ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 184 (2003) 53–78

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

Radiation diffusion for multi-fluid Eulerian hydrodynamics with adaptive mesh refinement[☆]

Louis H. Howell[*], Jeffrey A. Greenough

Lawrence Livermore National Laboratory, MS L-560, Livermore, CA 94550, USA

Received 30 October 2001; received in revised form 4 September 2002; accepted 26 September 2002

Abstract

Block-structured meshes provide the ability to concentrate grid points and computational effort in interesting regions of a flow field, without sacrificing the efficiency and low memory requirements of a regular grid. We describe an algorithm for simulating radiation diffusion on such a mesh, coupled to multi-fluid gasdynamics. Conservation laws are enforced by using locally conservative difference schemes along with explicit synchronization operations between different levels of refinement. In unsteady calculations each refinement level is advanced at its own optimal timestep. Particular attention is given to the appropriate coupling between the fluid energy and the radiation field, the behavior of the discretization at sharp interfaces, and the form of synchronization between levels required for energy conservation in the diffusion process. Two- and three-dimensional examples are presented, including parallel calculations performed on an IBM SP-2.

Keywords: Radiation transport; Radiation diffusion; Multi-fluid; Adaptive mesh; Parallel

1. Introduction

The block-structured adaptive mesh refinement (AMR) algorithm, originally developed by Berger and Oliger [6], was shown to be effective for 2D gasdynamics by Berger and Colella [5] and was then extended to 3D by Bell et al. [4]. Coupling the explicit integrator for hyperbolic equations with an implicit multigrid solver permits the solution of systems which also involve parabolic and elliptic effects. For example, an algorithm for the incompressible Navier–Stokes equations, presented as a prototype by Howell and Bell [12] and as a fully developed code by Almgren et al. [1], uses multigrid solvers to implicitly model the elliptic incompressibility constraint and the parabolic terms for diffusion. Other examples of implicit processes

[☆] This work was performed under the auspices of the US Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

[*] Corresponding author. Tel.: 1-925-422-6105; fax: 1-925-422-6287.
E-mail addresses: nazgul@llnl.gov (L.H. Howell), greenough1@llnl.gov (J.A. Greenough).

0021-9991/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0021-9991(02)00015-3

coupled to explicit advection in AMR codes include self-gravitation in astrophysics [23], and discrete ordinates for radiation in combustion applications [14].

In this paper we will present an Eulerian AMR implementation of single-group radiation diffusion, coupled to a Godunov integrator for explicit multi-fluid hydrodynamics. Some aspects of the radiation discretization, particularly with regard to the treatment of sharp interfaces in material properties, deserve close attention. Radiative fluxes across these interfaces have an immediate, global effect on the solution, and the adaptive mesh algorithm gives superior results when such effects are computed accurately on coarse as well as fine grids. Our treatment of these fluxes is, so far as we are aware, original. While it is not our primary intention to break new ground with respect to these uniform-grid aspects of the scheme, they are crucial to understanding the overall performance of the adaptive mesh algorithm.

Once the uniform grid algorithm is established, our focus will shift to examining the implications of the adaptive grid structure and the means of ensuring overall energy conservation while advancing different grid levels at different timesteps. The presence of multiple grids at each refinement level provides a convenient approach to parallelism, which we have exploited along the lines of [20] and by use of the *hypre* parallel linear solver package [9], in particular the semicoarsening multigrid solvers described in [3,8,10,22]. We will not discuss the hydrodynamics algorithm itself, the criteria for grid refinement, or the methods used to allocate refined grids and distribute them across processors, since this paper introduces no new material relating to any of these topics.

An earlier treatment of the present work, prior to the development of parallel and 3D capabilities, appeared in [13].

2. AMR essentials

A complete description of the AMR timestepping scheme can be found in many of the references cited in the previous section, but for this paper we will confine ourselves to a very brief overview. The computational mesh consists of a hierarchy of refinement levels, each of which is locally uniform and organized as a union of rectangular patches. Cells on different levels are aligned, and refinement ratios between levels are typically 2 or 4. Edges of different levels may not be coincident except at the physical boundary—that is, level $l+2$ may not directly border level l without an intervening region at refinement level $l+1$. For computational convenience coarse cells are defined under finer grids, but the finest solution in a given region takes precedence over coarser versions.

The scheme typically refines in time as well as space, both for efficiency and to help maintain an optimal Courant condition on all levels. The refinement ratios r^l in time are the same as those in space—these are specified by the user along with the specific criteria for refinement. Timestepping is a recursive process, where a level l is advanced first without reference to any finer levels, then level $l+1$ is advanced r^{l+1} times to catch up (this in turn requires advancing levels $l+2$ and higher). After the finer levels have reached the time corresponding to the end of the coarse timestep, synchronization operations must be performed to correct any flux mismatches that have occurred at the coarse–fine interface. For hyperbolic terms in the equations this is just a local correction, called refluxing [5]. For elliptic and parabolic equations a separate call to the implicit solver is required.

Fig. 1 shows a possible spatial layout for a grid hierarchy with two refined levels, and Fig. 2 shows a space–time outline of the temporal refinement strategy. The latter figure shows the refinement ratios (4 and 2 in this case), and features a regridding event where the boundary between levels 1 and 2 was moved. Note that regridding of finer levels can occur even in the middle of a coarse level timestep.

Our parallel programming model is a natural development of the adaptive mesh data structure. Grids at each level are distributed across the entire processor set in an attempt to balance the load as evenly as possible. (Software support for this model is provided both by the BoxLib class library [20] and the *hypre*

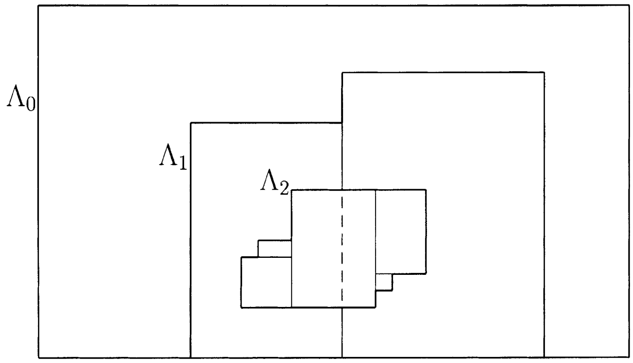

Fig. 1. A properly nested hierarchy of grids.

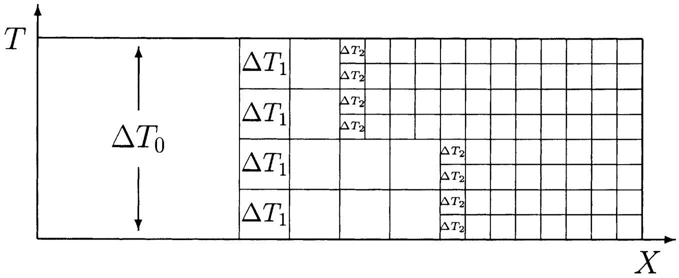

Fig. 2. Fine levels advance at smaller timesteps than coarse levels.

linear solver package [9].) Grids at different refinement levels are not active at the same time, so they are not separated onto different processors. MPI-based communications are required between adjacent grids of the same refinement level, and between grids on different levels that occupy the same part of the domain.

3. Radiation diffusion

Radiative transport is modeled using the diffusion approximation, which satisfies the following equation:

$$\frac{\partial E_{\mathrm{R}}}{\partial t} = \nabla \cdot \left(\frac{c\lambda(E_{\mathrm{R}})}{\kappa_{\mathrm{R}}} \nabla E_{\mathrm{R}} \right) + \kappa_{\mathrm{P}}\left(4\sigma T^4 - cE_{\mathrm{R}}\right), \tag{1}$$

where E_{R} is the radiation energy density and κ_{P} and κ_{R} are the Planck and Rosseland mean absorption coefficients, respectively. The diffusion coefficient $c\lambda(E_{\mathrm{R}})/\kappa_{\mathrm{R}}$ contains a nonlinear (E_{R}-dependent) limiter as described by Levermore and Pomraning [17], the details of which are not a concern here. Note only that $\lambda \leqslant 1/3$, where $1/3$ is the value without limiting.

We have neglected terms relating to radiation pressure and to motion of the fluid in the Eulerian frame. These effects may be included in future versions of the algorithm without major changes to the solution procedure. The $\partial E_{\mathrm{R}}/\partial t$ term is retained because without it the radiation field can transport but cannot contain energy. At high temperatures the energy content of the radiation field may be an important part of the total energy in the system.

The solution of (1) is the radiation energy density, defined as

$$E_{\mathrm{R}} = \frac{1}{c}\int_0^{\infty}\int^{4\pi} I_{\nu}\,\mathrm{d}\Omega\,\mathrm{d}\nu. \tag{2}$$

This is the zeroth moment of the radiative intensity I_{ν}, integrated over all frequencies. We will also have use for the first moment, the radiative flux vector

$$\mathbf{F} = \int_0^{\infty}\int^{4\pi} \Omega I_{\nu}\,\mathrm{d}\Omega\,\mathrm{d}\nu. \tag{3}$$

The diffusion approximation assumes that I_{ν} is weakly dependent on angle and that therefore $\mathbf{F}$ can be represented as

$$\mathbf{F} = -\frac{c\lambda(E_{\mathrm{R}})}{\kappa_{\mathrm{R}}}\nabla E_{\mathrm{R}}. \tag{4}$$

This implies that E_{R} is slowly varying. There is no implication, however, that the radiation field and the fluid are locally coupled.

If they are coupled, so that a common temperature applies to both the fluid and the radiation, we can replace ∇E_{R} with $\nabla(4\sigma T^4/c)$ in (4), obtaining

$$\mathbf{F} = -\frac{16\sigma T^3}{3\kappa_{\mathrm{R}}}\nabla T. \tag{5}$$

This is the radiation heat conduction approximation. We will not use this form of the equation directly, but the rapid, diffusive transport of fluid energy by the radiation field has implications for the design of the time-stepping scheme. In particular, it motivates the need for implicit coupling of the new-time radiation field with the temperature. We also note—without going into the details—that the Rosseland mean coefficient κ_{R} and the Planck mean κ_{P} are defined by different frequency averages appropriate to their differing roles in this approximation.

At vacuum boundaries we use the Marshak boundary condition

$$\hat{n}\cdot\mathbf{F} = \frac{c}{2}E_{\mathrm{R}} - 2F_{\mathrm{inc}}, \tag{6}$$

where F_{inc} is a specified incoming flux. This is a mixed boundary condition involving both E_{R} and its normal derivative. Dirichlet, Neumann and periodic boundary conditions on the radiation field are also supported.

4. Energy update

The radiation field cannot be considered in isolation, as it continually exchanges energy with the fluid medium via the last two terms of (1). The fluid itself obeys the compressible Euler equations with corresponding energy source terms. We use a second-order Godunov method to explicitly advance the density, momentum, and energy of the fluid at each timestep. In addition, we maintain a multi-fluid representation of the state in order to prevent numerical mixing of different materials. The effect of this modification is that the volume fraction f^{α}, density fraction $f^{\alpha}\rho^{\alpha}$, and total energy fraction $f^{\alpha}\rho^{\alpha}E^{\alpha}$ for each fluid α become additional advected quantities. For a description of the multi-fluid algorithm see [11] and [19].

We will start with the single-fluid form of the energy update, and for simplicity of notation will use this form for many of the derivations that follow. Most of the effects of the multi-fluid representation on the radiation algorithm are to require various sums and averages over the fluids in each cell, which must be specified for completeness but add little to the exposition.

The single-fluid energy equation with contributions from advection and radiant energy takes the form

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\boldsymbol{u}\rho E + \boldsymbol{u}p) = -\kappa_{\mathrm{P}}\big(4\sigma T^4 - cE_{\mathrm{R}}\big). \tag{7}$$

Since our concern is primarily with the radiation terms on the right hand side of this equation, we define $(\rho E)^-$ as the fluid energy density updated to time $n+1$ for the advection process, but not for radiation. That is,

$$(\rho E)^- = (\rho E)^n - \Delta t[\nabla \cdot (\boldsymbol{u}\rho E + \boldsymbol{u}p)]^{n+(1/2)}. \tag{8}$$

In each timestep this quantity is computed by the Godunov module before the radiation calculation takes place.

In a single-grid code we could evaluate the radiation terms at time $n + (1/2)$ in order to do a time-centered update. For AMR this is not appropriate, since multilevel solves should be performed at integral times for which both coarse and fine level data exist. The time-centered form of the radiation portion should therefore be

$$-\frac{1}{2}\left[\kappa_{\mathrm{P}}^n\Big(4\sigma(T^n)^4 - cE_{\mathrm{R}}^n\Big) + \kappa_{\mathrm{P}}^{n+1}\Big(4\sigma(T^{n+1})^4 - cE_{\mathrm{R}}^{n+1}\Big)\right]. \tag{9}$$

The time n quantities are known, so we are left with the problem of computing T^{n+1} and E_{R}^{n+1}.

The following sections develop the radiation update process in detail. While we present this material in a partially time-centered, second-order form, in practice we often run our code in a fully implicit, backwards-Euler mode which is first-order in time. This is done for stability reasons, due to the stiffness of the radiation equation and the fact that we often run strongly driven problems featuring sharp interfaces in material properties. Our goal has been to preserve the ability to run a time-centered discretization in the low temperature limit, analogous to the treatment of both thermal diffusion and radiation transport in [14].

5. Implicit radiation update

In some regions of the flow field the fluid and radiation temperatures may be tightly coupled. It is therefore inappropriate to advance T and E_{R} separately—instead, we need an implicit form of the update which takes the coupling of these quantities into account where it exists.

Let us first define the integrated blackbody intensity $B = 4\sigma T^4$. The update for the fluid energy then takes the form

$$(\rho E)^{n+1} = (\rho E)^- - \Delta t\big\{(1-\theta)\kappa_{\mathrm{P}}^n\big(B^n - cE_{\mathrm{R}}^n\big) + \theta\kappa_{\mathrm{P}}^{n+1}\big(B^{n+1} - cE_{\mathrm{R}}^{n+1}\big)\big\}, \tag{10}$$

where we can set $\theta = 1/2$ for a time-centered update, or $\theta = 1$ for backwards-Euler.

Time centering, with the goal of second-order accuracy in time, seems appropriate for the fluid state updates so long as we can perform them without stability problems. In the radiation diffusion equation, though, the time difference is small (stiff) compared to the other terms by a factor of $1/c$. It therefore seems advisable to difference this term in backwards-Euler fashion. The equation to be satisfied is

$$\frac{E_{\mathrm{R}}^{n+1} - E_{\mathrm{R}}^n}{\Delta t} = \nabla \cdot \left(\frac{c\lambda(E_{\mathrm{R}}^{n+1})}{\kappa_{\mathrm{R}}^{n+1}}\nabla E_{\mathrm{R}}^{n+1}\right) + \kappa_{\mathrm{P}}^{n+1}\big(B^{n+1} - cE_{\mathrm{R}}^{n+1}\big). \tag{11}$$

In the limit of large κ, this formulation has the desirable property of forcing emission and absorption—which are then both large—to balance each other. Their difference must be small enough to be comparable to the other terms in the equation. When the $\partial E_{\mathrm{R}}/\partial t$ term is negligible the only remaining term is the flux, so the only possible solution is a quasi-steady balance among emission, absorption and flux at a single point in time.

There is a problem, however. The quantities $\kappa_{\mathrm{P}}(B - cE_{\mathrm{R}})$ represent transfer of energy from the fluid to the radiation field. If we include these in the fluid update in Crank–Nicolson form, but in the radiation update in backwards-Euler form, the two will not balance and the system will not conserve energy. If we difference both in Crank–Nicolson form, then the radiation update (11) becomes

$$\frac{1}{2}v^{n+1} + \frac{1}{2}v^{n} = 0, \tag{12}$$

where $v = \kappa_{\mathrm{P}}(B - cE_{\mathrm{R}}) - \nabla \cdot \mathbf{F}$ and the time derivative is omitted. This supports an oscillatory solution $v^n = a(-1)^n$. A numerical imbalance among emission, absorption and flux at time n is merely inverted, rather than corrected, at time $n + 1$. In the limit of large κ we can have emission and absorption far out of sync, a nonphysical situation.

The alternative we have chosen is to modify the fluid energy update to be of the form

$$(\rho E)^{n+1} = (\rho E)^{-} - \Delta t\{(1-\theta)(\nabla \cdot \mathbf{F}^{n} - \nabla \cdot \mathbf{F}^{n+1}) + \kappa_{\mathrm{P}}^{n+1}(B^{n+1} - cE_{\mathrm{R}}^{n+1})\}. \tag{13}$$

Since the emission and absorption appear as in a backwards-Euler difference, and the flux terms do not affect conservation on a uniform grid, the combined radiation and fluid energy update is conservative. On the other hand, this form reduces to the true Crank–Nicolson form (10) when $\partial E_{\mathrm{R}}/\partial t$ is small. With the radiation equation permanently in backwards-Euler form this is the only case in which we would expect to see second-order accuracy. (The Crank–Nicolson discretization for diffusion is itself vulnerable to oscillations when Δt is sufficiently large. These are not as serious, though, as the problems we would expect from time-centering (11).)

6. Multi-fluid modifications, average coefficients

In the multi-fluid form of the algorithm, each cell may potentially be occupied by more than one material. Each fluid α in a cell takes up a volume fraction f^{α}, with $\sum_{\alpha} f^{\alpha} = 1$. Most quantities derived from the fluid state, including ρ, T and κ_{P}, are derived separately for each fluid. The radiation energy density E_{R}, though, is given a single value over the entire cell. There is thus a separate update equation for each material energy, but a single equation for the radiation:

$$(f^{\alpha}\rho^{\alpha}E^{\alpha})^{n+1} = (f^{\alpha}\rho^{\alpha}E^{\alpha})^{-} - \Delta t\left\{(1-\theta)\frac{f^{\alpha}\kappa_{\mathrm{P}}^{\alpha,n+1}}{\kappa_{\mathrm{P}}^{n+1}}(\nabla \cdot \mathbf{F}^{n} - \nabla \cdot \mathbf{F}^{n+1}) + f^{\alpha}\kappa_{\mathrm{P}}^{\alpha,n+1}(B^{\alpha,n+1} - cE_{\mathrm{R}}^{n+1})\right\}, \tag{14}$$

$$\frac{E_{\mathrm{R}}^{n+1} - E_{\mathrm{R}}^{n}}{\Delta t} = \nabla \cdot \left(\frac{c\lambda(E_{\mathrm{R}}^{n+1})}{\kappa_{\mathrm{R}}^{n+1}}\nabla E_{\mathrm{R}}^{n+1}\right) + \sum_{\alpha} f^{\alpha}\kappa_{\mathrm{P}}^{\alpha,n+1}(B^{\alpha,n+1} - cE_{\mathrm{R}}^{n+1}). \tag{15}$$

Note that summing the equations (14) over all fluids recovers the single-fluid form of the energy update (13). The average Planck mean κ_{P} appearing in (14) is simply $\sum_{\alpha} f^{\alpha}\kappa_{\mathrm{P}}^{\alpha}$, and is chosen to make this sum come out correctly.

The average Rosseland mean κ_{R} in the diffusion coefficient is not so trivial to define, however. We have experimented with a number of different formulas with various motivations, and have achieved best results with a mass-weighted arithmetic average as shown in Fig. 3.

There is another context in which we must average κ_{R}, not because of the multi-fluid representation but because of the difference stencil for diffusion. The Rosseland mean appears in the diffusion coefficient, which falls on cell edges rather than cell centers in the standard 5-point discretization. The need to average values onto edges gives us an opportunity to address one of the awkward cases in the diffusion model—that

$$\kappa_R = \frac{\sum_\alpha f^\alpha \rho^\alpha \kappa_R^\alpha}{\sum_\alpha f^\alpha \rho^\alpha}$$

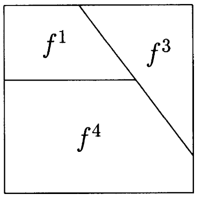

Fig. 3. The Rosseland mean must be averaged over the various fluids in each cell.

$$\kappa_R = \min\left[\frac{\kappa_R^l + \kappa_R^r}{2}, \max\left(\frac{2\kappa_R^l \kappa_R^r}{\kappa_R^l + \kappa_R^r}, \frac{4}{3\Delta x}\right)\right]$$

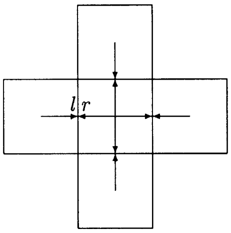

Fig. 4. The difference stencil for diffusion uses coefficients averaged to edges. The value $4/(3\Delta x)$ gives a reasonable flux across an interface where optically thick and optically thin materials meet.

of getting the correct flux at a sharp interface between optically thick and optically thin materials. In the formula shown in Fig. 4 the value $4/(3\Delta x)$ is selected for this configuration, producing a reasonable radiative flux (σT^4) across an interface. Because it is not tied to the multi-fluid representation, this surface flux formula applies even at interfaces—thermal waves, for example—where no material boundary is present.

7. Energy/radiation update loop

Solution of Eq. (15) requires several quantities not yet available, most importantly the integrated blackbody intensity $B^{\alpha,n+1} = 4\sigma(T^{\alpha,n+1})^4$. We can estimate this based on a temperature T^α from the partially updated fluid energy $(f^\alpha \rho^\alpha E^\alpha)^-$, perhaps with some explicit guesswork about the expected effect of the radiation field, but whatever we do the estimate will not match the updated energy to be obtained via equation (14). In general we will have to obtain $T^{\alpha,n+1}$ through an iterative process.

Linearization of B and other quantities about the temperature T^α which is the current best estimate to $T^{\alpha,n+1}$ gives us an iteration in the form of a modified Newton's method. For independent variables we use E_R and the fluid internal energies $Q_\alpha = f^\alpha \rho^\alpha e^\alpha$. (The kinetic portion of $f\rho E$ is the same on both sides of (14) and is not altered by the radiation module, so we can subtract it off and work with internal energies only.) Since $(f^\alpha \rho^\alpha)^{n+1}$ is known, from each Q_α we can find a current internal energy e^α, and then obtain T, c_v and other physical quantities as functions of ρ and e from tabular physical data.

The coupled equations are written as dependent variables that should all be zero when the system is converged:

$$F_\alpha = (f^\alpha \rho^\alpha e^\alpha)^{n+1} - (f^\alpha \rho^\alpha e^\alpha)^- + \Delta t\left\{(1-\theta)\frac{f^\alpha \kappa_\mathrm{P}^{\alpha,n+1}}{\kappa_\mathrm{P}^{n+1}}(\nabla\cdot\mathbf{F}^n - \nabla\cdot\mathbf{F}^{n+1}) + f^\alpha \kappa_\mathrm{P}^{\alpha,n+1}\left(B^{\alpha,n+1} - cE_\mathrm{R}^{n+1}\right)\right\}, \tag{16}$$

$$F_\mathrm{R} = E_\mathrm{R}^{n+1} - E_\mathrm{R}^n - \Delta t\nabla\cdot(D\nabla E_\mathrm{R}^{n+1}) - \Delta t\sum_\alpha f^\alpha \kappa_\mathrm{P}^\alpha\left(B^{\alpha,n+1} - cE_\mathrm{R}^{n+1}\right). \tag{17}$$

Partial derivatives of these dependent variables form the Jacobian matrix. We then have a linear system to solve for updates to the radiation and fluid energies. Note that for each cell there are multiple fluid energies Q_α and multiple fluid energy equations represented by the F_α, one for each fluid:

$$\begin{bmatrix} \frac{\partial F_\alpha}{\partial Q_\alpha} & \frac{\partial F_\alpha}{\partial E_\mathrm{R}} \\ \frac{\partial F_\mathrm{R}}{\partial Q_\alpha} & \frac{\partial F_\mathrm{R}}{\partial E_\mathrm{R}} \end{bmatrix} \begin{bmatrix} \delta Q_\alpha \\ \delta E_\mathrm{R} \end{bmatrix} = \begin{bmatrix} -F_\alpha \\ -F_\mathrm{R} \end{bmatrix}. \tag{18}$$

A few simplifications are helpful when forming the partial derivatives in (18). While some quantities, such as $f^{\alpha,n+1}$ and $(f^\alpha \rho^\alpha)^{n+1}$, are truly fixed, several others depend on some or all of the independent variables. We neglect the dependence of the diffusion coefficient both on fluid temperature (through κ_R) and on radiation energy (through the flux limiter). We also ignore the dependence of the $(1-\theta)$ correction term on temperature through κ_P. We do, however, include the temperature dependence of emission and absorption.

Dropping the $n+1$ superscripts, we have

$$\frac{\partial F_\alpha}{\partial Q_\alpha} = 1 + \frac{\Delta t}{f^\alpha \rho^\alpha c_v^\alpha} \left\{ f^\alpha \kappa_\mathrm{P}^\alpha \frac{\partial B^\alpha}{\partial T^\alpha} + f^\alpha (B^\alpha - cE_\mathrm{R}) \frac{\partial \kappa_\mathrm{P}^\alpha}{\partial T^\alpha} \right\}, \tag{19}$$

$$\frac{\partial F_\alpha}{\partial E_\mathrm{R}} = \Delta t \left\{ (1-\theta) \frac{f^\alpha \kappa_\mathrm{P}^\alpha}{\kappa_\mathrm{P}} \nabla \cdot (D\nabla) - f^\alpha \kappa_\mathrm{P}^\alpha c \right\}, \tag{20}$$

$$\frac{\partial F_\mathrm{R}}{\partial Q_\alpha} = -\frac{\Delta t}{f^\alpha \rho^\alpha c_v^\alpha} \left\{ f^\alpha \kappa_\mathrm{P}^\alpha \frac{\partial B^\alpha}{\partial T^\alpha} + f^\alpha (B^\alpha - cE_\mathrm{R}) \frac{\partial \kappa_\mathrm{P}^\alpha}{\partial T^\alpha} \right\}, \tag{21}$$

$$\frac{\partial F_\mathrm{R}}{\partial E_\mathrm{R}} = 1 - \Delta t \nabla \cdot (D\nabla) + \Delta t \kappa_\mathrm{P} c. \tag{22}$$

The upper-left block of the Jacobian matrix is diagonal—as a consequence of the simplifications given above—and therefore trivial to invert. We can therefore solve the system by forming the Schur complement:

$$\begin{bmatrix} \frac{\partial F_\alpha}{\partial Q_\alpha} & \frac{\partial F_\alpha}{\partial E_\mathrm{R}} \\ 0 & \frac{\partial F_\mathrm{R}}{\partial E_\mathrm{R}} - \frac{\partial F_\mathrm{R}}{\partial Q_\alpha} \left(\frac{\partial F_\alpha}{\partial Q_\alpha} \right)^{-1} \frac{\partial F_\alpha}{\partial E_\mathrm{R}} \end{bmatrix} \begin{bmatrix} \delta Q_\alpha \\ \delta E_\mathrm{R} \end{bmatrix} = \begin{bmatrix} -F_\alpha \\ -F_\mathrm{R} + \frac{\partial F_\mathrm{R}}{\partial Q_\alpha} \left(\frac{\partial F_\alpha}{\partial Q_\alpha} \right)^{-1} F_\alpha \end{bmatrix}. \tag{23}$$

The quantities

$$\eta^\alpha = -\frac{\partial F_\mathrm{R}}{\partial Q_\alpha} \left(\frac{\partial F_\alpha}{\partial Q_\alpha} \right)^{-1} \tag{24}$$

appearing on both sides of this system are local to each cell, so the equation for δE_R is still a diffusion equation. It is no more difficult to solve than (15) would be if the fluid temperatures were known in advance. The Newton iteration thus requires one call to the implicit diffusion solver per iteration.

We normally iterate to convergence, using a criterion based on the pointwise fluid energy change $\sum_\alpha |\delta f^\alpha \rho^\alpha e^\alpha|$. Quantities such as the κ's and the flux limiter may be updated at each iteration even if they are not included in the linearization (19)–(22). These updates are optional in our implementation, because though they may improve the quality of the solution, they often make the iteration converge more slowly, and because repeated calls to the data tables may be expensive. We stress that the scheme is designed to conserve energy whether these quantities are updated or lagged, and whether or not the nonlinear system itself is iterated to convergence. Conservation depends only on the accurate solution of the diffusion equation at the final step of the iteration.

If we wanted to make the linearization more complete, probably the most important term to add would be temperature dependence of the diffusion coefficient through κ_{R}. The presence of this term inside the divergence operator would greatly complicate the linear system to be solved at each iteration, however. Linearizing the dependence of the limiter on E_{R} would lead to similar complications. Probably the easiest term to add would be the κ_{P} dependence in the $(1-\theta)$ correction term. Even this would couple the different fluids within each cell, though, and thus make $\partial F_\alpha/\partial Q_\alpha$ harder to invert. It probably wouldn't be worth the trouble to do this since the term is only nonzero in cells with more than one fluid present.

8. Stability of the update

There are two separate questions we may ask about the update iteration: "Does the iteration itself converge?" and "Given convergence at each timestep, what behavior do we expect over multiple timesteps?" To address the first, we note that since the iteration is essentially Newton's method, we can expect the usual behavior of Newton iterations. The method should converge rapidly near the solution if the Jacobian approximation is reasonable, but may perform poorly if the terms we have omitted are important. If started far from the solution, in the sense that nonlinearities become dominant, the method may diverge. This suggests a timestep limitation based on a maximum permissible change in temperature.

Most of the complexity of the linearization (19)–(22) involves the transfer of energy between the fluid components and the radiation. To simplify the analysis of the algorithm, we first consider only the total energy in each cell. Letting $F_\Sigma = \sum_\alpha F_\alpha$, we have

$$\sum_\alpha \left(\frac{\partial F_\Sigma}{\partial Q_\alpha} + \frac{\partial F_{\mathrm{R}}}{\partial Q_\alpha}\right)\delta Q_\alpha + \left(\frac{\partial F_\Sigma}{\partial E_{\mathrm{R}}} + \frac{\partial F_{\mathrm{R}}}{\partial E_{\mathrm{R}}}\right)\delta E_{\mathrm{R}} = -F_\Sigma - F_{\mathrm{R}}. \tag{25}$$

The simplification then results from the facts that $\partial F_\Sigma/\partial Q_\alpha + \partial F_{\mathrm{R}}/\partial Q_\alpha = 1$, and that the emission and absorption terms on the right hand side also cancel.

To write the update (25) out in full, we need to make clearer the iterative nature of the scheme. We are updating quantities to time $n+1$. Let us use a * superscript to indicate the previous estimates to those quantities. We will then have $E_{\mathrm{R}}^{n+1} = E_{\mathrm{R}}^{*} + \delta E_{\mathrm{R}}$ and so on. We iterate the process by substituting these new $n+1$ estimates in place of the older * quantities and computing the update again.

The total energy update can then be written as

$$\begin{aligned}\sum_\alpha \delta Q_\alpha + \delta E_{\mathrm{R}} - \theta\Delta t\nabla\cdot(D^*\nabla\delta E_{\mathrm{R}}) = {}& -\sum_\alpha Q_\alpha^* + \sum_\alpha Q_\alpha^- - E_{\mathrm{R}}^* + E_{\mathrm{R}}^n \\ & + \Delta t\big[(1-\theta)\nabla\cdot(D^n\nabla E_{\mathrm{R}}^n) + \theta\nabla\cdot(D^*\nabla E_{\mathrm{R}}^*)\big],\end{aligned} \tag{26}$$

which simplifies to

$$\sum_\alpha Q_\alpha^{n+1} + E_{\mathrm{R}}^{n+1} = \sum_\alpha Q_\alpha^- + E_{\mathrm{R}}^n + \Delta t\big[(1-\theta)\nabla\cdot(D^n\nabla E_{\mathrm{R}}^n) + \theta\nabla\cdot(D^*\nabla E_{\mathrm{R}}^{n+1})\big]. \tag{27}$$

Eq. (27) closely resembles a standard diffusion update evaluated at time $n+\theta$, with the nonlinear diffusion coefficient represented by D^*. This is even clearer in limits where separate energies are not important—either the low-temperature limit where $E_{\mathrm{R}} \ll Q_\alpha$, or the close-coupling limit where the fluid and radiation temperatures are equal. The latter case can be written as a simple Crank–Nicolson update for radiation heat conduction:

$$\rho^{n+1}c_v\frac{T^{n+1}-T^-}{\Delta t} = (1-\theta)\nabla\cdot\left(\frac{16\sigma(T^n)^3}{3\kappa_{\mathrm{R}}^n}\nabla T^n\right) + \theta\nabla\cdot\left(\frac{16\sigma(T^*)^3}{3\kappa_{\mathrm{R}}^*}\nabla T^{n+1}\right), \tag{28}$$

where c_v may include separate fluid and radiation contributions. The usual stability considerations for a Crank–Nicolson update then apply. When the timestep is too large this discretization (with $\theta = 1/2$) can be vulnerable to oscillations, but these can be avoided (with some loss of accuracy) by running the scheme in backwards-Euler mode ($\theta = 1$).

The behavior of the total energy seems reasonable, but what can we say about the fluid and radiation separately, with the strongly nonlinear coupling between them? If we take the radiation update by itself (the second row of (23)), collapse the separate fluids into one, and write it as an equation for E_{R}^{n+1}, we obtain

$$\begin{aligned} &E_{\mathrm{R}}^{n+1} + \Delta t(1-\eta^*)\kappa_{\mathrm{P}}^* c E_{\mathrm{R}}^{n+1} - \Delta t[1-(1-\theta)\eta^*]\nabla\cdot(D^*\nabla E_{\mathrm{R}}^{n+1}) \\ &\quad = E_{\mathrm{R}}^n + \Delta t(1-\eta^*)\kappa_{\mathrm{P}}^* B^* - \eta^*(Q^* - Q^-) + \Delta t(1-\theta)\eta^*\nabla\cdot(D^n\nabla E_{\mathrm{R}}^n). \end{aligned} \tag{29}$$

There are two cases to consider. Normally, increasing fluid temperature increases the transfer of energy from the fluid to the radiation field. In this case the linearization parameter η^* is positive, varying from 0 for small Δt up to 1 for large Δt. Equation (29) is then a well-behaved diffusion equation. The linear system for E_{R}^{n+1} is symmetric, positive definite, diagonally dominant, and solvable by standard software packages. The effect of the linearization on this linear system is simply to damp the absorption term. In the limit of large Δt equation (29) reduces to

$$-\theta\nabla\cdot(D^*\nabla E_{\mathrm{R}}^{n+1}) = (1-\theta)\nabla\cdot(D^n\nabla E_{\mathrm{R}}^n). \tag{30}$$

Barring nonlinear effects of D^*, this is stable for $\theta = 1$ and vulnerable to oscillations for $\theta = 1/2$, as we would expect.

More troubling is the unusual case of positive feedback, where increasing fluid temperature increases absorption more than emission. The parameter η^* can then be negative, and the system is stable only for small Δt. With increasing timestep η^* becomes singular. The simplest course of action is to inhibit linearization and arbitrarily set η^* to zero when this inverted condition is detected.

9. Single-grid tests

Before describing the adaptive mesh refinement algorithm, we present two numerical examples to illustrate features of the discretization. Both are pure radiation problems with no hydrodynamics. The purpose of both is to show how the method handles sharp interfaces between regions with differing material properties.

The first test is an idealized problem showing the effect of the surface flux formula (Fig. 4) for averaging κ_{R} onto cell edges. The domain is the unit square with insulating boundary conditions. This region is divided into two different materials, one optically thick ($\kappa_{\mathrm{P}} = \kappa_{\mathrm{R}} = 10^6$) and one optically thin ($\kappa_{\mathrm{P}} = 0$, $\kappa_{\mathrm{R}} = 10^{-4}$). The thick material forms a quarter-circle with radius 0.5, centered at one corner of the domain. Cells along the interface between the two fluids are multi-fluid cells with appropriate volume fractions. We set uniform conditions $\rho = 1$, $c_v = 1$, $e = T = 100$, $E_{\mathrm{R}} = 0$ at time 0, the same for both materials. For constants we use $\sigma = 2.5 \times 10^{-9}$, $c = 10^6$.

When the system comes to equilibrium we should have $E_{\mathrm{R}} = (4\sigma T^4)/c = 10^{-6}$ throughout the domain. The thick region reaches this value very rapidly. With an initial $\partial E_{\mathrm{R}}/\partial t = 4\kappa_{\mathrm{P}}\sigma T^4 = 10^6$ it is approaching equilibrium by time 10^{-12}. The thin region, on the other hand, is dependent on radiation emitted at a rate of $\sigma T^4 = 0.25$ from an interface of length $\pi/4$. With an area of $1 - \pi/16$ to fill, the radiation in the thin material is not approaching its final value until time 4×10^{-6}.

We run the simulation until time 10^{-7}. At this point the total radiation energy should be approximately 2.160×10^{-7} (that is, $(\pi/16) \times 10^{-6}$ in the thick region plus $(\pi/16) \times 10^{-7}$ in the thin region). We compare results using three different ways of computing κ_R at edges for use in the diffusion coefficient: an arithmetic average, a harmonic average, and the surface formula, all without the flux limiter, and then the latter two again with the flux limiter [17] turned on.

The limiter is important to consider because $t = 10^{-7}$ is only a tenth of the light travel time across the domain. In a sense the surface formula and the limiter are trying to produce the same thing: a physically reasonable flux into and through a transparent medium. We should therefore be concerned whether the two will interfere with each other and possibly limit the flux too much.

To test the effects of resolution, we run the problem on two different meshes (64×64 and 128×128) and with two different timesteps. With $\Delta t = 10^{-7}$ the code takes a single step, while 10 are required for $\Delta t = 10^{-8}$. Backward-Euler differencing is used, and the limiter is iterated to convergence based on new-time E_R. Results are summarized in Table 1.

The energies computed using the arithmetic average correspond closely to the total area of cells (full and mixed) containing some of the thick material. Almost no radiation makes it into the thin region. The values for the harmonic average without the flux limiter, on the other hand, are near the final equilibrium value for the problem. These two methods drastically under- and over-estimate the flux out of the thick region, respectively. Even with the limiter the harmonic average gives too much flux across the thick–thin interface, but the limiter is partially successful at keeping the radiation from streaming instantly across the domain. The good results are the ones using the surface formula. Far from interfering with each other, the limiter and surface formula together give better performance than any other option tried.

This calculation seems to depend very little on the timestep. Spatial resolution has a much greater impact, probably due to discretization effects at the interface itself. The interface cells are the only ones with significant coupling to the radiation field in the thin region.

The arithmetic average of κ_R over the fluids within each mixed cell (Fig. 3) is also important to the result. We do not want to lengthen this section with more numerical comparisons, but other averaging schemes we tried gave too much weight to the optically thin components, allowing the mixed cells to couple strongly to the thin region and thus overestimating the flux. It is interesting that the approach that gives the best results for averaging within each cell is the one that gives the worst behavior for averaging onto edges. The arithmetic fluid average causes mixed cells to be associated with the thick region, then the edge average controls the coupling between this region and the rest of the domain.

The very short time scale of the preceeding example shows the surface flux formula at its best advantage. The result is correct, but perhaps misleading. We now examine a model problem which develops over a longer time, with temperature-dependent coefficients, where a radiation front burns through a layer of optically thick material.

Table 1
Integrated radiant energy for surface flux test problem (all values $\times 10^{-7}$)

Mesh	64×64		128×128	
Timestep	10^{-7}	10^{-8}	10^{-7}	10^{-8}
Arithmetic average	2.029	2.029	1.997	1.997
Harmonic average	9.980	10.000	9.968	10.000
Harmonic with limiter	2.854	2.852	2.779	2.801
Surface formula	2.270	2.274	2.237	2.241
Surface with limiter	2.205	2.207	2.167	2.167

The correct value is near 2.160.

The problem is one-dimensional, with three material regions and no hydrodynamics. In all three regions we have a constant $c_v = 10^8$, an initially constant $T = 10^4$, variable absorption coefficients $\kappa_P = \kappa_R = K\rho^2T^{-1}$, and no initial radiation field. The material-dependent properties are as follows:

Region	ρ	K
$0.000 < x < 0.005$	1	10^6
$0.005 < x < 0.015$	10	10^{10}
$0.015 < x < 0.020$	1	10^4

Physical constants have the values $c = 3 \times 10^{10}$ and $\sigma = 5.6697 \times 10^{-5}$. We drive the problem by applying a Dirichlet boundary condition on the radiation energy density at $x = 0.02$ corresponding to a temperature of 3×10^6. There is no radiation flux at the other boundary ($x = 0$). The radiation flux limiter is not used.

Fig. 5 shows the evolution of the temperature field at time intervals of 2×10^{-6}, computed with $\Delta x = 0.0002$ and $\Delta t = 10^{-8}$. The thick central material has an optical depth of 10^6 (20000 per cell) when cold, dropping to 3333 (67 per cell) when hot. A sharp front develops until the radiation has burned all the way through this thick material. The purpose of the test is to see which parameters and features of the algorithm are most important to the accuracy of the simulation. Plotting all of the comparisons would consume too much space, but a summary of the results should suffice.

First, the results do not depend strongly on spatial resolution. The example shown, with 50 cells through the thick material, is well-resolved. Finer meshes reproduce the same profiles. If we coarsen the mesh so that only 20 cells are in the thick material, the front moves slightly faster due to imperfect resolution of the steep gradient region. With 10 thick cells the front has begun to break through the interface at $x = 0.005$ by time 6×10^{-6} (corresponding to the middle curve of the figure). The important point is that it is not necessary to resolve an optical depth in any of the materials. It is only necessary to resolve the actual profiles observed in the simulation.

Timestepping is a more critical issue, though not so much the size of the timestep as the details of the timestepping algorithm. The simulation shown in the figure iterated each implicit update to convergence, updating κ_P and κ_R to the new fluid temperature for every iteration. For this mode of operation using $\Delta t = 10^{-8}$ was severe overkill. A timestep of 10^{-7} gives almost identical results, and even $\Delta t =$

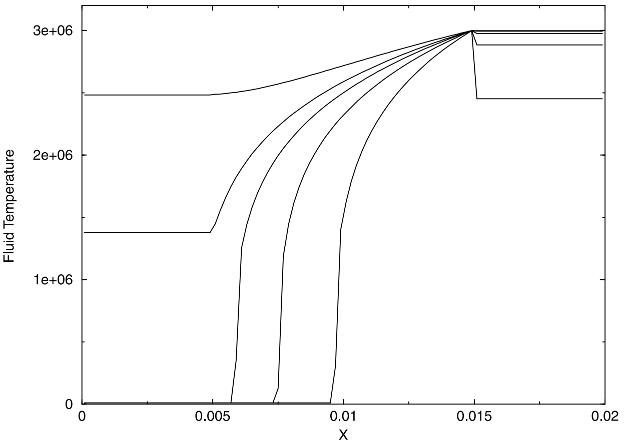

Fig. 5. Thermal wave moving through a layer of optically thick material (from the right): temperature profiles are shown for times 2×10^{-6}, 4×10^{-6}, 6×10^{-6}, 8×10^{-6}, 1×10^{-5}.

10^{-6} (2 timesteps per curve in the figure) only distorts the results by a few percent. If, on the other hand, we iterate the σT^4 factor to convergence but do not update κ_P and κ_R, we need timesteps 10× smaller to achieve a similar level of accuracy. Runs which did not iterate to convergence, but only performed one implicit update iteration per timestep, did not give stable results when Δt was much over 10^{-10}. All of these results were in backward-Euler ($\theta = 1$) mode. Centered-difference timestepping did not work for any reasonable timestep due to oscillations at the right (Dirichlet) boundary.

The difference in results between a harmonic average onto edges for κ_R and the surface formula (Fig. 4) were very minor. There are at least two ways to explain this. One is to observe that the temperature curve through the thick material adjusts itself to give a balanced flux, becoming steeper in colder, more opaque material. The interface at $x = 0.015$, the only point where the surface formula applies, is not a bottleneck. The other explanation is that the light travel time, and the time taken to heat the first thick cell, are both much shorter than the timescale of the problem. Examination of early time behavior shows that the surface formula makes a large difference up to time 10^{-10}. Also, because it helps limit the temperature change in the interface cells, it makes the nonlinear update loop more robust during the first few timesteps.

An arithmetic mean for the average onto edges gave significantly worse performance. It slowed the progress of the front through the thick material, and this effect was more serious on coarser grids. While we recommend the surface formula, if the choice must be made between an arithmetic and a harmonic mean then the harmonic is definitely better.

Additional support for the importance of converging the nonlinearities, and its connection with time step stability issues is presented in [15,16,21].

10. AMR time step

Up to this point we have been discussing operations confined to a single level of refinement. In the AMR timestepping scheme each level advances at its own local timestep, obtaining boundary data from the next coarser level or from the physical boundary as appropriate. The pattern of rectangular grids within a level should make no difference to the computed solution, since a different division of a refined region into patches does not alter the discretization.

The real issue comes at the end of a coarse timestep, when coarse and fine levels have advanced to equivalent points in time. Fluxes across the coarse–fine interface have been computed separately by the two levels and therefore do not match exactly. In the explicit advection scheme this mismatch can be remedied by a local update to the coarse cells adjacent to the interface. In the parabolic radiation update, however, information from a localized region can potentially affect the entire domain, so an additional call to the linear solver is required.

Unlike the equations solved in advancing a level, the synchronization operation directly impacts at least two levels simultaneously. We therefore need a multilevel solver which computes a solution on two or more levels at a time, in the style of [2] or [18]. Fortunately, though, the corrections are generally small, so we require only a single linear solution, not an iterative nonlinear solution as we do for the level advance operations. An extensive discussion of multilevel synchronization for elliptic and parabolic effects is presented in [1] in the context of incompressible fluid flow.

The form of synchronization required for radiative energy conservation in an AMR time step is similar to that developed in [14] for an AMR discrete ordinates code, with the added complication that in the present algorithm the radiation field stores energy and is not merely a transport mechanism. The algorithm in [14] also included correction terms designed to improve the accuracy of the coarse grid solution by anticipating the effects of the finer levels. These corrections appear in the algorithm below as the "multilevel init option". This option is only needed when finer levels have an immediate, global effect on the radiation field in the coarse region of the mesh. Since it costs an additional call to the multilevel solver per coarse timestep, we usually do not use it. Implications of the multilevel init option are discussed for the first numerical test problem in the next section.

The AMR time step is defined recursively in terms of operations on a level l and its interactions with coarser and finer levels. We are advancing level l from time index n to $n+1$, corresponding to time values $t^{\mathrm{old},l}$ and $t^{\mathrm{new},l}$, respectively. (Even though levels other than l have executed different numbers of timesteps, we will use the $n+1$ superscript to refer to values at time $t^{\mathrm{new},l}$ on all levels involved in the calculation.) The region covered by level l is denoted Λ^l, its border is $\partial\Lambda^l$, and the border of the next finer level, projected onto level l, is $\mathbf{P}(\partial\Lambda^{l+1})$.

The notation to describe all of this is unavoidably complex due to the quantities at different times, levels, and stages of the update process. In the following outline, we attempt to specify the update for the level l and its synchronization with finer levels. We use a check (e.g., $\check{E}_{\mathrm{R}}$) to denote new-time values resulting from the level l implicit update, prior to the sync solve, and for the current values on levels $l+1$ and higher. (These higher-level values will include the effects of higher-level sync solves already completed.) Bar values (e.g., $\bar{\mathbf{F}}$) are time averages over the level l step. A tilde (e.g., $\tilde{E}_{\mathrm{R}}$) is for the corrections resulting from the sync solve between levels l and $l+1$, and a hat (e.g., $\hat{E}_{\mathrm{R}}$) is for corrections deriving from the multilevel init option.

Note that while $\nabla\cdot\mathbf{F}$ is needed everywhere, the flux $\mathbf{F}$ itself is only stored on the borders between levels. Our code has data structures called flux registers designed for this purpose. The notation $\langle\cdot\rangle$ indicates an average in space over the fine cells (or cell faces) making up each corresponding coarse cell (or face), while $\langle\langle\cdot\rangle\rangle$ denotes an average in both space and time over the coarse (level l) time step. Thus, at the only point in the algorithm where this notation is used,

$$\langle\langle\bar{\mathbf{F}}^{l+1}\rangle\rangle = \frac{1}{r^{l+1}} \sum_{m=r^{l+1}n}^{r^{l+1}(n+1)-1} \langle\bar{\mathbf{F}}^{l+1,m+\theta}\rangle, \tag{31}$$

where the sum is over the r^{l+1} level $l+1$ timesteps making up the level l timestep from n to $n+1$.

The function D_{R} appearing as a source term in each of the diffusion equations is called the reflux divergence, because it takes the form of the divergence of a flux difference $\delta\mathbf{F}$ stored in the flux registers. This term is evaluated *only* in the coarse cells bordering an interface with a finer level. It does not affect fine cells at the interface because their flux calculations are already correct; the D_{R} term represents the corrected fluxes from these fine cells being imposed onto the coarse grid.

Another way to understand the D_{R} term in the synchronization step is to consider $\delta\mathbf{F}$ to be energy that has been "misplaced" at the coarse–fine interface during the level time steps, due to the differing flux calculations on the different levels. If the solution were not corrected, this energy would be lost and the system would not be conservative. Instead, the D_{R} term re-introduces the missing energy into the system as a radiation source in the coarse cells next to the interface.

We present all of this material in single-fluid form, as the multi-fluid discretization has no impact on the adaptive mesh algorithm. The values of E_{R} and ρE at the end of the synchronization process are the corrected new-time data, on all affected levels, for the end of the level l timestep. It is still possible for these values to change if there are any sync solves pending for levels $l-1$ and below. With this one major exception, we have attempted to write this algorithm without updates or reassignments—each quantity at each point is assigned only once. (The other exceptions to this rule are trivial. Some fluxes are multiply assigned at domain boundaries, for example, but these values are never used since refluxing does not happen at domain boundaries.)

```
**If** $(l < l_{\max})$ and (regrid requested for timestep $n$) **then**
  **For** $l' \in \{l_{\max}-1,\ldots,l\}$ **do**
    • Determine new grid layout for level $l'+1$.
    • Interpolate data to new grids from level $l'$.
    • Copy data on intersection with old level $l'+1$ grids.
  Enddo
Endif
If $(l < l_{\max})$ or $(t^{\mathrm{old},l} = t^{\mathrm{old},l-1})$ then
  • Store $\mathbf{F}^{l,n}$ in flux registers on $\partial\Lambda^l$ and $\mathbf{P}(\partial\Lambda^{l+1})$, compute $\nabla\cdot\mathbf{F}^{l,n}$ on $\Lambda^l$.
Endif
If $(l < l_{\max})$ and $(t^{\mathrm{old},l} \neq t^{\mathrm{old},l-1})$ and (multilevel init option) then
  For $l' \in \{l_{\max},\dots,l\}$ do
    ☆ $(\partial E_{\mathrm{R}}/\partial t)^{l',\mathrm{est}} = -\nabla\cdot\mathbf{F}^{l',n} + \kappa_{\mathrm{P}}^{l',n}\left(B^{l',n} - cE_{\mathrm{R}}^{l',n}\right)$    on $\Lambda^{l'} - \mathbf{P}(\Lambda^{l'+1})$
    ☆ $(\partial E_{\mathrm{R}}/\partial t)^{l',\mathrm{est}} = \langle(\partial E_{\mathrm{R}}/\partial t)^{l'+1,\mathrm{est}}\rangle$    on $\mathbf{P}(\Lambda^{l'+1})$, $l' < l_{\max}$
  Enddo
  Multilevel Solve: $l' \in \{l,\dots,l_{\max}\}$    ($\hat{E}_{\mathrm{R}}^{l',n}$ and $\hat{\mathbf{F}}^{l',n}$ are computed)
    ☆ $\left(\frac{\partial E_{\mathrm{R}}}{\partial t}\right)^{l',\mathrm{est}} = \nabla\cdot\left(\frac{c\lambda(E_{\mathrm{R}}^{l',n})}{\kappa_{\mathrm{R}}^{l',n}}\nabla\hat{E}_{\mathrm{R}}^{l',n}\right)$
            $+ \kappa_{\mathrm{P}}^{l',n}\left(B^{l',n} - c\hat{E}_{\mathrm{R}}^{l',n}\right) - D_{\mathrm{R}}(\delta\hat{\mathbf{F}}^{l'+1,n})$    on $\Lambda^{l'}$
    ☆ $\delta\hat{\mathbf{F}}^{l'+1,n} = \langle\hat{\mathbf{F}}^{l'+1,n}\rangle - \hat{\mathbf{F}}^{l',n}$    on $\mathbf{P}(\partial\Lambda^{l'+1})$, $l' < l_{\max}$
    ☆ $\delta\hat{E}_{\mathrm{R}}^{l'+1,n} = \langle\hat{E}_{\mathrm{R}}^{l'+1,n}\rangle - \hat{E}_{\mathrm{R}}^{l',n}$    on $\mathbf{P}(\Lambda^{l'+1})$, $l' < l_{\max}$
    ☆ $\delta\hat{E}_{\mathrm{R}}^{l'+1,n} = 0$    on $\Lambda^{l'} - \mathbf{P}(\Lambda^{l'+1})$
  End Multilevel Solve
Else If (not using multilevel init option)
  For $l' \in \{l,\dots,l_{\max}\}$ do
    ☆ $\delta\hat{\mathbf{F}}^{l'+1,n} = 0$    on $\mathbf{P}(\partial\Lambda^{l'+1})$, $l' < l_{\max}$
    ☆ $\delta\hat{E}_{\mathrm{R}}^{l'+1,n} = 0$    on $\Lambda^{l'}$
  Enddo
Endif

Level Time Step, level $l$:
  Multi-Fluid Advection    (not described here)
  Implicit Energy Update    (“check” quantities are computed)
    ☆ $\frac{\check{E}_{\mathrm{R}}^{l,n+1} - E_{\mathrm{R}}^{l,n}}{\Delta t^l} = \nabla\cdot\left(\frac{c\lambda(\check{E}_{\mathrm{R}}^{l,n+1})}{\check{\kappa}_{\mathrm{R}}^{l,n+1}}\nabla\check{E}_{\mathrm{R}}^{l,n+1}\right)$
            $+ \check{\kappa}_{\mathrm{P}}^{l,n+1}\left(\check{B}^{l,n+1} - c\check{E}_{\mathrm{R}}^{l,n+1}\right) - D_{\mathrm{R}}(\delta\hat{\mathbf{F}}^{l+1,n})$
    ☆ $(\rho\check{E})^{l,n+1} = (\rho E)^{l,-} - \Delta t^l\{(1-\theta)\left(\nabla\cdot\mathbf{F}^{l,n} - \nabla\cdot\check{\mathbf{F}}^{l,n+1}\right)$
            $+ \check{\kappa}_{\mathrm{P}}^{l,n+1}\left(\check{B}^{l,n+1} - c\check{E}_{\mathrm{R}}^{l,n+1} - c\delta\hat{E}_{\mathrm{R}}^{l+1,n}\right)\}$
  End Implicit Energy Update
  • $\bar{\mathbf{F}}^{l,n+\theta} = (1-\theta)\mathbf{F}^{l,n} + \theta\check{\mathbf{F}}^{l,n+1}$    on $\mathbf{P}(\partial\Lambda^{l+1})$, $l < l_{\max}$
  • Advance levels $l+1,\dots,l_{\max}$, including higher-level sync solves.
  • $\delta\bar{\mathbf{F}}^{l+1,n+\theta} = \langle\langle\bar{\mathbf{F}}^{l+1}\rangle\rangle - \bar{\mathbf{F}}^{l,n+\theta} - \delta\hat{\mathbf{F}}^{l+1,n}$    on $\mathbf{P}(\partial\Lambda^{l+1})$, $l < l_{\max}$
  • $\bar{\mathbf{F}}^{l,n+\theta} = (1-\theta)\mathbf{F}^{l,n} + \theta\check{\mathbf{F}}^{l,n+1}$    on $\mathbf{P}(\partial\Lambda^l)$, $l = l_{\max}$
End Level Time Step
If $(l < l_{\max})$ then    (synchronization/refluxing between levels $l$ and $l+1$)
  Multilevel Solve: $l' \in \{l,\dots,l_{\max}\}$    ($\tilde{E}_{\mathrm{R}}^{l',n+1}$ and $\tilde{\mathbf{F}}^{l',n+1}$ are computed)
    ☆ $\frac{\tilde{E}_{\mathrm{R}}^{l',n+1}}{\Delta t^l} = \nabla\cdot\left(\frac{c\lambda(\check{E}_{\mathrm{R}}^{l',n+1})}{\check{\kappa}_{\mathrm{R}}^{l',n+1}}\nabla\tilde{E}_{\mathrm{R}}^{l',n+1}\right) - \check{\kappa}_{\mathrm{P}}^{l',n+1}c\tilde{E}_{\mathrm{R}}^{l',n+1} - D_{\mathrm{R}}(\delta\bar{\mathbf{F}}^{l+1,n+\theta})$    on $\Lambda^{l'} - \mathbf{P}(\Lambda^{l'+1})$
    ☆ $\tilde{\mathbf{F}}^{l',n+1} = \langle\tilde{\mathbf{F}}^{l'+1,n+1}\rangle$    on $\mathbf{P}(\partial\Lambda^{l'+1})$, $l' < l_{\max}$
  End Multilevel Solve
  For $l' \in \{l_{\max}, \ldots, l\}$ do
    ☆ $(\rho E)^{l',n+1} = (\rho\check{E})^{l',n+1} + \Delta t^l \check{\kappa}_{\mathrm{P}}^{l',n+1} c\tilde{E}_{\mathrm{R}}^{l',n+1}$    on $\Lambda^{l'} - \mathbf{P}(\Lambda^{l'+1})$
    ☆ $E_{\mathrm{R}}^{l',n+1} = \check{E}_{\mathrm{R}}^{l',n+1} + \tilde{E}_{\mathrm{R}}^{l',n+1}$    on $\Lambda^{l'} - \mathbf{P}(\Lambda^{l'+1})$
    ☆ $(\rho E)^{l',n+1} = \langle(\rho E)^{l'+1,n+1}\rangle$    on $\mathbf{P}(\Lambda^{l'+1})$, $l' < l_{\max}$
    ☆ $E_{\mathrm{R}}^{l',n+1} = \langle E_{\mathrm{R}}^{l'+1,n+1}\rangle$    on $\mathbf{P}(\Lambda^{l'+1})$, $l' < l_{\max}$
  Enddo
  • $\bar{\mathbf{F}}^{l,n+\theta} = (1-\theta)\mathbf{F}^{l,n} + \theta\check{\mathbf{F}}^{l,n+1} + \tilde{\mathbf{F}}^{l,n+1}$    on $\mathbf{P}(\partial\Lambda^l)$, $l > 0$
Endif    (end synchronization/refluxing)
```

The final line of this algorithm includes a correction to the flux registers on $\mathbf{P}(\partial\Lambda^l)$ due to the $l, l+1$ sync, in preparation for the synchronization between levels $l-1$ and l. (In case the order of flux register operations is unclear, note that $\bar{\mathbf{F}}^l$ is set for the coarse level during its level time step, just before the finer levels are advanced. Finer level $\bar{\mathbf{F}}^{l+1}$ values are then set either at the end of the $l+1$ level time step (if $l+1 = l_{\max}$), or at the end of the following synchronization block (if $l+1 < l_{\max}$). Once execution returns to the coarse level, coarse and fine values of $\bar{\mathbf{F}}$ are combined to form $\delta\bar{\mathbf{F}}^{l+1}$.)

As written above, the sync solve is a backwards-Euler discretization of the radiation diffusion equation for the correction $\tilde{E}_{\mathrm{R}}$ over a level l timestep. We solve for this correction on levels l to $l_{\max}$ simultaneously, in a coupled fashion, and then apply it to the existing radiation and fluid energy fields. In many cases computational effort can be saved by performing the multilevel linear solve over a restricted set of levels—perhaps only l and $l+1$—and then interpolating the corrections to higher levels in a conservative fashion.

In our present implementation we have found it beneficial to use a more coupled formulation for the sync solve, with emission and absorption linearized about the change in fluid energy in much the same way as we did for the implicit update. This version alters the values passed to the multilevel linear solver, but no additional calls to the solver are required since we do not attempt to iterate the nonlinear effects to convergence. Without this linearization some runs showed irregularities in the fluid energy at the interfaces between levels. This feature is not difficult to implement, but is omitted from the outline given above to avoid complicating the notation any further.

11. Adaptive mesh results

We will now present three test problems to illustrate different characteristics of the AMR radiation algorithm. The first example is a pure radiation problem designed to explore the limitations of the method when important features are not resolved on coarser mesh levels. The second is a thermal wave calculation showing the code executing in parallel with a large 3D mesh. This example is also used to show the effect of the coarse/fine synchronization step, and to demonstrate that the adaptive mesh does not introduce significant grid artifacts into the solution. The final example is a physics simulation using realistic tabulated material data.

Even though we will present tables of numerical results, the examples are intended to be more illustrative than quantitative—we show that the solutions are not marred by serious grid effects, and that the method is practical for large-scale computations. We will say little about efficiency, and essentially nothing about parallel scalability. There are two main reasons for this: First, efficiency and scalability depend to a large degree on the underlying libraries. More than half of our time is spent in the *hypre* linear solvers. Speed these up, or substitute a different solver library, and the results are changed. Calls to physical data tables are also an important expense. Much of the remaining time is spent doing Godunov hydrodynamics, which is local, explicit, and therefore scales well in parallel, but is not the primary focus of this paper.

Second, the efficiency of AMR calculations depends strongly on the grid layout. This is true in serial as well as in parallel calculations. A refined region covering a major portion of the domain may be less efficient than one covering *all* of the domain, due to AMR overhead issues. Small grids have poor surface-to-volume ratios and therefore increase overhead. Large grids, on the other hand, can suffer from problems related to memory size and cache efficiency. Optimal parameters are architecture-dependent, and also influenced by the particular details of each application.

Our first test is a pure radiation problem, shown in Fig. 6. This example is not presented as a typical application of the method, rather it has been chosen to show how the AMR timestepping scheme loses accuracy when features critical to the global solution are not resolved on the coarse mesh. This gives insight into the AMR algorithm itself, and also provides motivation for the discussion of material interfaces presented earlier in the paper. In the single-grid examples we showed that fluxes across sharp interfaces could be computed accurately without a strong dependence on spatial or temporal resolution. Without that capability, such problems would become more difficult to solve with an AMR algorithm due to inadequate coarse grid solutions—perhaps as difficult as the example we are presenting now:

The problem is motivated by an idea of sunlight reaching the ground through gaps in a layer of cloud, but we make little attempt to establish realistic parameters for the physical system. The domain is a square in 2D Cartesian coordinates, 10^5 cm on a side. The coefficient κ_P is zero, decoupling the radiation field from the fluid energy, and all other parameters relating to the fluid are therefore irrelevant. The coefficient κ_R is 10^{-2} cm^{-1} within "clouds", and 10^{-6} cm^{-1} elsewhere. The clouds form a horizontal layer across the middle of the square. Each cloud is circular, gaps between the clouds are $1/10$ of a cloud diameter, and there are 8.5 clouds across the width of the domain. The top of the domain has a Marshak boundary condition with an imposed incident flux of 4×10^5 erg/cm^2/s, the bottom is a Marshak boundary with zero imposed flux, and the sides are reflecting (Neumann) boundaries. The speed of light c is 3×10^{10} cm/s. The timestep centering parameter θ is set to 1—this parameter primarily affects the coupling between radiation and the fluid, and is therefore not relevant for this problem.

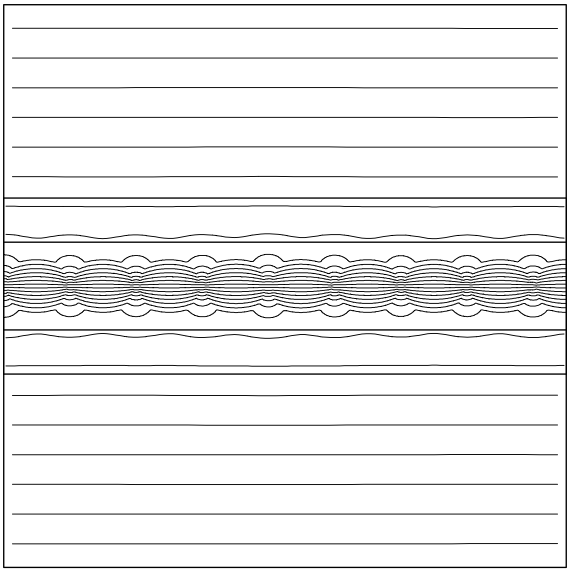

Fig. 6. Contour plot of radiation energy density for the first AMR test problem. Radiation enters the domain at the top and must pass through a row of circular "clouds" across the center of the domain before reaching the bottom. This particular example is a steady-state solution computed, with no flux limiter, on a 32 × 32 base grid with two levels of refinement around the cloud layer (the resolution labled "32,128,512" in Table 2). Fine grid boundaries are visible as thicker lines in the plot.

So much for the litany of parameters. Speaking more heuristically, what happens is that a flux of 400,000 units enters at the top of the domain and interacts with the clouds in the middle. If the material were transparent, all of the energy would reach the bottom and the average outward flux there would be 400,000. If the clouds were opaque, all of the energy would reflect back out the top, and the outward flux at the bottom would be zero. We place finer grids around the cloud layer only, since the solution is smooth in the other parts of the domain. Fig. 6 shows a steady-state solution with a 32 × 32 base grid and two levels of refinement, each by a factor of 4.

The narrow gaps between the clouds are what make this an interesting and difficult problem for AMR. On coarse grids the gaps are not resolved, and very little energy makes it through the cloud layer. On fine grids, where the gaps are well-resolved, the flux may actually be overestimated unless a flux limiter is used. The higher-resolution, non-limited fluxes given in Table 2 are over 350,000, too high to be physically realistic. Use of the flux limiter brings this down to the neighborhood of 100,000—probably still too high but nearer the right ballpark. A physically accurate answer may be more than we can realistically ask of a diffusion approximation. Fortunately, though, physical reality is not the immediate concern for this test problem. What we want to know is how AMR performs when important features are not resolved on the coarse grids. We can explore this question even with the physically unrealistic, non-limited solution. Since the limiter complicates the study, all of the following results except those in the last column of Table 2 were obtained with the limiter turned off.

The system as described so far has no time dependence, so we initialize all cases with a steady-state solution obtained using the multilevel linear solver. In each of the cases tested, this multilevel solution gives a close match to the fluxes computed with a uniform grid covering the entire domain at the same resolution as the finest level. The hard part of the calculation is the time stepping. The AMR timestep algorithm first advances the coarse level, then advances the finer levels using boundary data (interpolated in space and time) from the coarser levels ([1] gives a detailed exposition of the boundary treatment). In the current example, though, the coarse grid solution is wildly inaccurate, even in parts of the domain outside the fine grids. The sync step brings the new-time fluxes (third column of Table 2) up to the right order of magnitude, but does not restore the accuracy of the true multilevel solution.

Table 2
Spatially averaged flux across the lower boundary of the problem of Fig. 6 at time $t = 1$, with a constant incident flux imposed at the top boundary

Resolutions	Total cells	Standard init option		Multilevel init option		Limiter
	Advanced	Flux	Time	Flux	Time	Flux
32	1024	12,709	0.183	12,709	0.167	12,632
64	8192	15,636	0.867	15,636	0.867	15,133
128	65,536	320,550	5.917	320,550	5.900	65,223
32,128	21,504	222,617	4.167	320,996	3.083	65,233
256	524,288	353,040	49.00	353,040	48.17	91,371
64,256	122,880	276,431	17.30	353,314	15.73	91,380
512	4,194,304	356,853	490.2	356,853	489.7	110,309
128,512	720,896	347,588	112.9	356,993	114.4	110,316
32,128,512	676,864	207,106	114.1	357,556	110.1	110,341
256,1024	4,980,736	356,834	858.3	358,424	958.0	
64,256,1024	4,579,328	273,660	857.1	358,706	831.8	

Timesteps on each level are in proportion to the spatial mesh size, with a 32 × 32 grid taking a single timestep. Timings are in seconds on a single Compaq Alpha EV5.6 processor. The flux limiter is used only for the last column of tests.

Resolutions are specified by the numbers of cells required to span the domain, so that, for example "64,256" refers to a 64 × 64 base grid and one level of refinement by a factor of 4.

Significant improvement is obtained by using the multilevel init option to the AMR timestep algorithm. This version stores flux corrections ($\delta\hat{\mathbf{F}}$) that are applied during the coarse level advance to approximate the effects of the finer levels. The cost is an additional multilevel solver call at the beginning of each coarse timestep. Since the test problem is constant in time the corrected solution recovers the accuracy of the steady-state multilevel solve (column five of Table 2). The sync is nearly eliminated, so the timings (column six) are similar to those for the standard init option (column four).

The final column of Table 2 gives fluxes computed with the flux limiter turned on. This makes the problem nonlinear, so the initial steady-state solution requires repeated calls to the multilevel solver until the limiter is converged. This is a very slow process, and requires considerably more time than the time stepping algorithm that we have been examining. For this reason we do not report timings for this case, and did not complete the runs with the finest grid resolutions. We can report, however, that once the initialization phase is complete the times required for these runs are very similar to those needed without the limiter. Also note that since the multilevel init option is used for these runs, the computed fluxes depend almost entirely on the finest grid resolution. (It is only the problem initialization that is expensive, not the multilevel init option itself, since that does not involve a nonlinear iteration.)

While it is reassuring that the multilevel init option can recover the accuracy of a steady-state solution, what we really care about are time-dependent solutions. For Table 3 we have replaced the constant incident flux across the top boundary with a time-varying flux of $(t^2 + t + 2) \times 100{,}000$ units. Note that this boundary condition is applied to the coarse part of the mesh; fine grids are not exposed to it directly. The multilevel init option still improves the solutions, but its flux estimates are no longer correct for the new time level. Solutions obtained with base grids that do not resolve the gaps between clouds (32×32 and 64×64) are poor. Even with a very coarse base grid, though, the lower portion of the table shows that computed fluxes converge to first order in Δt.

We stress that this test case was chosen to stress the AMR algorithm to its limits. The degraded accuracy at some resolutions is due to the instantaneous and time-varying dependence of the coarse grid solution on the resolution at finer grid levels. Most problems do not exhibit this behavior to such an extreme degree. When the multilevel init option is used, acceptable performance can be obtained either when the coarse

Table 3
Spatially averaged flux across the lower boundary of the problem of Fig. 6 at time $t = 1$, with temporally varying incident flux imposed at the top boundary

Resolutions	Coarse steps	Fine steps	Cells advanced	Flux	Time
32	1	1	1024	12,708	0.167
64	2	2	8192	15,636	0.717
128	4	4	65,536	320,549	5.917
32,128	1	4	21,504	262,805	3.967
256	8	8	524,288	353,039	50.02
64,256	2	8	122,880	321,923	19.53
512	16	16	4,194,304	356,852	492.0
128,512	4	16	720,896	354,359	134.2
32,128,512	1	16	676,864	278,873	138.1
256,1024	8	32	4,980,736	357,859	1117.0
64,256,1024	2	32	4,579,328	322,334	1050.0
32,128,512	1	16	676,864	278,873	138.1
32,128,512	2	32	1,353,728	320,895	274.4
32,128,512	4	64	2,707,456	337,735	548.9
32,128,512	8	128	5,414,912	347,373	1093.0

Columns two and three specify the numbers of timesteps on the coarsest and finest levels, respectively. The multilevel init option is used, with no flux limiter. In the upper portion of the table timesteps are determined in the same way as for Table 2, while the lower part of the table shows the temporal convergence behavior for a single fixed grid layout.

level resolves the essential features of the solution, or when rapid resolution-dependent temporal variation is confined to the fine level. (If neither of these conditions hold, the best approach might be to abandon subcycling, and advance all levels at the same timestep.) When the multilevel init option is not used, the method works well only when the coarse level resolves the essential features of the solution. Finally, careful examination of the timings presented in the tables suggests that the multilevel solver is a disproportionately expensive part of the algorithm. (In some cases this solver accounts for more than half of the total time.) New features being added to the *hypre* library should improve the multilevel solver performance significantly.

The multilevel init option is not used for any of the remaining examples in this paper, since they do not exhibit the extreme dependence on resolution that would make it necessary.

Our next test is a 3D nonlinear thermal wave, computed in an insulating unit cube with a $64 \times 64 \times 64$ base level and a single additional level refined by a factor of 2. We take a single fluid, initially cold ($T = e = 10^{-6}$, $c_v = 1$, $\rho = 1$), and add a hot spot against one wall with total energy 0.5. Constants are $\sigma = 10^{-2}$, $c = 10^8$, coefficients are $\kappa_\mathrm{R} = T^{1/2}$, $\kappa_\mathrm{P} = 10^5 T^{-1}$. Hydrodynamics is enabled, with $\gamma = 1.4$. (This seemingly bizarre combination of parameters has made the problem well-suited for exploring a variety of issues, including radiation-fluid coupling, linearization strategies, flux limiters, and so on. Most of these will not be detailed here.)

At early time the hot fluid radiates over 96% of its energy into the radiation field. The hot spot expands through nonlinear diffusion with a sharp thermal front and very little fluid motion. As the heated region expands and cools, the progress of this front slows dramatically. Energy moves from the radiation back into the fluid. At later times a shock develops, and hydrodynamics takes over as the dominant physical process.

In order to capture the behavior over the wide range of time scales, we use a timestep $\Delta t = 1.1^{n-1} \times 10^{-15}$ for step n on the coarse level. Time differencing uses the centered scheme ($\theta = 1/2$). Nonlinearities are converged. The flux limiter is turned off, so that we can compare against an analytic solution, presented in [26], that is valid during the thermal wave phase. A simple refinement criterion is used which places fine grids near the front, rearranging them as needed every two coarse timesteps.

The job was run in parallel on an IBM SP-2. Grid size was limited to a maximum of $16 \times 16 \times 16$, so there were 64 grids at level 0. Level 1 went from an initial 4 grids to 357 grids comprising 30% of the domain at timestep 280 ($t = 0.00389$). We used 32 processors for the early part of the calculation, increasing this to 64 after timestep 200.

Fig. 7 shows radiation energy, density, temperature, and the difference between computed temperature and the analytic solution. The slices are taken in three perpendicular planes containing the original hot spot, to show that the grid arrangement is not completely symmetric. The grid generator [7] fits grids around all tagged cells, then trims and splits grids as necessary until a specified efficiency is reached. The resulting grid arrangement is typically not symmetric, even for symmetric data as we have here. The important thing to note, though, is that the asymmetric grid distributions have not caused noticeable asymmetries in the computed solution. Fig. 8 further illustrates this point.

The solution near the core of the hot region is smooth, and has therefore been de-refined. Significant energy fluxes cross the coarse–fine interfaces inside the hot region. No grid artifacts associated with these interfaces are visible in the solutions. The most noticeable differences between the computed and the analytic solution are simply discretization effects at the sharp front. The system conserves energy from beginning to end of the calculation to the full accuracy of the multigrid solver.

Fig. 9 shows the same system recomputed several times in 1D spherical coordinates to compare the effects of different options for the sync solve. The AMR time step algorithm in the previous section presented the synchronization step as the computation of a correction $\tilde{E}_\mathrm{R}$ to the radiation field, which was then applied directly to the radiation and fluid energies. Use of this form of the sync with $\theta = 1$ produces a solution with an obvious zig-zag marking the location of the coarse–fine interface. Linearization of the coupling between the fluid and radiation energies, as is done in the level solves, produces the smoother solid

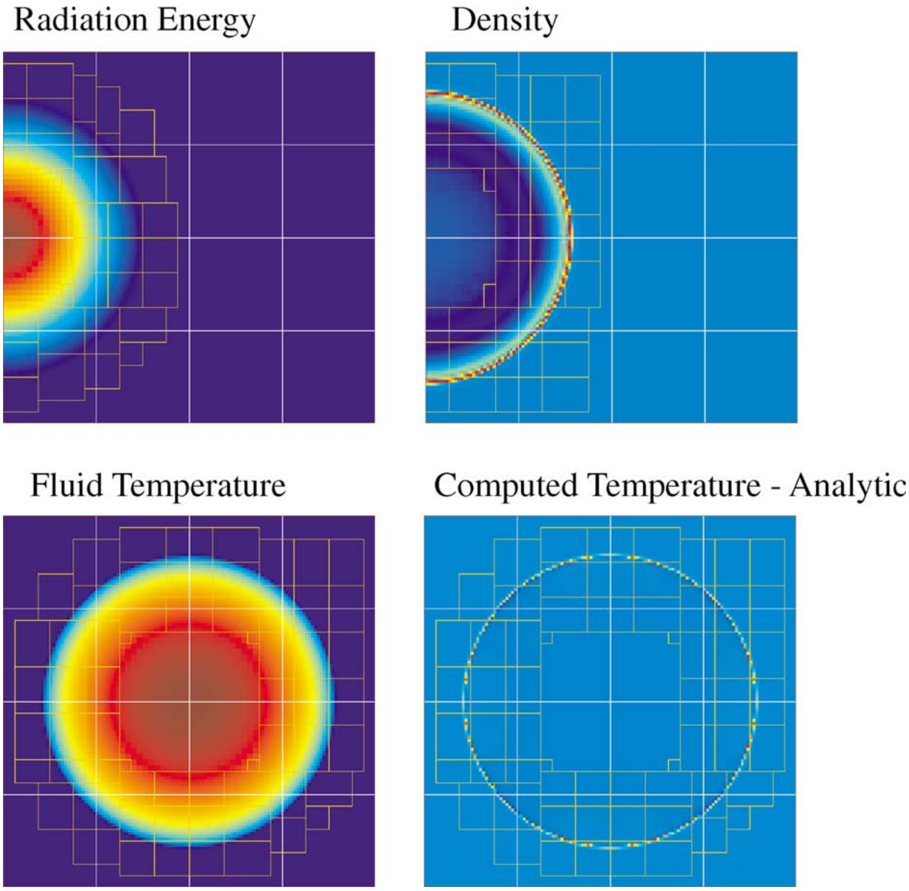

Fig. 7. A nonlinear thermal wave expanding from a point heat source in 3D, with adaptive mesh refinement. Different slices are shown to illustrate asymmetries in grid layout. Fields shown are radiation energy, density, temperature, and the difference between computed temperature and an analytic solution. The maximum value in this last field is only a tenth of the maximum temperature. At this point in the calculation the density variations shown are less than 1% of the background density, as radiation transport is still dominant.

line. A time-centered discretization ($\theta = 1/2$) yields the still more accurate dotted line—linearization of the sync for this case makes too small a difference to see clearly in the plot. For comparison, the two dashed lines show the nonconservative results for both time discretizations with no sync solves at all. The upper curve uses $\theta = 1$ and has gained about 2% over the original energy in the problem, while the lower curve with $\theta = 1/2$ has lost about 1% of its energy.

Tables 4 and 5 show the convergence behavior for this problem as the grids are refined. Timestep sequences are chosen so that doubling the resolution of the coarse grid doubles the number of coarse timesteps. For all of the tests shown—both single grid and AMR—the dominant errors occurred at the front. The basic features seen in Fig. 9, including the greatly reduced errors behind the front for $\theta = 1/2$, were reproduced at all resolutions. The error norms changed significantly from one timestep to the next as the front moved from cell to cell, so the exact numbers presented here should not be given too much weight in the interpretation.

Conclusions we can draw from the convergence study are as follows: First, the effects of hydrodynamics were not significant at this stage in the calculation and can safely be ignored. Second, results were improved by use of the time-centered formulation, even though convergence rates were limited to first order (possibly

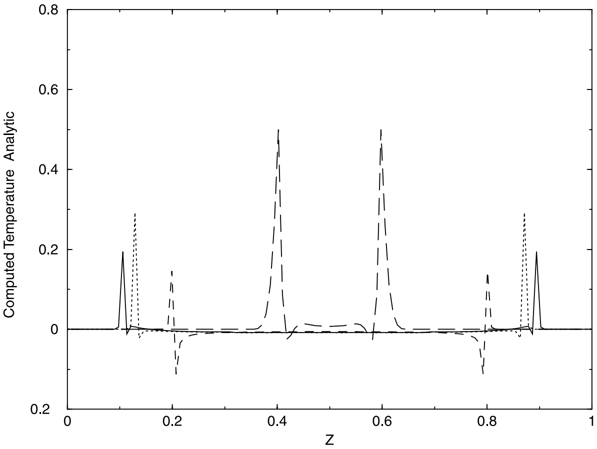

Fig. 8. Four cross-sections through the fourth (lower-right) plot of Fig. 7: difference between the computed temperature and the analytic solution. That plot is in the yz plane, in this one cuts are shown at $y = 0.5$ (solid), 0.625 (dots), 0.75 (dashes) and 0.875 (long dashes). The point is that each curve is symmetric, even though the grid layout is not.

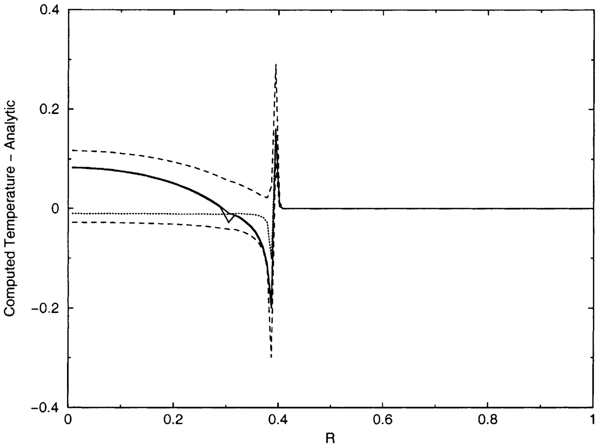

Fig. 9. The thermal wave of Fig. 7, recomputed in 1D spherical coordinates with different algorithmic options. The difference between the computed temperature and the analytic solution is shown as a function of radius. Solid lines: backward-Euler discretization ($\theta = 1$), upper line uses linearized sync, lower line uses the sync as presented in the previous section (these lines only differ visibly near $R = 0.3$). Dotted line: time-centered discretization ($\theta = 1/2$). Dashed lines: no sync solve, nonconservative, upper line has $\theta = 1$, lower line has $\theta = 1/2$.

due to the dominance of errors at the front). Third, the sync solve was essential to the performance with an adaptive mesh. (Though the characteristic zig-zag associated with the simple sync was visible in AMR plots at all resolutions, no significant difference between the simple and linearized sync options is observable in the norms.) Fourth, and most important, errors on adaptive meshes corresponded most closely to errors obtained with uniform meshes equivalent in resolution to the finest AMR grid levels.

A final test of the code includes multiple materials with realistic tabular equations of state and opacities [25]. It is a prototype of a class of experiments featuring supersonic jets and interacting shock waves currently being conducted on the Omega Laser at the University of Rochester [24]. In these experiments, laser light is directed into two gold hohlraums surrounding the experimental package as shown in Fig. 10.

Table 4
Single grid convergence study for the thermal wave problem in 1D spherical coordinates

Resolutions	Cells advanced	Hydro off		Hydro on	
		$\theta = 1$	$\theta = 1/2$	$\theta = 1$	$\theta = 1/2$
32	4480	0.0168	0.0149	0.0169	0.0150
64	17920	0.00597	0.00337	0.00580	0.00328
128	71680	0.00303	0.00162	0.00289	0.00156
256	286720	0.00168	0.00096	0.00155	0.00094

Since the analytic solution does not include hydrodynamic effects, results are shown both with and without hydrodynamics. Values given are the 1-norm differences between the computed temperatures and the analytic solution.

Table 5
Adaptive mesh convergence study for the thermal wave problem in 1D spherical coordinates with hydrodynamic effects included

Resolutions	Cells advanced	No sync		Simple sync		Linearized sync	
		$\theta = 1$	$\theta = 1/2$	$\theta = 1$	$\theta = 1/2$	$\theta = 1$	$\theta = 1/2$
32,64	8480	0.00637	0.00417	0.00560	0.00330	0.00592	0.00332
64,128	27136	0.00466	0.00414	0.00356	0.00162	0.00362	0.00163
128,256	92912	0.00389	0.00370	0.00221	0.00097	0.00222	0.00097

The figures for cells advanced are based on the $\theta = 1/2$ case with linearized sync. (Other options gave slightly different figures because grid generation was based on computed solution gradients.) Values given are the 1-norm differences between the computed temperatures and the analytic solution.

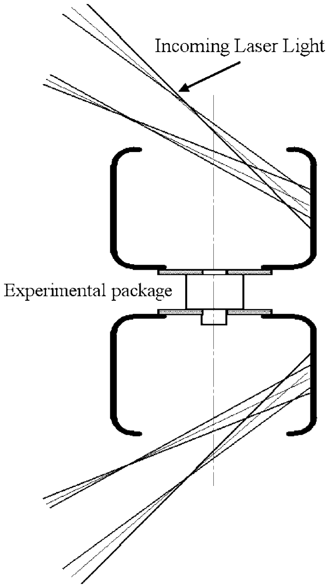

Fig. 10. The experimental package is situated between two gold hohlraums as shown in this axisymmetric view. Laser light enters through the upper and lower holes and shines on the inner gold surfaces producing a hot plasma that serves as the radiation source to drive the experiment.

The gold absorbs the laser energy and emits X-rays forming a hot plasma, or thermal radiation bath, within the hohlraum. This hot plasma serves as the radiation source that then drives the experiment. Note that in the simulations presented here, no attempt is made to explicitly model the hohlraum.

A detailed view of the experimental package is shown on the left side of Fig. 11. The materials are labelled in the figure. They are initialized to room temperature (298 K) with initial densities of $\rho_{\mathrm{CH(1)}} = 1.046$ g/cc, $\rho_{\mathrm{CH(2)}} = 0.001$ g/cc, $\rho_{\mathrm{Al}} = 2.7$ g/cc and $\rho_{\mathrm{Mo}} = 10.228$ g/cc. The plastics (polystyrene), CH(1) and CH(2) indicate two different densities, not different compositions. Molybdenum is used as a simulant for the gold in the physical experiment because material data tables for gold were not available. The entire computational domain is shown in the figures. The geometry is axisymmetric about the left boundary (z-axis). The radial extent (left to right) is 0.1 cm and the axial extent is 0.2 cm. For the hydrodynamics, the axis of revolution is a symmetry boundary while all others are inflow/outflow. For the radiation field, all boundaries are homogeneous Neumann (i.e., insulated).

The simulation was run with a 64 × 128 base grid and 3 levels of refinement, each by a factor of 2. This gave an effective resolution of 2 μm on the finest level. All mixed cells were tagged for refinement at the finest level; this was the only refinement criterion used. A CFL number of 0.3 was used to determine Δt. Each call to the radiation solver iterated the nonlinearity in the blackbody emission to convergence, but held the absorption coefficients fixed.

In the calculation, radiation energy is introduced into the low density plastic regions symmetrically both above and below the aluminum blocks as a function of time. This effectively models the radiation field

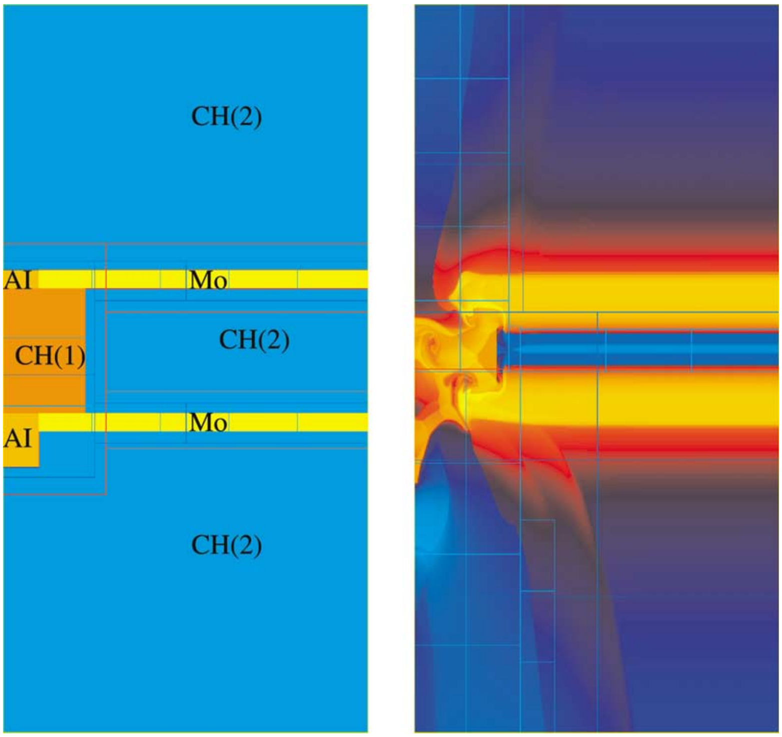

Fig. 11. The initial and final ($t = 7.43$ ns) $\log(\rho)$ fields for the Omega Laser supersonic jet and interacting shock wave experiment are shown from left to right, respectively. The materials are as labeled and designated by color. CH(1) and CH(2) are polystyrene at two different densities, Al is aluminum and Mo is molybdenum. At the final time the upward propagating Al jet and the counter-propagating jet have interacted. The AMR grid layout is shown for reference.

produced by the 1 ns pulse laser incident onto the hohlraums. As the radiation burns into the metals (Al and Mo), it ablates and drives shock waves into the materials. Note that the asymmetry in the aluminum blocks (top to bottom) leads to an asymmetric flow later in time. Converging shock waves in the lower Al block lead to the formation of a supersonic jet and accompanying bow shock. In the upper aluminum block, a nearly planar shock is launched that later interacts with the jet. This is shown with $\log(\rho)$ in the right side of Fig. 11 at $t = 7.43$ ns. The physical experiment was imaged by radiography (5 keV photon energy backlighter). Simulated radiographs prepared from the numerical simulation showed good qualitative agreement with the experimental results, and more detailed work is currently underway.

Acknowledgements

The authors gratefully acknowledge Drs. Bernard Wilde, John Foster, Ted Perry, Paula Rosen, and Mike Fell for providing the experimental data corresponding to the simulation in Fig. 11. We also thank Dr. John Edwards for designing simulations of the experiment and assessing the performance of the algorithms described herein.

References

[1] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1.

[2] A.S. Almgren, T. Buttke, P. Colella, A fast adaptive vortex method in three dimensions, J. Comput. Phys. 113 (1994) 177.

[3] C. Baldwin, P.N. Brown, R. Falgout, F. Graziani, J. Jones, Iterative linear solvers in a 2D radiation-hydrodynamics code: methods and performance, J. Comput. Phys. 154 (1999) 1.

[4] J. Bell, M. Berger, J. Saltzman, M. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, SIAM J. Sci. Comput. 15 (1994) 127.

[5] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64.

[6] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484.

[7] M.J. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, IEEE Trans. Syst, Man Cybern. 21 (1991) 1278.

[8] P.N. Brown, R.D. Falgout, J.E. Jones, Semicoarsening multigrid on distributed memory machines, SIAM J. Sci. Comput. 21 (2000) 1823.

[9] E. Chow, A.J. Cleary, R.D. Falgout, Design of the hypre preconditioner library, in: SIAM Workshop on Object Oriented Methods for Interoperable Scientific and Engineering Computing, Yorktown Heights, NY, October 21–23, 1998.

[10] R.D. Falgout, J.E. Jones, Multigrid on massively parallel architectures, in: Sixth European Multigrid Conference, Ghent, Belgium, September 27–30, 1999.

[11] J.A. Greenough, D.T. Graves, P. Colella, A second-order method for interface reconstruction in orthogonal coordinate systems, in: Proceedings of the Nuclear Explosives Code Development Conference, Las Vegas, October 25–30, 1998.

[12] L.H. Howell, J.B. Bell, An adaptive mesh projection method for viscous incompressible flow, SIAM J. Sci. Comput. 18 (1997) 996.

[13] L.H. Howell, J.A. Greenough, A block-structured adaptive mesh refinement algorithm for diffusion radiation, in: Proceedings of the Nuclear Explosives Code Development Conference, Las Vegas, October 26–30, 1998.

[14] L.H. Howell, R.B. Pember, P. Colella, J.P. Jessee, W.A. Fiveland, A conservative adaptive mesh algorithm for unsteady, combined-mode heat transfer using the discrete ordinates method, Num. Heat Trans. B 35 (1999) 407.

[15] D.A. Knoll, W.J. Rider, G.L. Olson, An efficient nonlinear solution method for non-equilibrium radiation diffusion, J. Quant. Spect. Rad. Trans. 63 (1999) 15.

[16] D.A. Knoll, W.J. Rider, G.L. Olson, Nonlinear convergence, accuracy, time step control in nonequilibrium radiation diffusion, J. Quant. Spect. Rad. Trans. 70 (2001) 25.

[17] C.D. Levermore, G.C. Pomraning, A flux-limited diffusion theory, Astrophys. J. 248 (1981) 321.

[18] S.F. McCormick, Multilevel Adaptive Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1989.

[19] G.H. Miller, E.G. Puckett, A high-order Godunov method for multiple condensed phases, J. Comput. Phys. 128 (1996) 134.

[20] C.A. Rendleman, V.E. Beckner, M. Lijewski, W. Crutchfield, J.B. Bell, Parallelization of structured, hierarchical adaptive mesh refinement algorithms, Comput. Visual. Sci. 3 (2000) 147.

[21] W.J. Rider, D.A. Knoll, G.L. Olson, A multigrid Newton–Krylov method for multimaterial equilibrium radiation diffusion, J. Comput. Phys. 152 (1999) 164.

[22] S. Schaffer, A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients, SIAM J. Sci. Comput. 20 (1998) 228.

[23] J.K. Truelove, R.I. Klein, C.F. McKee, J.H. Holliman II, L.H. Howell, J.A. Greenough, The Jeans condition: a new constraint on spatial resolution in simulations of isothermal self-gravitational hydrodynamics, Astrophys. J. 489 (1997) L179.

[24] B.H. Wilde, P.A. Rosen, J.M. Foster, M. Fell, T.S. Perry, Simulations of supersonic jet and shock interaction experiments at OMEGA, American Physical Society, Division of Plasma Physics, Quebec City, Canada, October 23, 2000. [Also see LANL Report LA-UR-00-3194].

[25] D. Young, E. Corey, Release of new LEOS equation of state library, Lawerence Livermore National Laboratory unpublished memo, April 1, 1998.

[26] Ya.B. Zel'dovich, Yu.P. Raizer, in: Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena, vol. 2, Academic Press, New York, 1967, pp. 664–672.