_j is a constant dependent on the order of the method and the values of previous time steps, β_0 is a constant depending only on the order of the method and $\Delta t_n = t_n - t_{n-1}$ is the time step. Thus, an implicit solve for y_n is required at each discrete time, t_n. In our case, y_n is composed of energy densities for each frequency and spatial cell and of material temperatures for each cell.

3 Solution Techniques

In this section, we present the solution techniques we use to solve the implicit, nonlinear problems for each discrete time.

Inexact Newton Method

We use an inexact Newton method to solve the nonlinear systems. In this method, the linear Jacobian system is solved only approximately. Suppose we want to solve the nonlinear equation,

$$F(u) = 0. \tag{8}$$

Let u^* be the true solution of (8), and let $F'\left(u^k\right)$ denote the Jacobian of the nonlinear function, F, at the kth approximation step. Then, Algorithm 3.1 describes an inexact Newton method applied to equation (8).

Algorithm 3.1

1. Let u^0 be an initial guess.
2. For $k = 0, 1, 2, \ldots$ until convergence, do
   (a) Using some method, compute a vector s^k satisfying

$$F'(u^k)s^k = -F(u^k) + r^k, \tag{9}$$

   where r^k is the error from the linear system solve.

(b) Set $u^{k+1} = u^k + s^k$.

In general, the standard Newton method gives quadratic convergence once the iterates to the solution fall within the radius of convergence for the sequence of iterates. For inexact Newton methods, we are guaranteed at least linear convergence to u^* under the condition that the linear system (9) is solved so that

$$\|r^k\| \leq \eta \|F(u^k)\|, \tag{10}$$

for all k, where $\eta \leq \eta_{max} < 1$ [8]. However, heuristic arguments given in [5] for the case of nonlinear systems arising in the context of implicit ODE integrators show that convergence obeys the relation,

$$\|u^{k+1} - u^*\| \leq C_k \|u^{k+1} - u^k\|(1+\epsilon), \tag{11}$$

where,

$$C_k = \|u^{k+1} - u^k\| / \|u^k - u^{k-1}\|, \text{ and } \epsilon > 0, \tag{12}$$

as long as $\|r^k\| \leq \delta$ for some δ sufficiently small.

The ODE integrator we employ, PVODE, uses a predictor-corrector form of the BDF methods. When solving stiff ODEs, the largest errors from the predictor step are typically in the stiff components. Thus, the corrector, which invokes the inexact Newton method, needs only to reduce errors in the stiff components in order to find the solution [5]. For this reason, PVODE uses a linear solver stopping condition of the form,

$$\|P^{-1} r^k\|_{wrms} \leq 0.05 \;\; C \tag{13}$$

where C is a constant between 0 and 1 related to the order of the ODE method, P is the left preconditioner, and $\|\cdot\|_{wrms}$ is the weighted root mean square norm,

$$\|x\|_{wrms} = \frac{\|Wx\|_2}{\sqrt{N}}, \tag{14}$$

for a vector x of length N. This residual stopping criteria is chosen so that the error in the solution to the ODE system arising from the linear solver is at most 5% of the error allowed in the local integration error test. The weighting matrix, W, is diagonal with components given by,

$$W_{ii} = \frac{1}{RTOL|y_i| + ATOL_i}, \tag{15}$$

where $RTOL$ is the requested relative tolerance, and $ATOL_i$ is the requested absolute tolerance for solution component y_i. By applying this weighting vector to norm calculations, we have a mechanism for handling scaling of unknowns within the ODE system. The root mean square norm is used here in order to increase the ODE integrator's scalability. As the problem size increases, the length of the solution vector increases causing the l_2 norm of that vector to increase and making the problem seem "harder" to solve. By dividing by the length of the vector, we offset some of this extra penalty.

Krylov Linear Solver

To solve the linear Jacobian systems at each Newton iteration, we use a Krylov iterative method. Combined with an inexact Newton method, these schemes are called Newton-Krylov methods. One big advantage of these methods is that the Krylov linear solver does not require knowledge of the matrix. Only the action of that matrix on vectors is needed. Since the system matrix is actually the Jacobian of the nonlinear function, this action can be approximated by taking differences of the nonlinear function of the form,

$$F'(u)v \approx \frac{F(u+\theta v) - F(u)}{\theta}, \tag{16}$$

where θ is a scalar. Thus, only the implementation of the nonlinear function is necessary, and matrix entries need never be formed or stored.

As our Krylov solver, we use GMRES [12]. Brown showed that an inexact Newton method using GMRES as the linear solver with finite differences as in (16) and the θ's chosen small enough, converges locally with the estimate for $k = 0, 1, \ldots$,

$$\|u^{k+1} - u^*\|_* \leq \alpha \|u^k - u^*\|_*, \tag{17}$$

where $\|\cdot\|_* = \|J(u^*)\cdot\|_2$, $\alpha < 1$ is given and $\eta < \alpha$ is used as in (10) [2]. However, heuristic arguments for the case of systems arising from the implicit integration of ODEs show that $\theta = 1$ works quite well [5], and that is the choice we use in our code.

The main advantage of using GMRES over other Krylov methods is that in the absence of roundoff error, the residual norm is nondecreasing with each iteration. However, the iteration can stagnate, and preconditioning is needed to enhance the robustness of the method.

Block Jacobi Preconditioner

We use a block Jacobi preconditioner on the left for our ODE Jacobian system. The Jacobian matrices are of the general form $(I - \Delta t \beta_0 J)$, where $J = \frac{\partial f}{\partial y}$ is the Jacobian of the nonlinear function, f, and the parameters Δt and β_0 are the current time step value and the order of the ODE integrator (see (7)). We impose an ordering of unknowns first by frequency and then by space within each frequency group. Equations are ordered in a similar manner. Recalling the definitions of the discrete divergence and source operators, defined in (3) and (4), the block form of the Jacobian of f is,

$$J = \begin{pmatrix} \frac{\partial L(\varepsilon_1)}{\partial \varepsilon_1} + \frac{\partial S(\varepsilon_1,T)}{\partial \varepsilon_1}\gamma_1 & 0 & \cdots & \cdots & 0 & \frac{\partial S(\varepsilon_1,T)}{\partial T}\gamma_1 \\ 0 & \ddots & \ddots & & \vdots & \vdots \\ \vdots & \ddots & & & \vdots & \vdots \\ \vdots & & & & 0 & \vdots \\ 0 & \cdots & \cdots & 0 & \frac{\partial L(\varepsilon_G)}{\partial \varepsilon_G} + \frac{\partial S(\varepsilon_G,T)}{\partial \varepsilon_G}\gamma_G & \frac{\partial S(\varepsilon_G,T)}{\partial T}\gamma_G \\ -\frac{\partial S(\varepsilon_1,T)}{\partial \varepsilon_1}\gamma_1 & \cdots & \cdots & \cdots & -\frac{\partial S(\varepsilon_G,T)}{\partial \varepsilon_G}\gamma_G & -\sum_{g=1}^{G}\frac{\partial S(\varepsilon_g,T)}{\partial T}\gamma_g \end{pmatrix},$$

where $\gamma_g = \frac{h\Delta\nu_g}{\rho C_v}$.

We precondition the Jacobian system with a block Jacobi preconditioner, ignoring couplings between the energy density and material temperature equations. So, an application of

the preconditioner involves inverting blocks consisting of the discrete diffusion operator and a contribution of local physics and blocks consisting of the derivative of the discrete source operator. These last blocks are diagonal and are easily inverted. Since multigrid methods are particularly effective on diffusion operators, and because they tend to be algorithmically scalable, we use a multigrid algorithm to invert the diagonal blocks corresponding to the energy density equations.

Semicoarsening Multigrid

We use the semicoarsening multigrid algorithm developed by Steve Schaffer [13] and the parallel implementation of this algorithm by Brown, Falgout and Jones [4]. This algorithm uses a combination of semicoarsening, plane-relaxation and operator-based interpolation. For simplicity, we describe the algorithm in two dimensions and comment on the extension to three dimensions subsequently.

We consider grouping the unknowns along each line on the spatial grid and denoting each line as either red or black with the colors alternating. Then, we consider coarsening in the y direction and let the unknowns with indices in the set $\{(i,j), j \text{ odd}\}$ be "red" unknowns and be used for the coarse grid. All remaining unknowns are "black" unknowns. This type of coarsening is semicoarsening, as the grid is coarsened in only one direction at a time and not all three, as in standard multigrid algorithms. Relaxation is done on the fine grid by performing red/black line relaxation where a tridiagonal solve for the equations over a line is conducted for each line.

The interpolation operator is motivated by the relationship of the error on red and black lines after a black line relaxation sweep. To see the relationship, let,

$$G_{j,j-1}e_{j-1} + G_{j,j}e_j + G_{j,j+1}e_{j+1} = 0, \tag{18}$$

be the error equation after relaxing on the block of equations for the jth line. Thus, the error on line j is related to the error on the neighboring red lines by,

$$e_j = -G_{j,j}^{-1}G_{j,j-1}e_{j-1} - G_{j,j}^{-1}G_{j,j+1}e_{j+1}. \tag{19}$$

This relationship shows exactly how to interpolate the error on the black lines from that on the red lines. However, since $G_{j,j}^{-1}$ is not sparse, the error at a point on a black line is coupled to the error on all points on the two red lines bordering the black line on which the point lies. We thus have a full interpolation operator, and we would prefer to have, instead, a sparse interpolation operator. We replace the operators, $G_{j,j}^{-1}G_{j,j-1}$ and $G_{j,j}^{-1}G_{j,j+1}$ with diagonal matrices having the same action on constant vectors as these operators. Constant vectors are used in this approximation since they best approximate the smoothest error. The transpose of this interpolation operator is used for restriction, and the Galerkin operator is used to generate the coarse grid operator. Thus, the coarse operator is the product of restriction, the fine grid operator, and prolongation (RAP).

In three dimensions, we group unknowns by planes and apply one V-cycle of the 2-D algorithm for the plane relaxation. Everything else is the same.

All these components are then put together in a standard V-cycle as our preconditioner for each energy equation block. Note that we apply only one V-cycle per frequency group per call to the preconditioner.

Recalculation of the Preconditioner

The matrix coefficients and multigrid interpolation operators used in the preconditioner are not recomputed for each Newton iteration. They are only recomputed when one of the following three criteria are met:

1. The Newton iteration fails to converge.
2. More than 20 time steps have passed since the last recomputation.
3. The relative change in $\Delta t * \beta$, where Δt is the time step value and β is related to the order of the ODE method (see (7)), is greater than 0.3.

As a result of these criteria, we find that the preconditioner is not recomputed very often. For more information on how these criteria were chosen, see [6].

4 Numerical Results

In this section, we present numerical results of applying our solution techniques to a radiation-diffusion test case from the literature. We look at both the accuracy of the method and the scalability of the solution algorithm.

Before we begin, however, we discuss scalability and the reasons to employ scalable algorithms. Our goal is to solve very large-scale problems. We therefore examine the algorithmic and implementation scalability of the solution techniques described above. A parallel code is considered *scalable* if the time to solve a problem with kN unknowns on k processors, $T(kN, k)$, is equal to the time to solve the same problem but with N unknowns on 1 processor, $T(N, 1)$. So, we are looking at scaling up the number of unknowns as we add more processors. One metric used to determine whether a code is scalable is *scaled efficiency*, E, given by

$$E = \frac{T(N,1)}{T(kN,k)}. \tag{20}$$

A scalable code will have scaled efficiency equal to 1.

In the context of iterative methods, we can look at algorithmic and implementation scalabilities separately. Both must be present for a parallel code to be scalable. Algorithmic scalability means that the work per iteration is $O(N)$, where N is the number of unknowns. In addition, the number of iterations must stay fixed as problem size increases. Implementation scalability means that the time to complete each iteration does not increase as the number of unknowns and processors increases.

Results in One Dimension

We use as our test case the one-dimensional Marshak two-temperature problem given in [14]. This problem is useful since a known analytical solution exists. The solution is given for an integrated-in-energy form of the multigroup diffusion equations where,

$$E(\mathbf{x}, t) = \int_0^\infty \varepsilon(\mathbf{x}, t, \nu) d\nu, \tag{21}$$

$$\int_0^\infty B(T, h\nu) d\nu = aT^4, \tag{22}$$

where a is the Stefan-Boltzmann constant and E is the total energy. The equations can then be given as,

$$\frac{\partial E(\mathbf{x})}{\partial t} = \nabla \cdot \left(\frac{1}{3\sigma_T} \nabla E(\mathbf{x}) \right) + \sigma_a \left(caT^4 - E(\mathbf{x}) \right), \tag{23}$$

$$\frac{\partial (C_v \rho T(\mathbf{x}))}{\partial t} = -\sigma_a \left(caT^4 - E(\mathbf{x}) \right). \tag{24}$$

This system is a special case of our code when we take $B = \frac{c}{4\pi} aT^4$ and use a single frequency group. The problem starts with a homogeneous initial condition for the total energy and the material temperature. A Robin boundary condition of the form,

$$E(\mathbf{0}, t) - \left(\frac{2}{3\sigma_T} \right) \frac{\partial E(\mathbf{0}, t)}{\partial \mathbf{x}} = 1, \tag{25}$$

is applied at $\mathbf{x} = 0$, and a homogeneous Dirichlet condition is applied at $\mathbf{x} = \infty$. In practice, this right-hand boundary condition is applied at $\mathbf{x} = 20$. Both the total and absorption opacities are taken to be 1.0.

Table 1 shows the absolute values (at steady state) of the differences between the computed and true radiation energy density and between the computed and true material temperature at the point $x = 0.1$. We note that because we are using the integrated-in-energy form of the equations, discretization in frequency is exact for this study. Each time the spatial mesh is doubled, the errors decrease by about half, indicating a spatial convergence rate of first order.

Table 1: Steady state spatial convergence data for the radiation energy density and material temperature at $x = 0.1$.

N_x	Error in Radiation Energy Density	Error in Material Temperature
200	5.0131e-3	5.1058e-3
400	2.5077e-3	2.5491e-3
800	1.2523e-3	1.2754e-3
1,600	6.2865e-4	6.4669e-4
3,200	3.1046e-4	3.1261e-4
6,400	1.5638e-4	1.5832e-4
12,800	7.7058e-5	7.8280e-5
25,600	3.6030e-5	1.4479e-5

We now use the original multigroup form of the equations to study the convergence of the frequency discretization. In this study we maintained a constant spatial mesh of $1,600$ cells, and considered adding more frequency groups. For each case, we took a discrete integral in frequency of the results and compared it against the integrated-in-energy results in the paper. Table 2 shows the same errors as above for this study. We see that as we increase the number of frequency groups, the errors decrease rapidly. This decrease is due to the enhanced ability of the finer-frequency mesh to approximate the Planck function.

Table 3 shows the number of time steps taken to achieve steady state, various iteration counts, and the total compute time for decreasing the spatial mesh size and using the integrated-in-energy form of the equations. Table 4 shows the ratios of the number of nonlinear iterations taken per time step and of the number of linear iterations taken per nonlinear iteration for this

Table 2: Steady state frequency convergence data for the radiation energy density and material temperature at $x = 0.1$.

Number of Groups	Error in Radiation Energy Density	Error in Material Temperature
1	5.0829e-1	5.9554e-0
2	5.0829e-1	5.9565e-0
4	4.4791e-1	5.5485e-0
8	1.5694e-1	2.5529e-0
16	3.4806e-2	6.2044e-1
32	6.6201e-3	2.7597e-1
64	6.8513e-3	1.6849e-1
128	1.4495e-4	1.3709e-2

study. As predicted above, only a few nonlinear iterations are required to converge the solution to the discrete time step. In addition, the ratio of the number of nonlinear iterations to the number of time steps remains fairly constant as the number of unknowns increases, indicating algorithmic scalability of the nonlinear iteration. Similarly, we see that the number of linear iterations required to converge the solution to the nonlinear iteration is very few, and that the ratio of the number of linear iterations to the number of nonlinear iterations also stays relatively constant as the problem size increases. Thus, algorithmic scalability is seen for both the linear and nonlinear iterations.

The number of time steps taken to solve for the same final time increases as the problem size increases. We believe this is due to a boundary layer effect. As more spatial cells are added, the discontinuity between the initial state at the first interior cell and the boundary value is refined. Initially, the ODE integrator takes smaller time steps to move past this discontinuity.

The numbers of preconditioner evaluations given in Table 3 shows that we are able to amortize the cost of preconditioner evaluation by using the same preconditioner for many time steps. We also see that as we double the number of unknowns, the total compute time approximately doubles. This last fact indicates that the implementation is scalable in a sequential sense.

Table 3: Total time steps, iteration counts and compute time data for spatial refinement study. Iteration counts indicate algorithmic scalability, and total compute times show implementation scalability.

N_x	Time Steps	Nonlinear Iterations	Linear Iterations	Preconditioner Evaluations	Preconditioner Solves	Time (sec)
200	464	503	684	49	1,152	8.82
400	509	555	778	56	1,294	22.44
800	555	588	774	59	1,334	46.52
1,600	581	614	815	55	1,401	83.65
3,200	644	680	897	59	1,551	190.29
6,400	624	670	859	60	1,504	352.61
12,800	632	670	891	62	1,531	751.38
25,600	717	748	997	64	1,732	1,614.67

Table 4: Iteration count ratios for spatial refinement study.

N_x	Nonlinear Iterations Per Time Step	Linear Iterations Per Nonlinear Iteration
200	1.08	1.36
400	1.09	1.40
800	1.06	1.32
1,600	1.06	1.33
3,200	1.06	1.32
6,400	1.07	1.28
12,800	1.06	1.33
25,600	1.04	1.33

Results in Three Dimensions

As our three-dimensional test case, we extend the one-dimensional problem described above by applying the same Robin boundary condition on the lower side for each of the y and z directions and a homogeneous Dirichlet condition on the upper side for each of the y and z directions. We used the integrated-in-energy form of the equations and ran to time $1,000$ for all cases.

Table 5 shows results from a scaled speedup study on the IBM SP2 (Technical Refresh) machine at Lawrence Livermore National Laboratory. In this case, we kept a constant number of spatial cells on each processor ($50 \times 50 \times 50$) and scaled up the problem by adding more processors. From the table, one sees that as we add more processors (and correspondingly, more unknowns), we see some increase in the number of time steps required to reach the final time. This may be due to the boundary effect described above. We also see corresponding increases in the number of nonlinear and linear iterations required for convergence. Table 6 shows the ratios of the number of nonlinear iterations taken per time step and of the number of linear iterations taken per nonlinear iteration for this study. Again, we see that these ratios hold relatively constant, indicating reasonable algorithmic scalability.

Table 5: Total time steps and iteration counts for parallel scalability study with $125,000$ spatial cells per processor (2 unknowns at each cell).

Number of Processors	Topology	Total Unknowns	Time Steps	Nonlinear Iterations	Linear Iterations	Preconditioner Evaluations
1	1× 1× 1	250,000	416	462	646	57
2	2× 1× 1	500,000	525	554	759	58
4	2× 2× 1	1,000,000	571	606	897	58
8	2× 2× 2	2,000,000	496	529	736	54
16	4× 2× 2	4,000,000	614	647	960	60
32	4× 4× 2	8,000,000	473	507	700	55
64	4× 4× 4	16,000,000	508	541	765	55
128	8× 4× 4	32,000,000	696	734	944	66

Figure 2 shows the scaled efficiency for this study. We see that the scaled efficiency degrades, ending at about 50% at 128 processors. We believe that this degradation is due to the nonlinearities in the problems we are solving. As we refine the mesh, we also refine the

Table 6: Iteration count ratios for parallel scalability study.

Processors	Nonlinear Iterations Per Time Step	Linear Iterations Per Nonlinear Iteration
1	1.11	1.40
2	1.06	1.37
4	1.06	1.48
8	1.07	1.39
16	1.05	1.48
32	1.07	1.38
64	1.06	1.41
128	1.05	1.29

nonlinearity. To see if we are scaling well for the linear iterations, we plotted the scaled efficiency of the time per linear iteration (the total run time divided by the number of linear iterations). Figure 3 shows this plot, and we see about 72% scaled efficiency at 128 processors. This measure, however, takes the ODE integrator time into consideration, so we expect that the linear iterations scale better than what is shown.

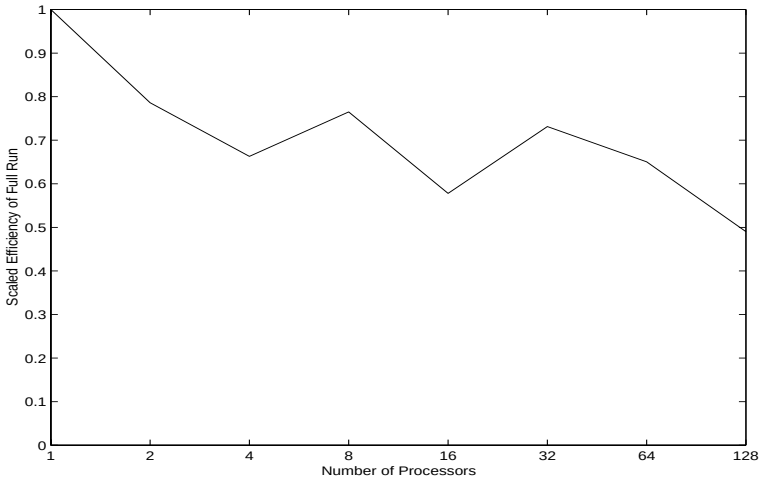


Figure 2: Scaled efficiency of the full run for 50^3 spatial cells on each processor.

5 Conclusions

We have developed a new solution technique for a fully implicit formulation of the multigroup diffusion equations. This technique is particularly appropriate for large-scale, parallel problems in which algorithmic and implementation scalability are essential. Numerical results show that our method is accurate in both space and frequency discretization.

On a one-dimensional test problem, the solution technique showed both algorithmic and implementation scalability. On a three-dimensional test problem, the method showed reasonable parallel scalability for the linear iterations and showed degraded parallel scalability for the full run. We are currently investigating scalability issues in the context of nonlinear problems.

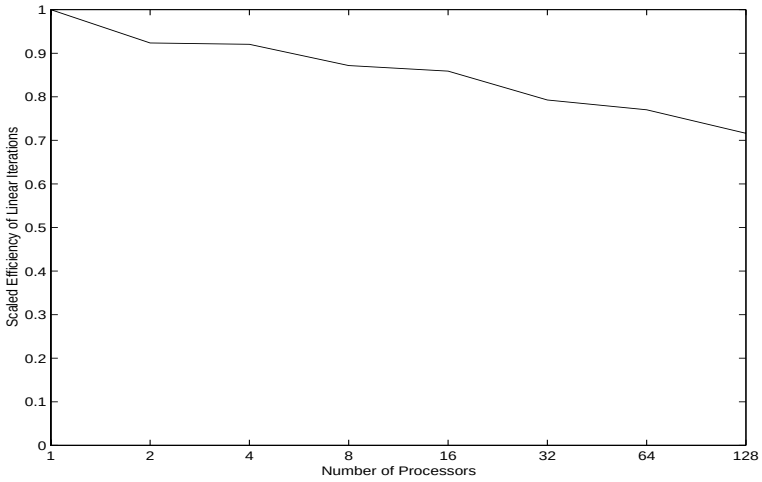


Figure 3: Scaled efficiency of the linear iterations for 50^3 spatial cells on each processor.

In future work, we plan to compare this solution method to the traditional approaches for solving this problem. We expect significant increases in accuracy and computation speed by using this method over the more common time-lagging and operator-splitting approaches.

Acknowledgments The authors wish the thank Alan C. Hindmarsh for his helpful comments throughout the development of the code and the writing of this paper. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.

References

[1] R. L. Bowers and J. R. Wilson, *Numerical Modeling in Applied Physics and Astrophysics*, Jones and Bartlett, Boston, 1991.

[2] P. N. Brown, *A local convergence theory for combined inexact-Newton/finite difference projection methods*, SIAM J. Numer. Anal., 24 (1987), pp. 407–434.

[3] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, *VODE: A variable-coefficient ODE solver*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 1038–1051.

[4] P. N. Brown, R. D. Falgout, and J. E. Jones, *Semicoarsening multigrid on distributed memory machines*. Submitted to the SIAM Journal on Scientific Computing special issue on the Fifth Copper Mountain Conference on Iterative Methods. Also available as LLNL technical report UCRL-JC-130720, 1998.

[5] P. N. Brown and A. C. Hindmarsh, *Matrix-free methods for stiff systems of ODE's*, SIAM J. Num. Anal., 23 (1986), pp. 610–638.

[6] ——, *Reduced storage matrix methods in stiff ODE systems*, J. of Appl. Math. and Comp., 31 (1989), pp. 40–91.

[7] G. D. Byrne and A. C. Hindmarsh, *PVODE, an ODE solver for parallel computers*, Tech. Rep. UCRL-JC-132361, Lawrence Livermore National Laboratory, 1998. Submitted.

[8] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Sci. Statist. Comput., 19 (1982), pp. 400–408.

[9] K. R. Jackson and R. Sacks-Davis, *An alternative implementation of variable step-size multistep formulas for stiff ODEs*, ACM Trans. Math. Software, 6 (1980), pp. 295–318.

[10] D. A. Knoll, W. J. Rider, and G. L. Olson, *An efficient nonlinear solution method for nonequilibrium radiation diffusion*, Tech. Rep. LA-UR-98-2154, Los Alamos National Laboratory, 1998. Submitted to J. Quant. Spec. and Rad. Trans.

[11] W. J. Rider, D. A. Knoll, and G. L. Olson, *A multigrid Newton-Krylov method for multimaterial radiation diffusion*, Tech. Rep. LA-UR-98-2153, Los Alamos National Laboratory, 1998. Submitted to J. Comp. Phys.

[12] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[13] S. Schaffer, *A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients*, SIAM J. on Sci. Comp., 20 (1998), pp. 228–242.

[14] B. Su and G. L. Olson, *Benchmark results for the non-equilibrium Marshak diffusion problem*, J. Quant. Spec. and Rad. Trans., 56 (1996), pp. 337–351.

Peter N. Brown
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-561
Livermore, CA 94551 USA
brown42@llnl.gov

Frank Graziani
Low Energy Density Physics
Lawrence Livermore National Laboratory
P.O. Box 808, L-098
Livermore, CA 94551 USA
graziani1@llnl.gov

Britton Chang
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-561
Livermore, CA 94551 USA
chang1@llnl.gov

Carol S. Woodward
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-561
Livermore, CA 94551 USA
cswoodward@llnl.gov