LBNL-41311
Preprint

A Conservative Adaptive-Mesh Algorithm for Unsteady, Combined-Mode Heat Transfer Using the Discrete Ordinates Method

Louis H. Howell, Richard B. Pember, Phillip Colella,
J. Patrick Jessee, and Woodrow A. Fiveland

Computing Sciences Directorate

January 1998

Submitted to
Numerical Heat Transfer,
Part B: Fundamentals

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or The Regents of the University of California.

Ernest Orlando Lawrence Berkeley National Laboratory
is an equal opportunity employer.

LBNL-41311

A Conservative Adaptive-Mesh Algorithm for Unsteady, Combined-Mode Heat Transfer Using the Discrete Ordinates Method

Louis H. Howell,* Richard B. Pember,** and Phillip Colella

Computing Sciences Directorate
National Energy Research Scientific Computing Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

J. Patrick Jessee and Woodrow A. Fiveland

Research and Development Division
McDermott Technology, Inc.
Alliance, Ohio 44601

January 1998

*Current affiliation and contact information: L-560, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, (510) 422-6105, howell4@llnl.gov

**Current affiliation and contact information: L-560, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, (510) 422-4549, pember@llnl.gov

This work was supported by the Office of Energy Research, Office of Computational and Technology Research, Mathematical, Information, and Computational Sciences Division, Applied Mathematical Sciences Program of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098, at Lawrence Berkeley National Laboratory. Additional support from this Program was provided at McDermott Technology, Inc., under Subcontract No. 3643100.

A CONSERVATIVE ADAPTIVE-MESH ALGORITHM FOR UNSTEADY, COMBINED-MODE HEAT TRANSFER USING THE DISCRETE ORDINATES METHOD[1]

Louis H. Howell[2], Richard B. Pember, and Phillip Colella
Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA

J. Patrick Jessee and Woodrow A. Fiveland
Research and Development Division, McDermott Technology, Inc.
Alliance, Ohio 44601, USA

Abstract

Adaptive mesh refinement (AMR) permits a computational algorithm to allocate additional grid resolution when and where it is most needed. A block-structured AMR scheme provides this capability without sacrificing the numerical and computational efficiencies associated with regular meshes. In this paper we describe an AMR implementation of the discrete ordinates method for radiative transfer, coupled with an existing projection method for low-Mach number flows. The complete algorithm constitutes a conservative scheme for unsteady combined-mode heat transfer. Results for two- and three-dimensional problems are shown.

Introduction

The computational modeling of practical combustion applications with limited computer resources can be made difficult by the presence of multiple length scales and high gradients and by the large number of species in a sufficiently detailed reaction mechanism. In addition, the locations of regions with high gradients or small length scales may change over time. An accurate prediction of a reacting flow may therefore require that the computational grid be dynamically adapted both in time and space.

Radiative heat transfer is the dominant mode of heat transfer in many combustion applications. Because it can significantly affect temperatures, and, hence, density distributions and reaction rates, radiative heat transfer may be very influential in combustion dynamics. Deterministic methods for radiative heat transfer have been formulated for globally refined, nonuniform grids (see, for example, [1, 2]). However, most of these methods use fixed computational grids and are appropriate only for steady flows.

In this paper we present a method based on a different approach, local adaptive mesh refinement (AMR). We develop an AMR algorithm to solve a system of equations for unsteady combined-mode heat transfer, using a hierarchical grid structure approach first developed

[1]Support for this work was provided by the Applied Mathematical Sciences Program of the DOE Office of Mathematics, Information, and Computational Sciences under contract DE-AC03-76SF00098. Additional support was provided by the Applied Mathematical Sciences Program of the DOE Office of Mathematics, Information, and Computational Science under Subcontract No. 3643100.

[2]Current Affiliation and contact information: L-560, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, (510) 422-6105, howell4@llnl.gov

NOMENCLATURE

c_p	specific heat at constant pressure, kJ/kg·K
D_R	Reflux divergence operator
e_{ij}	estimate of error in numerical solution in cell ij
$\dot{E}^n$	Correction term in multilevel algorithm, J/(m^{3}s)
h	Enthalpy per unit mass, J/kg
I	Instantaneous radiant intensity, W/(m$^2 \cdot$ Sr)
$\hat{I}_m^n$	Extension of I_m to coarse cells under fine grids
$\check{I}_m^{n+1}$	Trial solution to I_m at new time level
$\bar{I}_m^{n+1/2}$	Time-averaged intensity over a time step
$\tilde{I}_m$	Nonphysical multilevel solution used to enforce energy conservation
$\delta I_m^{\ell+1}$	Flux register for I_m at interface between levels ℓ and $\ell+1$ (at level ℓ resolution)
I_b	Blackbody intensity, $\sigma_b T^4/\pi$
L	Length of pipe, channel, or duct in computational domain
$L_1^{\Delta x}$	Estimate of error in numerical solution over entire domain
ℓ	Level index
ℓ_{max}	Level index of finest refinement level
m	Ordinate index
N	conduction-radiation parameter, $\rho\kappa\lambda/4\sigma_b T_0^3$
n	Time index
$\hat{n}$	Outward unit normal at a boundary
$\mathbf{P}$	Level projection operator
p	Fluid pressure, kg/ms^2
Pr	Prandtl number, $\mu c_p/\lambda$
q	Fluid pressure, kg/ms^2
r	Refinement ratio between levels
r	radial (cylindrical) coordinate
$\mathbf{q}_r$	radiative heat flux, W/m^2
R	Pipe radius or duct half-width or parallel plate channel half-width, m
R_m	Refluxing source at coarse-fine interfaces
Re	Reynolds number, $2R\rho u_0/\mu$
S	Radiation source term, W/(m$^3 \cdot$ Sr)
T	Temperature, K
T_b	Bulk temperature, K
T_w	Wall temperature, K
t	Time, s
U	Fluid velocity, m/s
u	x component of velocity, m/s
u_0	Mean inlet or initial axial velocity, m/s
v	y (r, if cylindrical) component of velocity, m/s
w	z component of velocity, m/s
w_m	Ordinate weight
x	axial coordinate
y, z	off-axis (Cartesian) coordinates
Δt	Size of a single time step, s
$\Delta x, \Delta y$	Width, height of a computational cell
ϵ	Wall emissivity
Θ	dimensionless time, $t\mu/R^2\rho\,\mathrm{Pr}$
κ	Absorption coefficients, m^{-1}
Λ_ℓ	Collection of computation cells at refinement level ℓ
λ	thermal diffusivity, W/m·K
$\partial\Lambda_\ell$	Boundary between levels $\ell-1$ and ℓ
μ	Dynamic viscosity, kg/m·s
μ, ξ, η	Direction cosines
ρ	Fluid density kg/m^3
ρ	Wall reflectivity
σ	Scattering coefficient, m^{-1}
σ_b	Stefan-Boltzmann constant, 5.669×10^{-8}W/M^2 s
τ	optical thickness, κR
$\mathbf{\Omega}_m$	Ordinate direction unit vector

Subscripts and superscripts

$(\cdot)_{ij}$	value at center of cell ij
$(\cdot)_{i,j+1/2}$	value at upper y-edge of cell ij
$(\cdot)^n$	value at time t^n
$(\cdot)^{n+1}$	value at time $t^n + \Delta t = t^{n+1}$
$(\cdot)^{n+1,p}$	predicted value at center of time t^n
$(\cdot)^{n+1/2}$	value at time $t^n + \Delta t/2$
$(\cdot)_{i+1/2,j}$	value at upper x-edge of cell ij
0	initial or inlet values
w	wall values

Other

$\langle\cdot\rangle$	Average from fine to coarse level, space only
$\langle\langle\cdot\rangle\rangle$	Average from fine to coarse level, space and time

by Berger and Oliger [3] and Berger and Colella [4] for hyperbolic conservation laws in two dimensions, and extended to three dimensions by Bell *et al.* [5]. The grid structure is dynamic in time and is composed of nested uniform rectangular grids of varying resolution. By using grids of finer resolution in both space and time in the regions of most interest, AMR allows one to model large problems more efficiently. The integration algorithm on the grid hierarchy is a recursive procedure in which a coarse grid is advanced, fine grids are

advanced multiple steps to reach the same time as the coarse grid, and the coarse and the fine grids are synchronized. The method is valid for multiple grids on each level and for multiple levels of refinement.

The central feature of the algorithm presented here is a new scheme for modeling unsteady radiative transport with a discrete ordinates method [6, 7, 8] on locally refined meshes. This scheme incorporates portions of the adaptive mesh refinement discrete ordinates algorithm for solving instantaneous or steady-state radiative heat transfer problems described in [9, 10]. The method is also based on two other earlier works, an incompressible adaptive mesh refinement algorithm (IAMR) [11] and a single grid projection method for unsteady low-Mach number combustion [12]. (There are other computational approaches for unsteady low-Mach number flows on locally refined meshes and for unsteady low-Mach number combustion; see the literature reviews in [11] and [13], respectively.)

The remainder of this paper is broken into four sections. The first section presents the governing equations. The second section discusses the numerical method, including a review of the three works on which it is based. The third section presents and discusses numerical results computed with the algorithm. The last section summarizes the work and discusses future directions.

Governing Equations

We make the following simplifications for the sake of exposition. The fluid is an incompressible single component gas with constant and uniform density, viscosity, thermal diffusivity, and specific heat. Body forces and the heat production due to viscous stress are negligible. The governing equations for unsteady combined mode heat transfer are then the following:

$$\frac{\partial U}{\partial t} + (U \cdot \nabla)U = \frac{1}{\rho}(-\nabla p + \mu \nabla^2 U) \tag{1}$$

$$\frac{\partial \rho c_p T}{\partial t} + \nabla \cdot \rho U c_p T = \lambda \nabla^2 T - \nabla \cdot \mathbf{q}_r \tag{2}$$

$$\nabla \cdot U = 0. \tag{3}$$

Eq. (2) is written in conservative form to underscore the fact that the algorithm described below conserves enthalpy.

This paper considers an emitting-absorbing and isotropically scattering gray medium, although the discrete ordinates method is not restricted to these conditions. For this medium, the radiative transfer equation (RTE) is

$$(\mathbf{\Omega} \cdot \nabla)\, I(\mathbf{\Omega}) + (\kappa + \sigma) I(\mathbf{\Omega}) = \kappa I_b + \frac{\sigma}{4\pi} \int_{4\pi} I\left(\mathbf{\Omega}'\right) d\mathbf{\Omega}'. \tag{4}$$

The divergence of the radiative heat flux is given by

$$\nabla \cdot \mathbf{q}_r = 4\pi\kappa I_b - \kappa \int_{4\pi} I\left(\mathbf{\Omega}'\right) d\mathbf{\Omega}'. \tag{5}$$

For gray surfaces which reflect diffusely, the radiative boundary condition for (4) is given by

$$I(\mathbf{\Omega}) = \epsilon I_b + \frac{\rho}{\pi} \int_{\hat{n}\cdot\mathbf{\Omega}'>0} (\hat{n}\cdot\mathbf{\Omega}')\, I(\mathbf{\Omega}') d\mathbf{\Omega}' \tag{6}$$

for $\hat{n}\cdot\mathbf{\Omega} < 0$ where $I(\mathbf{\Omega})$ is the intensity leaving the surface and ρ is the surface reflectivity.

Numerical Algorithm

The algorithm is derived from three previous works: the incompressible adaptive mesh refinement (IAMR) algorithm described in [11], the adaptive mesh refinement discrete ordinates algorithm for solving instantaneous or steady-state radiative heat transfer problems [9, 10], and the single grid projection method for low-Mach number combustion described in [12]. In the next three subsections, we briefly review these in the context of this paper in order to lay the groundwork for the discussion of the AMR scheme for unsteady radiative heat transfer in the fourth subsection. In the fifth subsection, we briefly discuss extensions to flows with variable properties.

AMR for Unsteady Convective Heat Transfer

IAMR [11] solves the unsteady Navier-Stokes equations for incompressible flow (Eqs. (1) and (3)) along with additional equations for advected and diffused scalars on a block-structured adaptive mesh (Figure 1). For the discussion here, we consider one such additional equation, the energy equation for unsteady convective and diffusive heat transfer:

$$\frac{\partial \rho c_p T}{\partial t} + \nabla\cdot\rho U c_p T = \lambda \nabla^2 T. \tag{7}$$

The IAMR algorithm uses a hierarchical grid structure, which changes dynamically in time, composed of rectangular, uniform grids of varying resolution. The collection of grids at a given resolution is referred to as a level, and is denoted by Λ^ℓ for $\ell \in \{0, 1, 2, \ldots\}$. Level 0 is the coarsest level and covers the entire problem domain. The widths of the cells in the level ℓ grids differ from those at $\ell+1$ by a even integer factor r_ℓ called the refinement ratio; r_ℓ is typically 2 or 4. In space, the levels are properly-nested, i.e., there must always be a region at least one cell wide at level $\ell+1$ separating two levels ℓ and $\ell+2$. (For convenience in structuring the computation, coarser levels extend under the finer levels that cover them, but values in these obscured regions are always overridden by finer level information as it becomes available.)

(It is useful to define at this point to define a level projection operator $\mathbf{P}$ for later use in the paper. $\mathbf{P}(\Lambda^{\ell+1})$ is the collection of cells at level ℓ covered by cells at level $\ell+1$; $\Lambda^\ell - \mathbf{P}(\Lambda^{\ell+1})$ is the portion of level ℓ that is not covered. Similarly, $\mathbf{P}(\partial\Lambda^{\ell+1})$ is the collection of level ℓ cell edges covered by $\partial\Lambda^{\ell+1}$, the border of level $\ell+1$.)

Most quantities are stored at cell-centers of the mesh, while fluxes between cells are represented as being on cell edges (faces in three dimensions). Both Cartesian (two- and three-dimensional) and axisymmetric coordinates are currently supported, but for simplicity most of the discussion in this paper is confined to two-dimensional Cartesian coordinates.

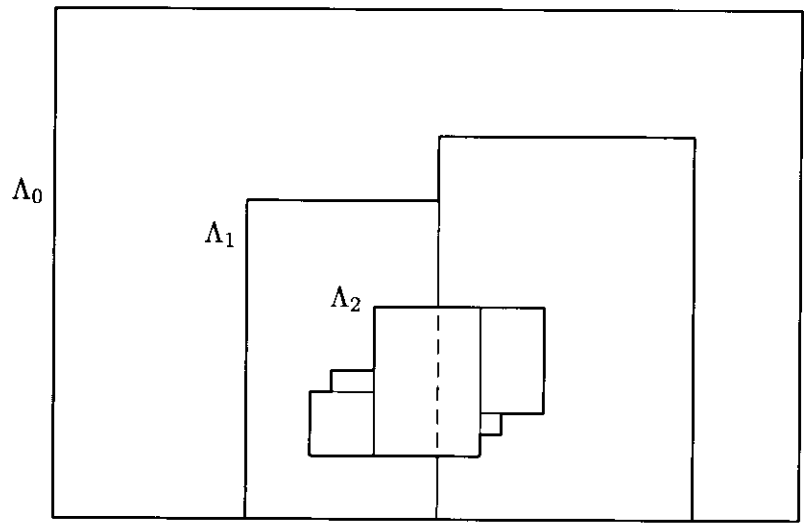

Figure 1: A properly nested hierarchy of grids

Two features of the IAMR algorithm must be outlined here in order to show how the radiative transfer scheme fits into it. These are the single-level timestep and the adaptive-mesh timestep. At a single level ℓ, the basic IAMR timestep to advance the solution from time t^n to time $t^n + \Delta t = t^{n+1}$ proceeds as follows:

1. A second-order Godunov method [14] uses state information from cell centers at time n to compute U at cell faces at time $n + 1/2$. Upwinding rules are applied to information on faces.

2. A MAC-style projection is applied to the face-based velocities to compute the advection velocities U^{ADV} at all cell edges. With this feature the scheme is both conservative and free-stream preserving for advected scalars. This projection requires solution of an elliptic equation across the level.

3. A second-order Godunov method uses the MAC-projected edge velocities to compute edge values of U and T at $t^{n+1/2}$.

4. The convective derivatives $(U \cdot \nabla U)_{ij}^{n+1/2}$ and $(\nabla \cdot UT)_{ij}^{n+1/2}$ are computed at cell centers at time time $n + 1/2$ using the values of U^{ADV} and the edge values of U and T.

5. The solution of the following Crank-Nicholson difference equations for (7) and (1) computes T and U at time $n + 1$:

$$\rho c_p \left(\frac{T_{ij}^{n+1} - T_{ij}^n}{\Delta t} + (\nabla \cdot UT)_{ij}^{n+1/2} \right) = \frac{\lambda}{2} \left(\left(\nabla^2 T\right)_{ij}^n + \left(\nabla^2 T\right)_{ij}^{n+1} \right) \tag{8}$$

$$\rho \left(\frac{U_{ij}^{n+1} - U_{ij}^n}{\Delta t} + (U \cdot \nabla U)_{ij}^{n+1/2} \right) = \frac{\mu}{2} \left(\left(\nabla^2 U\right)_{ij}^n + \left(\nabla^2 U\right)_{ij}^{n+1} \right) - \frac{(\nabla p)_{ij}^{n-1/2}}{\rho} \tag{9}$$

6. The velocity computed in step 5 does not necessarily satisfy (3). A node-based projection [15] is applied, then, to correct the new-time velocity divergence and to update the pressure to time $n + 1/2$.

The spatially implicit finite difference equations that arise in the MAC projection, the Crank-Nicholson differencing steps, and in the nodal projection are solved with multigrid techniques [16]. The cell-centered solves use V-cycles with red-black Gauss-Seidel relaxation and conjugate gradient at the bottom of the V-cycle. The nodal solve uses a similar approach. The details of the discretizations of the elliptic operators and of the pressure gradient are discussed in [11]. Note that the MAC projection, the Crank-Nicholson solves, and the nodal projection must be done on all grids in a level simultaneously.

On the full adaptive mesh, an IAMR timestep consists of separate timesteps on each of the levels, plus synchronization operations to insure correct behavior at the coarse-fine interfaces, plus regridding operations which permit the refined grids to track complex and/or interesting regions of the flow. The ratio of the level ℓ and the level $\ell + 1$ time steps is r_ℓ, so that the Courant number is roughly the same on all levels. Figure 2 shows a space/time diagram of a single coarse (level 0) timestep, during which a regridding operation moves the interface between levels 1 and 2. The timestep is a recursive procedure which proceeds as follows on level ℓ:

1. Advance level ℓ, using boundary information from level $\ell - 1$ as needed but ignoring levels $\ell + 1$ and higher.

2. Advance level $\ell + 1$ r_ℓ times. (This involves advancing levels $\ell + 2$ and higher, recursively.)

3. Synchronize levels ℓ and $\ell + 1$.

4. If the appropriate regridding interval has passed, tag cells at level ℓ that require refinement according to some predefined user criteria, determine the new level $\ell + 1$ grids to cover this region, and transfer data to the new grids (using conservative interpolation from level ℓ if necessary).

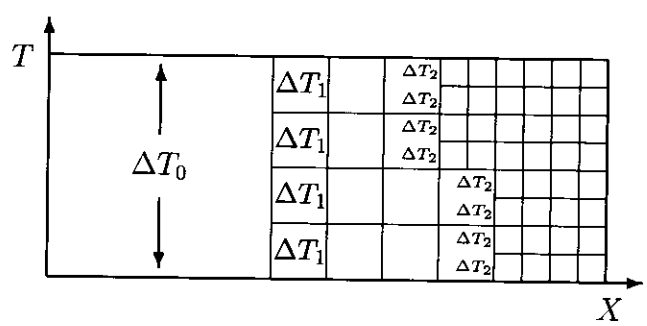

Figure 2: Multilevel timestep structure

The synchronization operations serve in part to ensure that the overall timestepping scheme conserves enthalpy. The operations are otherwise complex and mostly irrelevant to

the unsteady radiative transfer algorithm. The only part we describe here is the *refluxing* operation for the advective terms: During a coarse timestep, both coarse and fine grids compute advective fluxes on the faces that make up the coarse-fine interface. Since these fluxes do not match exactly, it is necessary to adjust the flux computed on the coarse grid to bring it into agreement with the integrated fluxes from the fine grid in order to insure conservation. For advection, it is then sufficient to adjust the state in coarse cells bordering a fine grid to account for the new flux information. The key data structure for refluxing is an object called the *flux register*, that exists at coarse-fine interfaces at the same resolution as the coarse grid.

Though the synchronization operations for the elliptic and parabolic portions of the algorithm are more complicated, they also depend on flux registers for storing information about various imbalances between coarse and fine grids. We likewise use these data structures for storing information relevant to energy conservation in the radiative transfer algorithm.

AMR Discrete Ordinates for Steady Radiative Transport

We now summarize the AMR discrete ordinates algorithm for solving instantaneous or steady-state radiative transfer problems presented in [9].

The instantaneous radiation field satisfies the radiative transfer equation (RTE) (4). The discrete ordinate method is based on a semi-discrete form of this equation,

$$(\mathbf{\Omega}_m \cdot \nabla) I_m + (\kappa + \sigma) I_m = \kappa I_b + \frac{\sigma}{4\pi} \sum_{m'} w_{m'} I_{m'}, \tag{10}$$

where the continuous dependence of the radiative intensity I on angle is reduced to consideration only of the intensity in the discrete directions $\mathbf{\Omega}_m$. The ordinate weights w_m satisfy $\sum w_m = 4\pi$. The corresponding discretization of (5) is

$$\nabla \cdot \mathbf{q}_r = \sum_m w_m \kappa (I_b - I_m) = 4\pi\kappa I_b - \sum w_m \kappa I_m. \tag{11}$$

Discretizing (10) conservatively over cells gives, in Cartesian coordinates,

$$\frac{\mu_m}{\Delta x}(I_{m,i+1/2,j} - I_{m,i-1/2,j}) + \frac{\xi_m}{\Delta y}(I_{m,i,j+1/2} - I_{m,i,j-1/2}) + (\kappa + \sigma) I_{m,ij} = S_m(I), \tag{12}$$

where the emission and scattering sources have been combined into a single source term $S_m(I)$. The system can be closed by specifying a relationship between the cell and the edge values, such as the diamond-difference or step approximation, but the multilevel algorithm does not depend on the details of this relationship. Finally, Eq. (6) is discretized to provide a physical boundary condition at exterior faces, combining the effects of emission and diffuse reflection at the walls:

$$I_m = \epsilon I_b^{\text{wall}} + \frac{\rho}{\pi} \sum_{\hat{n}\cdot\mathbf{\Omega}_{m'}>0} w_{m'} \left(\hat{n} \cdot \mathbf{\Omega}_{m'}\right) I_{m'}, \quad \hat{n} \cdot \mathbf{\Omega}_m < 0. \tag{13}$$

Note that the intensities I_m in different ordinate directions are coupled only through the scattering term and through reflection at the boundary. Eqs. (10) and (13) are solved by repeatedly sweeping through the grid for all ordinate directions. For each ordinate the sweep proceeds in the direction the radiation is moving. After each sweep the reflection and scattering sources are updated, and the iteration is continued until the system converges. If scattering is isotropic the source $S_m(I)$ is independent of direction and it is not necessary to maintain storage for all ordinate directions at once. (It is convenient, though, to allocate storage for edge fluxes at the boundary for all ordinate directions.)

So far we have described the discrete ordinates method as it has been presented in many earlier sources [6, 7, 8]. The contribution of [9, 10] was to extend this method to the adaptive grid structure described above. In each uniform region Λ_ℓ of the adaptive mesh, the solution satisfies equation (12) in the interior and the boundary condition (13) at the walls. At a coarse-fine interface $\partial\Lambda^{\ell+1}$ we require the coarse flux to be the average of the fine fluxes across each face:

$$I_m^\ell = \langle I_m^{\ell+1} \rangle \quad \text{on } \mathbf{P}(\partial\Lambda^{\ell+1}). \tag{14}$$

There are two new features added to the computation in order to deal with the adaptive mesh. One is the sequencing of grids on each individual level, the other is the method of cycling between levels to obtain a converged solution. The single-level algorithm is just an elaboration of that for a single grid: transport sweeps are performed for each ordinate in the direction of propagation. As shown in Figure 3, this imposes an ordering on the grids, since some grids must be solved before others. In two dimensions, the necessary orderings always exist, and to handle all ordinate directions a total of four orderings are required (one for transport towards the upper right, one towards the lower right, and the reversals of these for transport to the left).

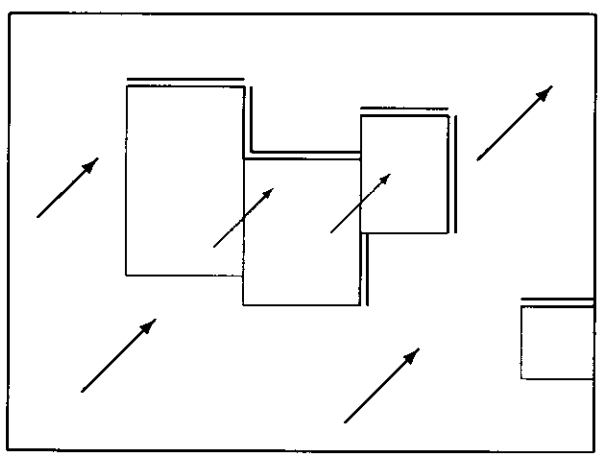

Figure 3: Transport sweeps must be performed on some grids before others. Double lines show locations of flux registers.

In three dimensions, however, there are cases where an ordering is not possible (this complication was not mentioned in [9, 10]). Figure 4 shows one such case. For radiation

transport in the increasing x, y and z directions, grid A must precede B, B must precede C, and C must precede A. We resolve the difficulty by dividing grids into pieces. The situation shown in the figure can be resolved by dividing any of the three grids into two parts; the resulting four grids can then be ordered appropriately. A total of eight orderings are required in three dimensions, of which four are reversals of the other four.

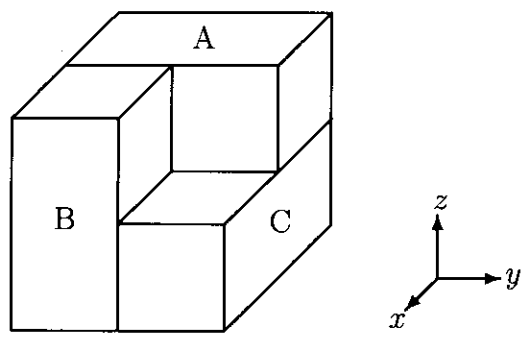

Figure 4: Grid ordering is not always possible in three dimensions. Some grids must be divided into pieces until a complete ordering is obtained.

The situation becomes more complicated when multiple levels are involved. Radiation passes from coarse to fine grids on the upstream edges of the fine grids, and from fine to coarse grids on the downstream edges. We compute the composite solution across all levels by again iterating a transport process until the system converges. The process now covers all active levels, however, in sequence from coarse to fine and back again.

The details of the multilevel iteration are not as important as the form of the resulting solution. We obtain at each level ℓ a solution I_m^ℓ satisfying (10) on the the exposed portion of the level $\Lambda^\ell - \mathbf{P}(\Lambda^{\ell+1})$, satisfying (13) on physical boundaries, and satisfying (14) on $\mathbf{P}(\partial\Lambda^\ell)$ and $\mathbf{P}(\partial\Lambda^{\ell+1})$. On interfaces where radiation passes from a coarse grid to a fine grid we enforce (14) trivially by setting $I_m^{\ell+1} := I_m^\ell$. Transport sweeps across a level do not stop when a finer level is encountered, however. There is therefore an extended solution $\hat{I}_m^\ell$, which agrees with I_m^ℓ in the exposed region where that exists, but is also defined in the region $\mathbf{P}(\Lambda^{\ell+1})$ under the finer level.

This extended solution obeys an equation

$$(\mathbf{\Omega}_m \cdot \nabla)\hat{I}_m + (\kappa + \sigma)\hat{I}_m = S_m(\hat{I}) - D_R(\mathbf{\Omega}_m \delta I_m^{\ell+1}) \tag{15}$$

over all of Λ^ℓ. The final term in this equation can be thought of as feeding fine grid information back into the coarse grid. $\delta I_m^{\ell+1}$ is accumulated in flux registers and is defined as

$$\delta I_m^{\ell+1} = \langle I_m^{\ell+1} \rangle - \hat{I}_m^\ell \quad \text{on } \mathbf{P}(\partial\Lambda^{\ell+1}). \tag{16}$$

(Note that $\delta I_m^{\ell+1} = 0$ along the upstream faces where radiation enters the fine grid.) D_R is the "reflux divergence" operator, which returns the divergence of its argument on the coarse cells bordering $\partial\Lambda^{\ell+1}$. The effect of this term is to enforce equation (14) along the downstream faces where radiation passes from the fine to the coarse grid.

Single Grid Algorithm for Unsteady Combined Mode Heat Transfer

A single grid projection method for low-Mach number combustion is described in [12]. The single grid method is a fractional step scheme in which we first advance the evolution equations of reacting flow and then solve an elliptic equation to correct the velocity and update pressure. In order that the method be second-order accurate in time for nonlinear differential equations with source terms, a sequential, predictor-corrector treatment of the evolution equations is used. The sequential approach ensures that all implicit finite difference equations are linear, while the predictor-corrector formulation guarantees second-order accuracy in time. The discrete-ordinates method is used for radiative transport.

We outline below the application of this algorithm to Eqs. (1)–(5). The basic steps are similar to the level solve for convective heat transfer described above. The main differences are the incorporation of the discrete ordinates method and the predictor-corrector treatment of Eq. (2) in order to time center the sources due to radiative transport:

1. The convective derivatives $U \cdot \nabla U$ and $\nabla \cdot UT$ at cell centers at time time $n + 1/2$ are computed as described above.

2. The difference equations

$$\rho c_p \left(\frac{T_{ij}^{n+1,p} - T_{ij}^n}{\Delta t} + (\nabla \cdot UT)_{ij}^{n+1/2} \right) = \frac{\lambda}{2} \left(\left(\nabla^2 T \right)_{ij}^n + \left(\nabla^2 T \right)_{ij}^{n+1,p} \right) - (\nabla \cdot \mathbf{q}_r)^n \tag{17}$$

   are solved for the predicted temperature $T_{ij}^{n+1,p}$.

3. $(\nabla \cdot \mathbf{q}_r)_{ij}^{n+1}$ is computed using the discrete ordinates method for the temperature field $T^{n+1,p}$.

4. The difference equations

$$\rho c_p \left(\frac{T_{ij}^{n+1} - T_{ij}^n}{\Delta t} + (\nabla \cdot UT)_{ij}^{n+1/2} \right) = \frac{\lambda}{2} \left(\nabla^2 T_{ij}^n + \nabla^2 T_{ij}^{n+1} \right) - \frac{1}{2} \left((\nabla \cdot \mathbf{q}_r)_{ij}^n + (\nabla \cdot \mathbf{q}_r)_{ij}^{n+1} \right) \tag{18}$$

   are solved for T_{ij}^{n+1}.

5. The Crank-Nicholson difference equations (9) are solved to compute U at time $n+1$.

6. A projection is applied to correct the new-time velocity divergence and to update the pressure.

Note that $(\nabla \cdot \mathbf{q}_r)^n$ is the divergence of the radiative flux computed in the previous time step using $T^{n,p}$. $(\nabla \cdot \mathbf{q}_r)^0$ is computed using T^0.

AMR for Time-dependent Radiative Transfer

The AMR algorithm for unsteady combined-mode heat transfer differs from the convective transport algorithm in two significant ways. The first is simply that the sequence of steps to advance a single level follows the same sequence of steps for the single grid algorithm presented above. Note that in the single level advance, the diffusive temperature fluxes $\lambda \nabla T$ at coarse-fine interfaces used in the synchronization step are stored only during the temperature correction step. The second difference entails the incorporation of the discrete ordinates algorithm. The remainder of this subsection focuses on this topic.

For a time step in a single level calculation, i.e., one in which $\ell_{max} = 0$, we solve for I_m at times n and $n+1$, compute $(\nabla \cdot \mathbf{q}_r)^n$ and $(\nabla \cdot \mathbf{q}_r)^{n+1}$, and then update T in a time-centered fashion by solving (18) so that the scheme is second-order in time. Note that this update is automatically conservative, since it involves a simple integration in time of a conservative radiative transport solution for the entire domain. In the multilevel adaptive algorithm, however, the domain is not all advanced at a single time step, and imbalances in the radiative energy field develop along level interfaces which must be stored and redistributed properly in order for the scheme to be conservative.

The multilevel algorithm is specified recursively by detailing the interactions of a single level ℓ with the levels both above and below in the hierarchy, as it advances from time index n to $n+1$. (In our description, we do not distinguish time indices on other levels, though we point out that many finer-level time steps are involved.) Before we provide the equations, though, it may be helpful to describe the basic form of the algorithm and the roles played by the important variables.

We begin a coarse time step by computing a multilevel solution I_m^n across all the active levels. Note that this defines $(\nabla \cdot \mathbf{q}_r)^n$ at level ℓ and all finer levels. We then advance the coarse level. We first compute a predicted temperature $T^{n+1,p}$ by solving (17) on level ℓ. We next compute a trial solution $\check{I}_m^{n+1}$ at the new time, approximating the effect of the missing finer levels on the coarse region $\Lambda^\ell - \mathbf{P}(\Lambda^{\ell+1})$ by reusing the reflux contribution from time n. We find $(\nabla \cdot \mathbf{q}_r)^{n+1}$ using the values of $\check{I}_m^n$ and compute T^{n+1} by solving (18).

The coarse cells in the fine region $\mathbf{P}(\Lambda^{\ell+1})$ are eventually overwritten by finer level data. Changes to the fluid state in this region affects cells outside the coarse-fine interface, however, so we also modify the enthalpy update in $\mathbf{P}(\Lambda^{\ell+1})$ with an estimate of the effects of levels $\ell+1$ and above. A quantity $\dot{E}^{\ell+1}$ is computed for this purpose from the time n multilevel solution, and is used in the coarse enthalpy update. These attempts to anticipate the effects of the finer levels are used only to improve accuracy—none are needed for conservation, as that is provided by the synchronization procedure at the end of the coarse time step. Note that by definition, $\dot{E}^{\ell+1} = 0$ on $\Lambda^\ell - \mathbf{P}(\Lambda^{\ell+1})$ and on $\Lambda^{l_{max}}$.

Once the trial solution to level ℓ is in place at time $n+1$, level $\ell+1$ is advanced through r cycles to bring it up to the same time. (Recall that this procedure in turn involves advancing and regridding still finer levels, if any.) Boundary conditions for level $\ell+1$ are obtained by interpolating the level ℓ solution in both space and time. For the radiative intensity I_m we use piecewise linear interpolation in time and piecewise constant interpolation in space, so the fine flux at each incoming face is set to the time-weighted average of the fluxes at the corresponding coarse faces. Outgoing fluxes at the coarse-fine interface are accumulated in

flux registers for use in the synchronization step.

At the end of a coarse time step we perform the radiation synchronization step. (This step follows all the other AMR synchronizations.) We compute two multilevel solutions. One is simply the new-time radiative transfer solution I_m^{n+1}, which is also used as the baseline solution for the next time step. The other is computed with the accumulated coarse-fine flux mismatch used as an additional source term. The difference between these solutions is applied to all active levels to insure conservation of energy. (The correction is computed in this manner because it can be negative as well as positive, but the method we use to solve the discrete ordinates equations on each grid has a limiter that disallows negative fluxes.)

When more than two levels begin or end together, the multilevel computations are combined. Consider, for example, a calculation with two levels of refinement. The end of a level 1 timestep requires synchronization of levels 1 and 2. If the level 0 step is also completed, however, we perform a single synchronization operation for levels 0, 1 and 2 together. Note that a synchronization of levels 1 and 2 also contributes an increment to the flux registers at the 0-1 interface, which will be part of the source at the next 0-1-2 synchronization.

We are now ready to formally specify the details of the radiative transfer algorithm for advancing a level ℓ. Note that the multilevel solve at the initialization step has to be performed only if level ℓ was regridded just prior to the beginning of the time step, and then only if it has not already been done for this time by a still coarser level than the current one. This is checked by comparing the time stamps for the current level to that of the next coarser one. Also note that the synchronization step does not have to be done if it is going to be done by a still coarser level that has been updated to the same time. Again, this is checked by comparing time stamps.

If ($\ell = 0$ or $t^{\text{old},\ell} > t^{\text{old},\ell-1}$) **then**

Initialization from coarsest to finest level at current time, levels $\ell' \in \{\ell, \ldots, \ell_{\max}\}$:

For $\ell' \in \{\ell_{\max} - 1, \ldots, \ell\}$ **do**

- $I_b^{\ell',n} := \langle I_b^{\ell'+1,n} \rangle$ on $\mathbf{P}(\Lambda^{\ell'+1})$

Enddo

Multilevel Solve: $\ell' \in \{\ell, \ldots, \ell_{\max}\}$

(Omit this computation if a valid solution is already available from the end of the previous time step.)

- $(\mathbf{\Omega}_m \cdot \nabla) I_m^{\ell',n} = \kappa I_b^{\ell',n} - (\kappa + \sigma) I_m^{\ell',n} + S_m(I^{\ell',n})$ on $\Lambda^{\ell'} - \mathbf{P}(\Lambda^{\ell'+1})$
- $I_m^{\ell',n} = \langle I_m^{\ell'+1,n} \rangle$ on $\mathbf{P}(\partial\Lambda^{\ell'+1})$, $\ell' < \ell_{\max}$

End

For $\ell' \in \{\ell, \ldots, \ell_{\max} - 1\}$ **do**

- Derive $(\mathbf{\Omega}_m \cdot \nabla) \hat{I}_m^{\ell',n} = \kappa I_b^{\ell',n} - (\kappa + \sigma) \hat{I}_m^{\ell',n} + S_m(\hat{I}^{\ell',n}) - D_R(\mathbf{\Omega}_m \delta I_m^{\ell'+1,n})$ on $\Lambda^{\ell'}$
- $\delta I_m^{\ell'+1,n} := \langle I_m^{\ell'+1,n} \rangle - \hat{I}_m^{\ell',n}$ on $\mathbf{P}(\partial\Lambda^{\ell'+1})$

- $\dot{E}^{\ell'+1,n} := \sum_m w_m \kappa(\langle I_m^{\ell'+1,n}\rangle - \hat{I}_m^{\ell',n})$ on $\mathbf{P}(\Lambda^{\ell'+1})$
- Compute $(\nabla \cdot \mathbf{q}_r)^n$ using $\hat{I}_m^{\ell',n}$

Enddo

Endif

Time step, level ℓ:

- Compute the predicted temperature $T^{n+1,p}$ by solving

$$\rho c_p \left(\frac{T_{ij}^{n+1,p} - T_{ij}^n}{\Delta t} + (\nabla \cdot UT)_{ij}^{n+1/2} \right) = \frac{\lambda}{2} \left(\left(\nabla^2 T\right)_{ij}^n + \left(\nabla^2 T\right)_{ij}^{n+1,p} \right) - (\nabla \cdot \mathbf{q}_r)^n + \dot{E}^{\ell'+1,n}$$

- Using $T^{n+1,p}$, solve
  $(\mathbf{\Omega}_m \cdot \nabla)\check{I}_m^{\ell,n+1} = \kappa I_b^{\ell,n+1} - (\kappa+\sigma)\check{I}_m^{\ell,n+1} + S_m(\check{I}^{\ell,n+1}) - D_R(\mathbf{\Omega}_m \delta I_m^{\ell+1,n})$ on Λ^ℓ
- $\bar{I}_m^{\ell,n+1/2} := 1/2(\hat{I}_m^{\ell,n} + \check{I}_m^{\ell,n+1})$
- Compute $(\nabla \cdot \mathbf{q}_r)^{n+1}$ using $\check{I}_m^{\ell,n+1}$
- Compute the corrected temperature T^{n+1} by solving

$$\rho c_p \left(\frac{T_{ij}^{n+1,p} - T_{ij}^n}{\Delta t} + (\nabla \cdot UT)_{ij}^{n+1/2} \right) = \frac{\lambda}{2} \left(\left(\nabla^2 T\right)_{ij}^n + \left(\nabla^2 T\right)_{ij}^{n+1,p} \right) - \frac{1}{2} \left((\nabla \cdot \mathbf{q}_r)^n + (\nabla \cdot \mathbf{q}_r)^{n+1} \right) + \dot{E}^{\ell'+1,n}$$

- Advance levels $\ell+1, \ldots, \ell_{\max}$.
- $\delta \bar{I}_m^{\ell+1,n+1/2} := \langle \langle \bar{I}_m^{\ell+1} \rangle \rangle - \bar{I}_m^{\ell,n+1/2}$ on $\mathbf{P}(\partial\Lambda^{\ell+1})$

If ($\ell = 0$ or $t^{\text{new},\ell} < t^{\text{new},\ell-1}$) **then**

Synchronization/refluxing, levels $\ell' \in \{\ell, \ldots, \ell_{\max}\}$:

For $\ell' \in \{\ell_{\max}-1, \ldots, \ell\}$ **do**

- $I_b^{\ell',n+1} := \langle I_b^{\ell'+1,n+1} \rangle$ on $\mathbf{P}(\Lambda^{\ell'+1})$
- $R_m^{n+1} := -D_R(\mathbf{\Omega}_m(\delta \bar{I}_m^{\ell'+1,n+1/2} - \delta I_m^{\ell'+1,n})) \cdot (\Delta t^{\ell'}/\Delta t^\ell)$ on $\mathbf{P}(\partial\Lambda^{\ell'+1})$

Enddo

Multilevel Solve: $\ell' \in \{\ell, \ldots, \ell_{\max}\}$

- $(\mathbf{\Omega}_m \cdot \nabla)\tilde{I}_m^{\ell',n+1} = \kappa I_b^{\ell',n+1} - (\kappa+\sigma)\tilde{I}_m^{\ell',n+1} + S_m(\tilde{I}^{\ell',n+1}) + R_m^{n+1}$ on $\Lambda^{\ell'} - \mathbf{P}(\Lambda^{\ell'+1})$
- $\tilde{I}_m^{\ell',n+1} = \langle \tilde{I}_m^{\ell'+1,n+1} \rangle$ on $\mathbf{P}(\partial\Lambda^{\ell'+1})$, $\ell' < \ell_{\max}$

End

Multilevel Solve: $\ell' \in \{\ell, \ldots, \ell_{\max}\}$

- $(\mathbf{\Omega}_m \cdot \nabla) I_m^{\ell',n+1} = \kappa I_b^{\ell',n+1} - (\kappa + \sigma) I_m^{\ell',n+1} + S_m(I^{\ell',n+1})$ on $\Lambda^{\ell'} - \mathbf{P}(\Lambda^{\ell'+1})$
- $I_m^{\ell',n+1} = \langle I_m^{\ell'+1,n+1} \rangle$ on $\mathbf{P}(\partial\Lambda^{\ell'+1})$, $\ell' < \ell_{\max}$

End

For $\ell' \in \{\ell_{\max} - 1, \ldots, \ell\}$ **do**

- $(\rho c_p T)^{\ell',n+1} := (\rho c_p T)^{\ell',n+1} + \Delta t^{\ell} \sum_m w_m \kappa (\tilde{I}_m^{\ell',n+1} - I_m^{\ell',n+1})$ on $\Lambda^{\ell'} - \mathbf{P}(\Lambda^{\ell'+1})$
- $(\rho c_p T)^{\ell',n+1} := \langle (\rho c_p T)^{\ell'+1,n+1} \rangle$ on $\mathbf{P}(\Lambda^{\ell'+1})$

Enddo

- $\bar{I}_m^{\ell,n+1/2} := \bar{I}_m^{\ell,n+1/2} + (\tilde{I}_m^{\ell,n+1} - I_m^{\ell,n+1})$ on $\mathbf{P}(\partial\Lambda^{\ell})$, $\ell > 0$

Endif

Note that the final enthalpy update can be written in the form

$$\rho c_p T := \rho c_p T + \Delta t \sum_m w_m ((\mathbf{\Omega}_m \cdot \nabla) \tilde{I}_m - (\mathbf{\Omega}_m \cdot \nabla) I_m - R_m). \tag{19}$$

The first two terms in the sum are fluxes, and thus have no effect on global energy conservation. The final term (R_m) undoes the failure to conserve energy due to the mismatch of fluxes at the coarse-fine grid interfaces and the inclusion of the D_R term in the level solve.

The composite solution $\tilde{I}_m^{n+1}$ does not have a clear physical interpretation. We are only interested in its difference with I_m^{n+1}, which we compute in this way to avoid problems with the limiter. In the description above, there are three calls to the multilevel solver. As noted, however, the first call is necessary only if there if the grid structure has been changed, rendering the previous solution to I_m^n invalid.

Extensions to Variable Properties

We now outline the extension of the algorithm described above to allow for variable density, thermal diffusivity, and specific heat. We assume that $h = h(T)$ and $\lambda = \lambda(T)$. By definition, $c_p = h'(T)$. The governing equations are now (1), (3), the continuity equation, and the energy equation

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot \rho U h = \nabla \cdot \lambda \nabla T - \nabla \cdot \mathbf{q}_r. \tag{20}$$

We also consider the following non-conservative form of (20):

$$\rho c_p \left(\frac{\partial T}{\partial t} + U \cdot \nabla T \right) = \nabla \cdot \lambda \nabla T - \nabla \cdot \mathbf{q}_r. \tag{21}$$

The algorithm to update a single level can be summarized as follows. The convective derivatives $\nabla \cdot \rho U$, $\nabla \cdot \rho U h$, and $U \cdot \nabla T$ at computed at cell centers at $t^{n+1/2}$ with the second-order upwind scheme. ρ^{n+1} is then computed by an explicit convective update. We solve the Crank-Nicholson difference equations for (21) to compute the predicted temperature $T^{n+1,p}$;

time n values are used for ρ, c_p, λ, and $\nabla \cdot \mathbf{q}_r$ in this step. The predicted temperature is used to compute $(\nabla \cdot \mathbf{q}_r)^{n+1}$. We then compute $c_p^{n+1/2} = (c_p(T^n) + c_p(T^{n+1,p})/2$, $\rho^{n+1/2} = (\rho^n + \rho^{n+1})/2$, and $\lambda^{n+1} = \lambda(T^{n+1,p})$. We again solve the Crank-Nicholson equations for (21) again to recompute $T^{n+1,p}$; half-time values of ρ and c_p, and time n and $n+1$ values of λ and $\nabla \cdot \mathbf{q}_r$ are now used. We next compute h^{n+1} by applying an explicit Crank-Nicholson update to (20). Time n and $n+1$ values of ρ, λ, and $\nabla \cdot \mathbf{q}_r$ and time n and $n+1, p$ values of T are used. Finally, T^{n+1} is computed by solving $h^{n+1} = h(T)$ for T.

The motivation for this method is that conservation is expressed in terms of h, but boundary conditions for heat conduction are based directly on T. With variable c_p we must explicitly allow for the nonlinear conversions between these two quantities.

The radiation synchronization step follows the algorithm in the previous subsection with two modifications: h is used instead of c_pT, and at the end of the step, T is found by inverting $h = h(T)$.

The algorithm can be further extended for variable viscosity, gravity, multiple species, and compressibility. Since these extensions do not directly affect the treatment of radiative heat transfer, we omit this discussion; see [12, 13] for further details.

Numerical Examples

In this section we present five numerical examples computed using the adaptive mesh algorithm for unsteady, combined-mode heat transfer described above. We first show four sets of results to validate the algorithm: thermally developing axisymmetric pipe flow, thermally developing two-dimensional flow in a parallel plate channel, simultaneously developing axisymmetric pipe flow, and simultaneously developing three-dimensional flow in a square duct. In the fifth example, we compute a shearing flow in a closed box to demonstrate that the algorithm is stable and conservative with a complex adaptive grid structure.

In all the examples, the computational cells are square and a Courant number of 0.5 is used. Physical properties are uniform and constant in time within each example, except as noted. The results with radiation are obtained using the step approximation unless otherwise indicated. The first three examples use an S_6 ordinate set, while the last two use S_4. Both sets are tabulated in [17], and are chosen because they have correct half-range first moments and therefore conserve energy at the boundaries.

In the first four examples, we assume black walls, i.e., $\epsilon = 1$. With respect to radiation, symmetry boundary conditions are used at the pipe, channel, or duct exit, while the inlet is treated as a nonreflecting wall at the inflow temperature. In the fifth example the walls are all perfect diffuse reflectors ($\epsilon = 0$, $\rho = 1$), so the heat content of the box is constant in time.

Thermally Developing Laminar Pipe Flow

The first example is the axisymmetric calculation of heat transfer to fully developed laminar flow in the thermal entry region of a circular pipe (see Fig. 5). The initial and inlet axial velocities and temperature are $u = 2u_0(1 - (r/R)^2)$ and $T = T_0$, respectively. The

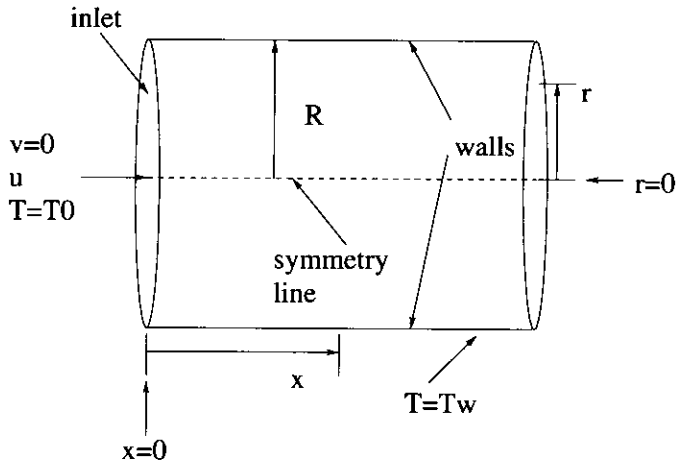

Figure 5: Specification and coordinate system for circular pipe.

temperature at the wall of the pipe is $T = T_w$. We compute the solution on the domain $0 \leq r \leq R$, $0 \leq x \leq L = .3R \cdot \mathrm{RePr} = 8R$. Symmetry boundary conditions are used at $r = 0$. For the results shown here. Re = 35.086, Pr = .76, $N = 1.149$, $\tau = 2.277$, and $T_w/T_0 = 1.000333$.

We use two analytical solutions as a basis for comparison. The first is the transient convective heat transfer solution due to Siegel [18]. This analysis is made under the assumption that axial heat conduction can be neglected; hence, we eliminate axial heat conduction from the calculations here as well. (Radiative heat transfer is not so restricted, however, and radial conduction is present in all cases.) The time dependent solution in [18] is found as an expansion about a steady-state solution, that of the Graetz problem [18, 19]. The analytic formulation is therefore correct only in the large time limit. In particular, at any given time there is a distance from the pipe inlet beyond which the temperature has evolved entirely by heat conduction from the walls. In this region, the analysis in [18] is only approximately correct. A better solution in this portion of the domain is the solution for conductive heat transfer in an infinite pipe [20].

We first show results from uniform grid calculations to establish a baseline for the adaptive results. Figs. 6 and 7 display the bulk temperature profiles resulting from computations on a uniform 64×512 grid at $\Theta = .05861$, .11721, .23442 and .46884. Both pure convection ($N = \infty$) and combined mode heat transfer results are shown. In Fig. 6, the exact solutions for pure convection and for conduction in an infinite pipe are shown as well. The computed $N = \infty$ solution does not quite match the analytic solution near the boundary of the pure conduction region. We believe this is due to numerical viscosity resulting from our convection scheme.

We next compare combined mode heat transfer results for five different adaptive grid hierarchies with the uniform grid results. The six cases, including the uniform grid case, are:

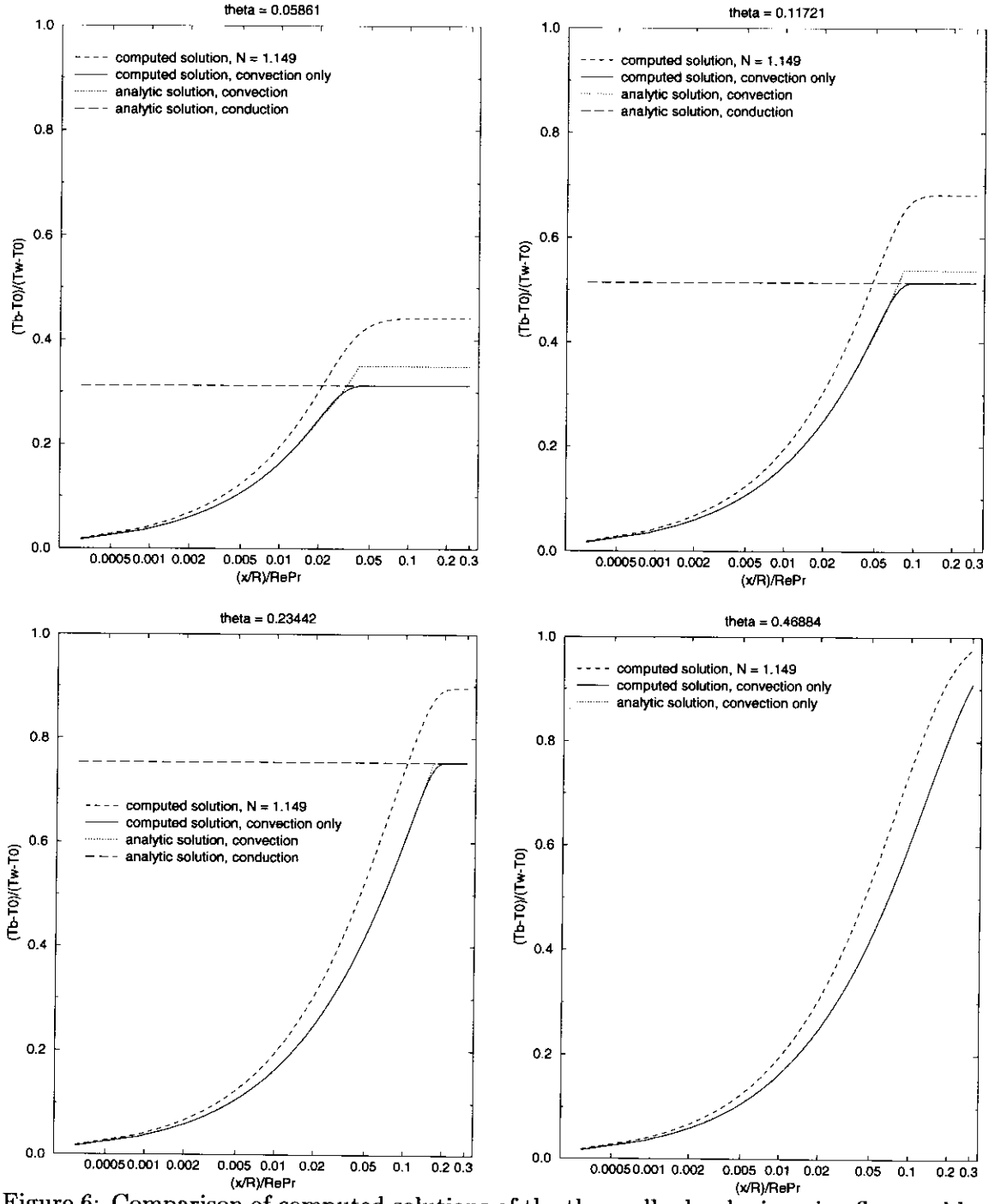

Figure 6: Comparison of computed solutions of the thermally developing pipe flow problem for $N = \infty$ (convection only) and $N = 1.149$ with analytic solutions for convection [18] and for conduction in an infinite pipe [20] at $\Theta = .05861$, .11721, .23442 and .46884. The solutions were computed on a uniform 64×512 grid. The conduction solution is not shown at $\Theta = .46884$ because the other three solutions are at steady-state by that time. The computed solutions are also displayed in Fig. 7.

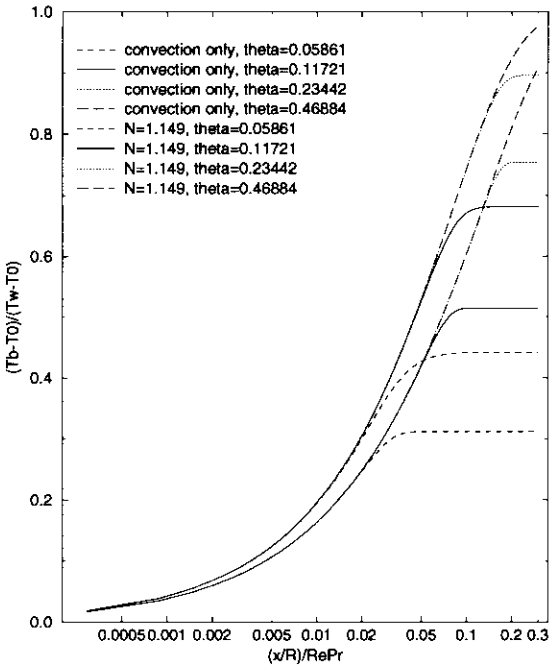

Figure 7: Computed solutions on a 64×512 uniform grid of the thermally developing pipe flow problem for convection only and $N = 1.149$ at $\Theta = .05861$, .11721, .23442 and .46884. These solutions are also shown separately in Fig. 6.

1) 64×512 level 0 grid, $\ell_{max} = 0$
2) 32×256 level 0 grid, $\ell_{max} = 1$, $r_0 = 2$
3) 16×128 level 0 grid, $\ell_{max} = 1$, $r_0 = 4$
4) 16×128 level 0 grid, $\ell_{max} = 2$, $r_0 = 2, r_1 = 2$
5) 8×64 level 0 grid, $\ell_{max} = 2$, $r_0 = 2, r_1 = 4$
6) 4×32 level 0 grid, $\ell_{max} = 2$, $r_0 = 4, r_1 = 4$.

The refinement strategy in cases 2–6 is to tag all cells for which $x/L < .0625$ and those cells for which $x/L < .375$ and $.3 \leq (T - T_0)/(T_w - T_0) \leq .7$. Note that with this strategy, the grids change in time as the temperature solution evolves. Note also that if the refinement strategy were instead to refine everywhere, the finest level would consist of a 64×512 grid covering the entire domain in all five adaptive cases.

Fig. 8 displays the bulk temperature at $\Theta = .05861$, .11721, .23442 and .46884 for all the cases except 4. The case 4 results are not shown because they are indistinguishable from those of case 3. At each time, the region where the profiles differ significantly coincides quite closely with the region where no refinement occurs, i.e., $(x/R)/RePr > .1125$. In other words, the discrepancies in the curves are due simply to differences in resolution in the unrefined region. Fig. 9 shows a time history of normalized temperature field for case 6.

We now present timings of the code for the six gridding strategies. The calculations were all run on a single 300 Mhz processor of a DEC Alpha workstation to a final time of $\Theta = .46884$. Table 1 shows the CPU time used to complete the calculation, the total number of cells advanced, the CPU time per cell, the CPU time per cell for radiation, the

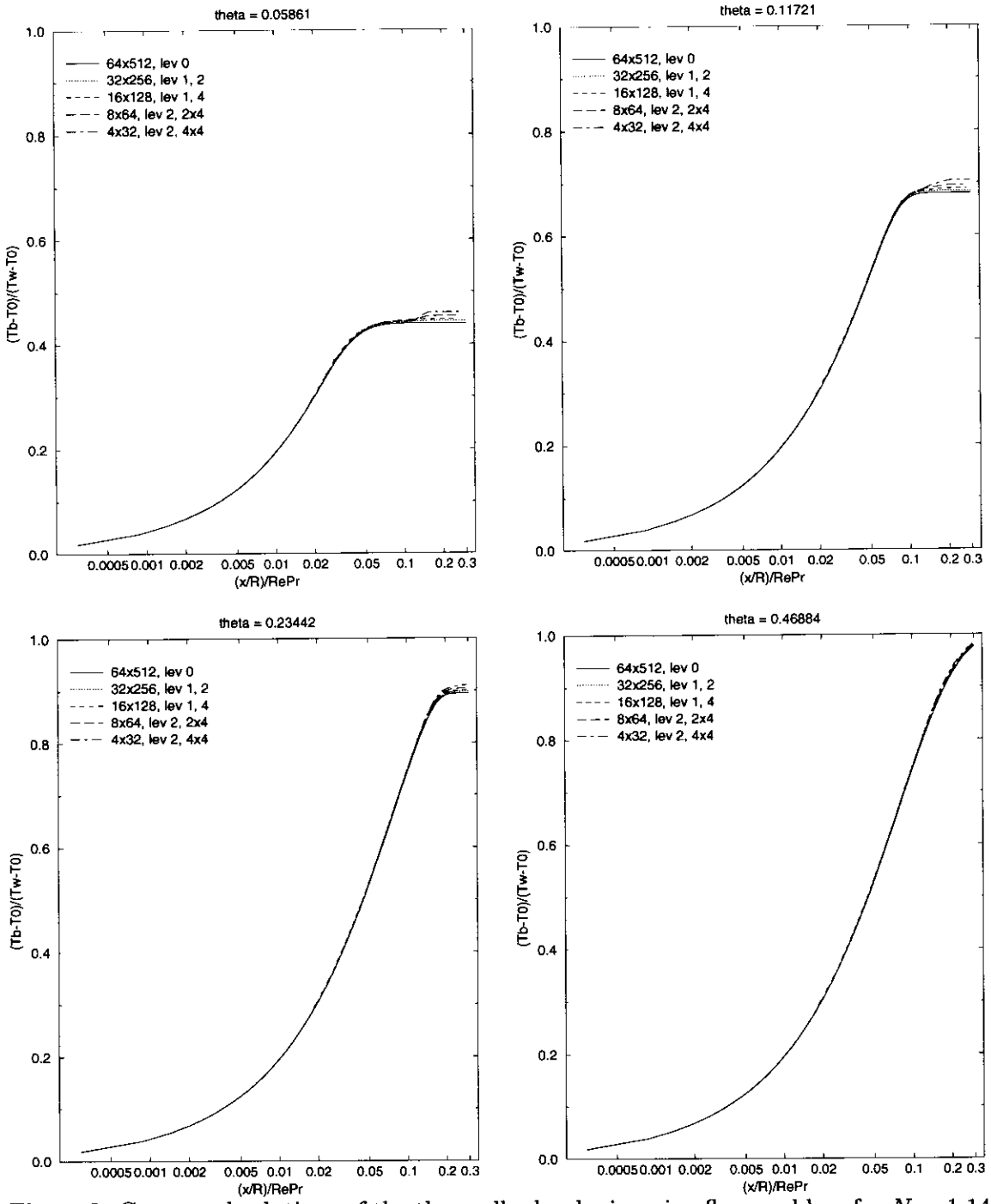

Figure 8: Computed solutions of the thermally developing pipe flow problem for $N = 1.149$ at $\Theta = .05861$, .11721, .23442 and .46884. The results for four different adaptive grid hierarchies are compared with uniform grid results.

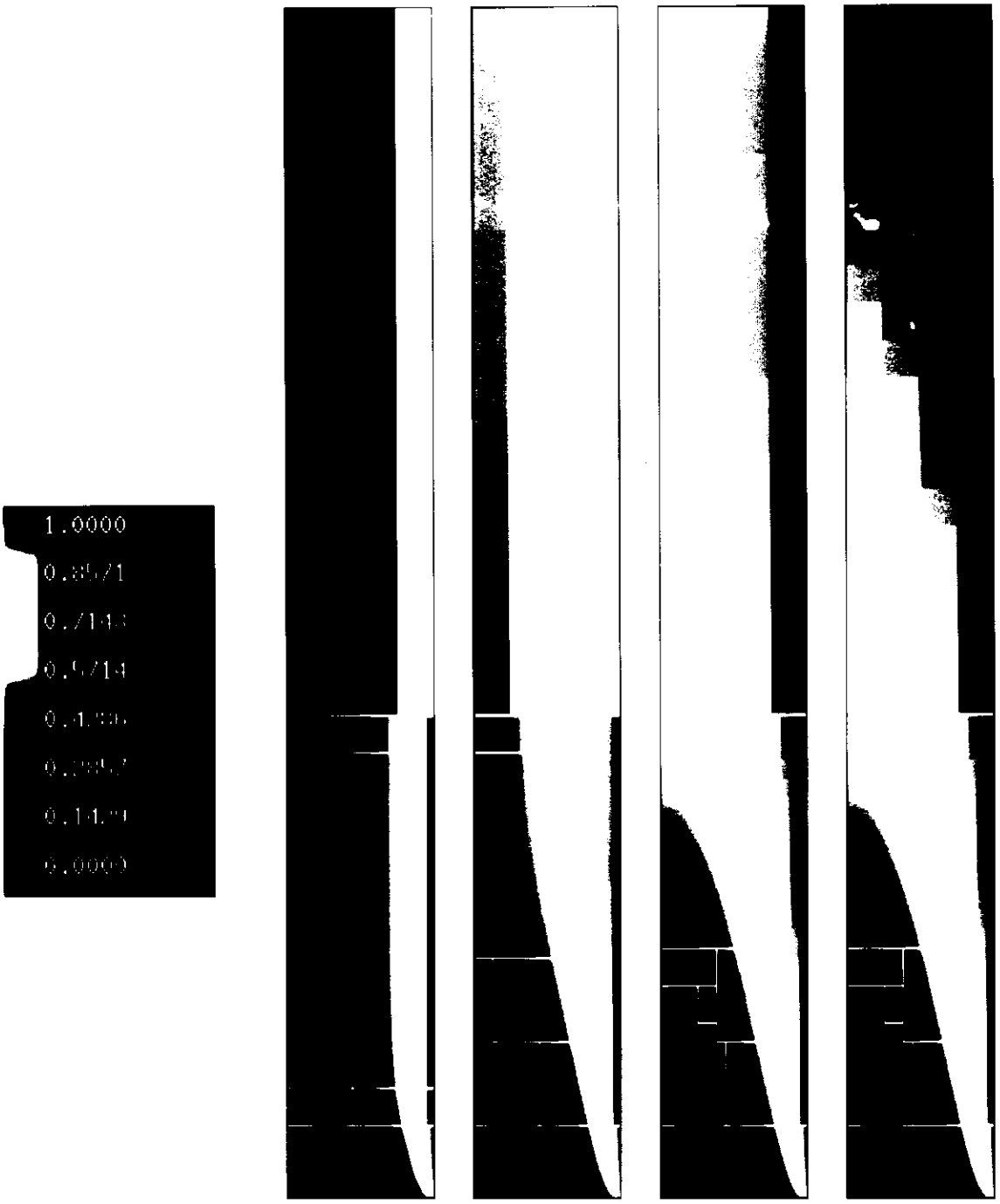

Figure 9: Computed normalized temperature $(T - T_0)/(T_w - T_0)$ at Θ = .05861, .11721, .23442 and .46884 for case 6 of the thermally developing pipe flow example. The boundaries of the level 1 and 2 grids are shown as thin lines.

Gridding			CPU Time		CPU Time for Radiation		Cells Advanced	Peak Memory Usage
Level 0 grid	ℓ_{max}	r	Total(s)	μs/cell	μs/cell	% CPU	Number	Mb
1) 64 × 512	0		22470.	429.	99.	22.9	52428800	31
2) 32 × 256	1	2	16770.	826.	233.	28.3	20308560	18
3) 16 × 128	2	2,2	12870.	759.	193.	25.5	16958800	16
4) 16 × 128	1	4	9797.	625.	117.	18.8	15679488	14
5) 8 × 64	2	2,4	9442.	619.	113.	18.3	15259632	13
6) 4 × 32	2	4,4	11950.	617.	116.	18.8	19367296	13

Table 1: Timings for uniform grid and refined grid calculations on a single processor of a DEC Alpha workstation for the thermally developing pipe flow problem.

Gridding			CPU Time	CPU Time for Multilevel Radiation Solves	
Level 0 grid	ℓ_{max}	r	Total(s)	Total (s)	% CPU
2) 32 × 256	1	2	16770.	3515.	20.0
3) 16 × 128	2	2,2	12870.	2317.	18.0
4) 16 × 128	1	4	9797.	930.	9.5
5) 8 × 64	2	2,4	9442.	831.	8.8
6) 4 × 32	2	4,4	11950.	992.	8.3

Table 2: Radiation timings for refined grid calculations on a single processor of a DEC Alpha workstation for the thermally developing pipe flow problem.

percentage CPU time for radiation, and the approximate peak memory usage. The total number of cells advanced is the sum over all levels of the number of cells advanced at that level. We note that the average percentage of the domain refined to the finest level is roughly the same—$21 \pm 1\%$—in all five cases. (This percentage is found in each case by dividing the total number of cells advanced at the finest level by the total number of cells which would have been advanced at the finest level if the entire problem domain were refined.)

The numbers in table 1 show that the adaptive mesh refinement scheme can reduce the computational cost in terms of both CPU time and memory usage. For this particular problem and refinement strategy, cases 4, 5, and 6 show the best reduction in computational cost over the uniform grid calculation. Table 2 shows the total CPU time, the CPU time for multilevel radiation solves, and the percentage of time spent on multilevel solves for radiation for the five adaptive cases. (Note that the total radiation time in table 1 and the multilevel radiation solve time reported in 2 are computed differently and should not be compared directly with each other.) In cases 4, 5, and 6 a substantially smaller proportion of time was spent on multilevel solves than in cases 2 and 3. The comparative advantage of the latter three cases is most likely due, then, to the fact that each uses relatively less CPU time on multilevel solves, which is in turn due to the use of one or more refinement ratios of 4. The AMR timestepping scheme requires two multilevel solves for every r fine time steps, so the overhead due to multilevel solves is smaller for cases 4, 5 and 6 than for cases 2 and 3.

Thermally Developing Laminar Flow in a Parallel Plate Channel

In the second example we consider the two-dimensional calculation of heat transfer to fully developed laminar flow in the thermal entry region of a parallel plate channel (see Fig. 10) in order to confirm that the algorithm is spatially and temporally second order accurate. The diamond differencing scheme is used here [21, 22] instead of the step scheme because the latter is only first order accurate. We note the diamond differencing scheme is unbounded and may lead to nonphysical over- and undershoots [17]. For the calculations performed here, however, we did not observe these problems.

(For the sake of brevity we confine our attention to the uniform grid case. Convergence studies of adaptive-mesh examples, e.g. [11], involve assumptions about regridding criteria that often differ from the way the methods are used in practice. In most realistic examples the portions of the calculation with large errors are confined to the finest level, so a single-level analysis is appropriate.)

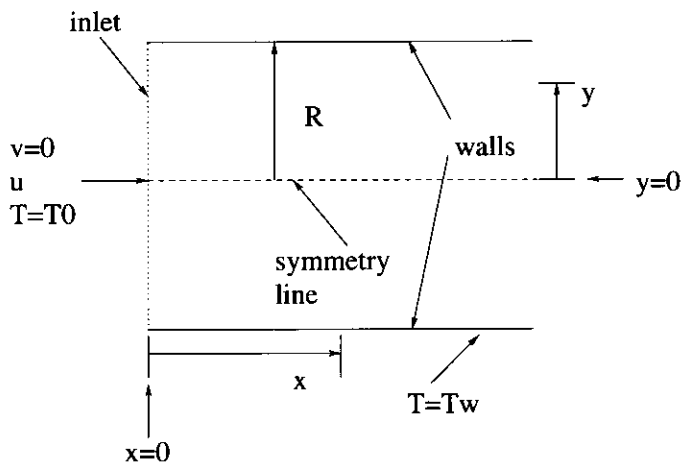

Figure 10: Specification and coordinate system for parallel plate channel.

The problem specification is the following. The initial and inlet axial velocities and temperature are $u = (3/2)u_0(1-(r/R)^2)$ and $T = T_0$, respectively. The temperature on the walls of the channel is $T = T_w$. We compute the solution on the domain $0 \le y \le R$, $0 \le x \le 8R$. Symmetry boundary conditions are used at $y = 0$. For the results shown here. Re $= 46.781$, Pr $= .76$, $N = 1.149$, $\tau = 2.277$, and $T_w/T_0 = 1.000333$.

We compute solutions on 4×32, 8×64, 16×128 and 32×256 uniform grids. The errors in the solution on the three coarser grids are computed at $\Theta = .05861$, $.11721$, $.23442$ and $.46884$. Because there is no exact solution, we estimate the error in the numerical solution by comparing solutions at successive resolutions. We first compute the error e_{ij} in a single coarse grid computational cell as the difference of the coarse grid result and the average of the solution in the overlying fine grid cells. The L_1 error on the entire coarse domain (assuming $\Delta x = \Delta y$) is then defined by

$$L_1^{\Delta x} = \sum_{ij} (\Delta x)^2 e_{ij}.$$

Θ	4×32	q	8×64	q	16×128
.05861	1.635×10^{-3}	2.01	4.057×10^{-4}	1.99	1.021×10^{-4}
.11721	1.127×10^{-3}	1.94	2.932×10^{-4}	1.96	7.495×10^{-5}
.23442	7.387×10^{-4}	1.92	1.951×10^{-4}	1.89	5.249×10^{-5}
.46884	5.305×10^{-4}	1.85	1.468×10^{-4}	1.84	4.091×10^{-5}

Table 3: L_1 errors and convergence rates for the parallel plate channel flow problem.

The convergence rate q can then be computed by comparing errors at successive resolutions:

$$q = \log_2 \left(L_1^{2\Delta x} / L_1^{\Delta x} \right).$$

The errors and convergence rates are shown in table 3. The results show that the numerical algorithm is second-order accurate for this problem.

Simultaneously Developing Laminar Pipe Flow

The third example is an axisymmetric computation of simultaneously developing laminar flow in the entry region of a pipe (see Fig. 5). The initial and inlet axial velocities and temperature are both uniform, $u = u_0$ and $T = T_0$, respectively. The temperature at the wall of the pipe is $T = T_w$. We compute the flow on the domain $0 \leq r \leq R$, $0 \leq x \leq L = .3R \cdot \mathrm{RePr} = 8R$. For the results shown here, Re = 35.086, Pr = .76, $N = 1$, $\tau = 1$ and $T_w/T_0 = 1.5$. The steady-state solution is found by timestepping until a time independent solution is achieved ($\Theta \approx .55$). We then compare this numerical solution with the steady-state boundary layer solution obtained by Pearce and Emery [23].

We compute results for the same six grid hierarchies described in the first subsection. For the multiple level calculations, the same refinement strategy is used as well. Fig. 11 compares the computed results at steady-state ($\Theta = .55$) for case 6 with the Pearce-Emery results; the bulk temperature curves for the other five cases are indistinguishable from the case 6 profile. The computed results and the Pearce-Emery predictions are in fairly good agreement. The differences may be due to the fact that Pearce and Emery make two approximations which we do not make. Specifically, in [23] the wall is considered to be an infinite tube at $T = T_w$, and axial temperature variations are neglected in the computation of $\nabla \cdot q_{rad}$.

Simultaneously Developing Three-Dimensional Duct Flow

The fourth example is a three-dimensional calculation of simultaneously developing laminar flow in the entry region of a square duct (see Fig. 12). The initial and inlet axial velocities and temperature are both uniform, $u = u_0$ and $T = T_0$, respectively. The temperature at the wall of the pipe is $T = T_w$. We compute the flow on the domain $0 \leq y,\ z \leq R$, $0 \leq x \leq L = .6R \cdot \mathrm{RePr} = 42R$. Symmetry boundary conditions are used at $y = 0$ and $z = 0$. For the results shown here, Re = 100, Pr = .7, $N = 100$, $\tau = 2$ and $T_0/T_w = .1$. The steady-state solution is found by timestepping until a time independent solution is

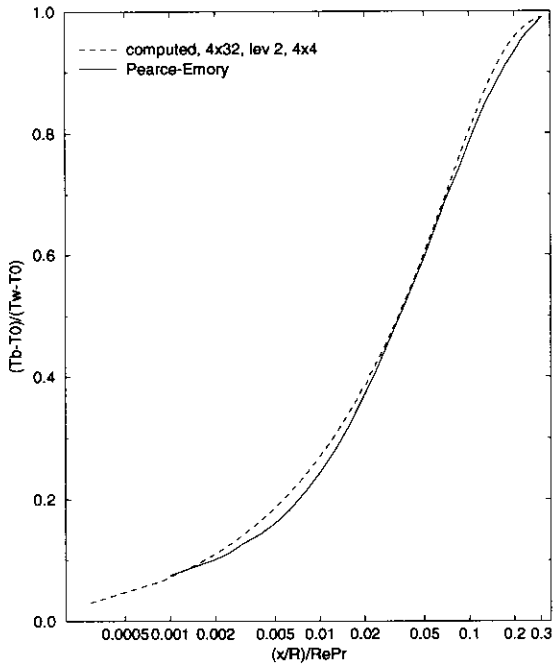

Figure 11: Comparison of the bulk temperature curve for case 6 of the simultaneously developing pipe flow example with the Pearce-Emery solution.

reached ($\Theta \approx .6745$). We then compare this numerical solution with a computation by a steady-state code implementing the algorithms in [17, 2].

We compute results for the following four cases:

1) $16 \times 16 \times 672$ level 0 grid, $\ell_{max} = 0$
2) $8 \times 8 \times 336$ level 0 grid, $\ell_{max} = 1$, $r_0 = 2$
3) $4 \times 8 \times 168$ level 0 grid, $\ell_{max} = 1$, $r_0 = 4$
4) $4 \times 4 \times 168$ level 0 grid, $\ell_{max} = 2$, $r_0 = 2, r_1 = 2$

The refinement strategy in cases 2–4 is to tag all cells for which $x/L < .07143$. Fig. 13 compares the computed results at steady-state ($\Theta = .6745$) for case 4 with results computed by the steady-state code described in [17, 2]; the bulk temperature profiles for the other three cases are indistinguishable from the case 4 curve. The results computed using the algorithm described here and the steady-state code agree fairly well.

We now present timings for these four cases. The calculations were all run on a single 300 Mhz processor of a DEC Alpha workstation to a final time of $\Theta = .6745$. The average percentage of the domain refined to the finest level is $7 \pm 1\%$ for cases 2-4. Table 4 shows the same fields shown in table 1. The relative cost of modeling radiation is seen to be significantly higher in three dimensions than in two. The results in table 4 again show a reduction in computational cost when adaptive mesh refinement is used. Case 4 shows the best reduction in computational cost over the uniform grid calculation. As in the case of the thermally developing pipe flow problem, the comparative advantage of case 4 is probably due to the use of a refinement ratio of 4. However, the reduction in cost from using $r = 4$ is not as great in three dimensions as it is in two. We are uncertain why the relative cost

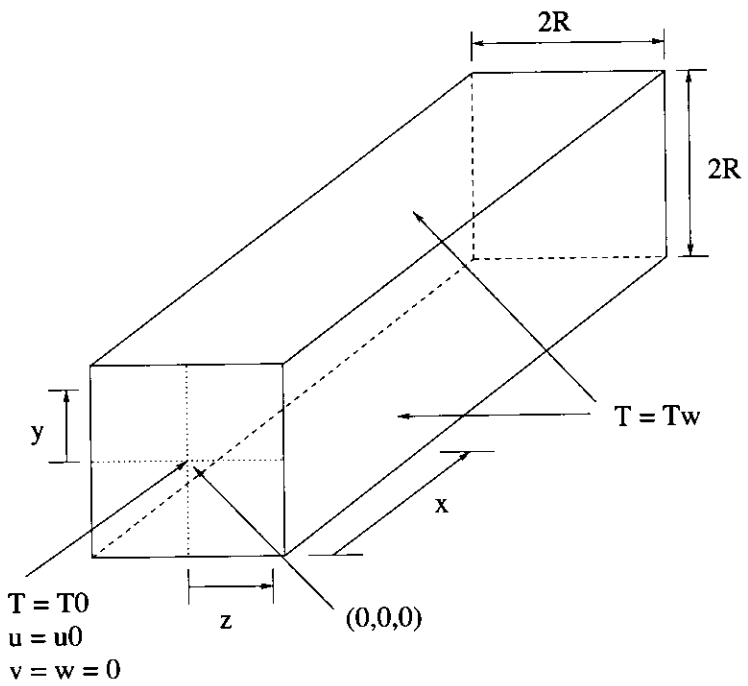

Figure 12: Specification and coordinate system for square duct.

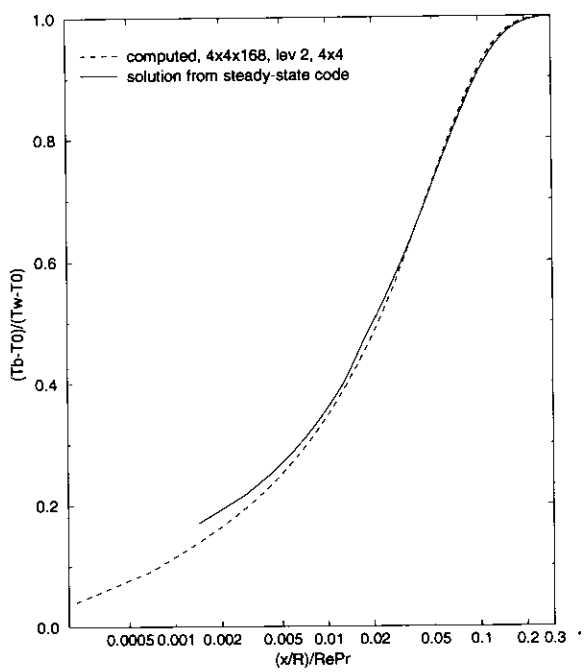

Figure 13: Comparison of the bulk temperature curve for case 4 of the simultaneously developing three-dimensional flow through a square duct example with the solution from the steady-state code described in [17, 2].

of modeling radiation is higher in three dimensions than in two. We do note that this higher cost is not due to the choice of ordinate sets; S_6 in two dimensions and S_4 in three dimensions both both have 24 ordinate directions.

Gridding			CPU Time		CPU Time for Radiation		Cells Advanced	Peak Memory Usage
Level 0 grid	ℓ_{max}	r	Total(s)	μs/cell	μs/cell	% CPU	Number	Mb
1) $16 \times 16 \times 672$	0		191500.	748.	330.	44.1	256155648	144
2) $8 \times 8 \times 336$	1	2	53470.	1491.	825.	55.3	35855360	36
3) $4 \times 4 \times 168$	2	2,2	23120.	1073.	553.	51.5	21550464	22
4) $4 \times 4 \times 168$	1	4	18000.	857.	374.	43.6	21011200	22

Table 4: Timings for uniform grid and refined grid calculations on a single processor of a DEC Alpha workstation for simultaneously developing three-dimensional flow through a square duct.

Shearing Flow in a Box

In the final example we consider an idealized flow in a closed unit box with reflecting walls ($\epsilon = 0$, $\rho = 1$), to confirm that the algorithm conserves energy even when the grid structure is complex and changes with time. The initial velocities are

$$u = -\sin^2(\pi x)\sin(2\pi y), \tag{22}$$

$$v = \sin^2(\pi y)\sin(2\pi x), \tag{23}$$

with slip-wall boundary conditions. Viscosity and conduction are set to 0. Density is a constant $\rho = 1$, as is $c_p = 5.0$. Temperature is a constant 150 except for a hot region of temperature 400, centered at (0.75, 0.25) and with radius 0.1. (No attempt has been made to choose physically reasonable values for any of these parameters, as the goal is only to illustrate the numerical properties of the scheme.)

The absorption coefficient κ is 2 throughout the domain, except for a disk of radius 0.1 at the center of the box where $\kappa = 50$. The base grid is 32×32, with two levels of refinement by factors of 2 and 4. An S_4 ordinate set is used. Figure 14 shows the temperature field at timesteps 0, 10 and 100 with no scattering, and at timestep 10 with a uniform scattering coefficient $\sigma = 10$. Adding scattering to the problem does not have a dramatic effect on the plots, but it does eliminate the ray effects visible in the corresponding frame of the purely absorbing calculation, and it reduces the heat transmitted to the central absorbing region.

There is no heat transfer through the walls, so the integral of enthalpy over the domain should be constant in time. In the experiment this quantity remained constant to 8 decimal places, with or without scattering. This precision is slightly better than expected, since the relative tolerance required for convergence of the radiation solutions was 10^{-7}, and the ordinates were taken from a table with only 7 decimal places.

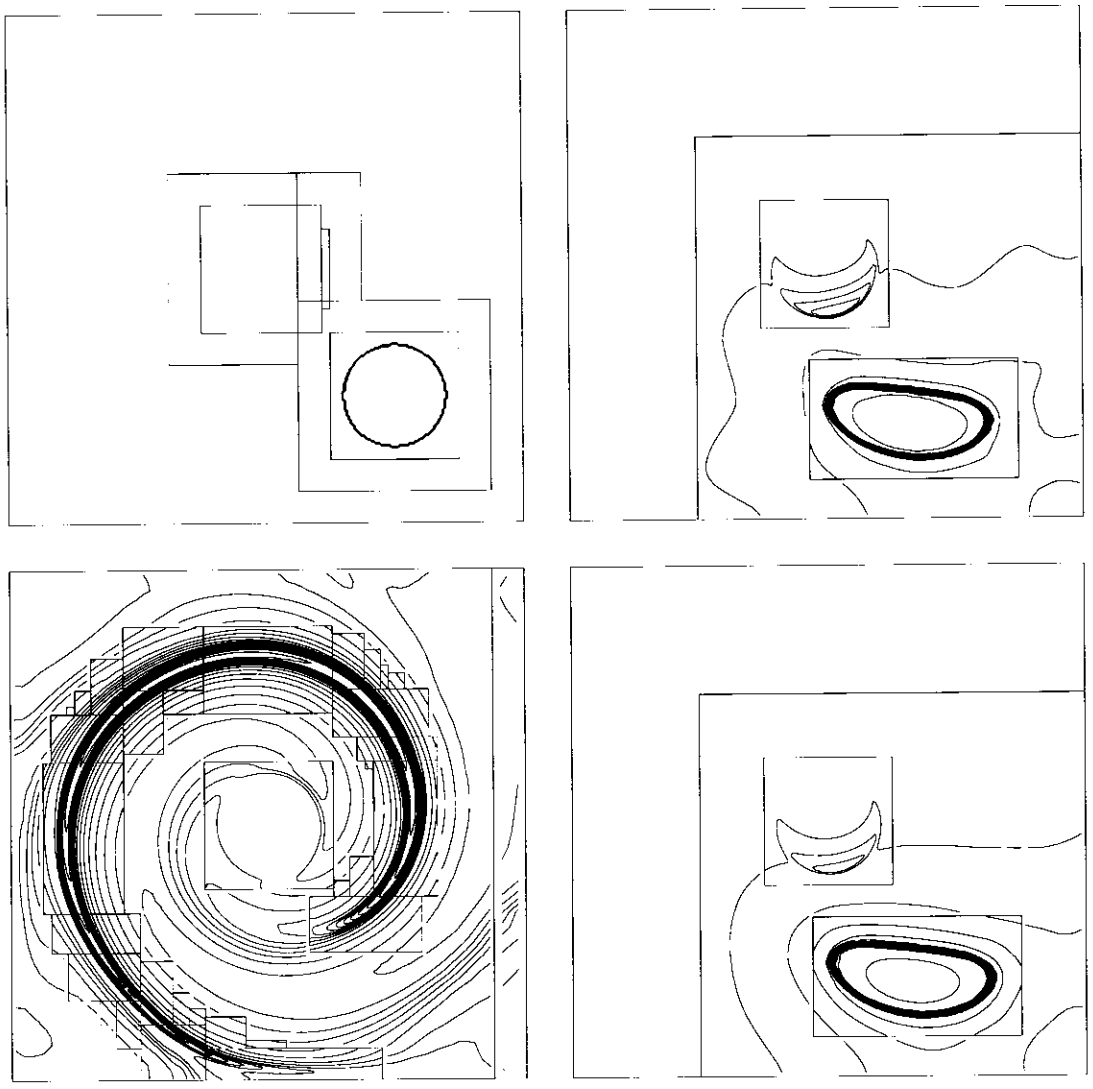

Figure 14: Evolution of an off-center hot spot in a spinning flow in a box with a central absorbing region. In the first three frames there is no scattering, while for comparison purposes the last frame (bottom right) shows the same calculation with scattering turned on.

Conclusions

We have presented a conservative, adaptive mesh algorithm for unsteady, combined mode heat transfer. The algorithm couples a new scheme for the modeling of unsteady radiative transport on locally refined meshes with an adaptive projection method for incompressible flow. This radiative transport scheme is itself based on a discrete ordinates algorithm for instantaneous radiative heat transfer on locally refined meshes. The methodology also incorporates a predictor-corrector formulation for second-order temporal accuracy.

The algorithm has been implemented and validated in two and three dimensions. The examples presented here demonstrate that the adaptive algorithm can compute accurate, energy conserving solutions while showing a significant reduction in CPU and memory usage over an uniform grid calculation. The results also show that the scheme is second-order accurate in time and space if a second-order discretization of the RTE is used.

In future work, the authors will incorporate the method presented here into an adaptive projection method for unsteady low-Mach number combustion [13]. The authors will also investigate the use of bounded, high-resolution differencing schemes [22] for the discrete-ordinates equations on locally refined meshes.

Acknowledgments

The code which implements the algorithm described in this paper is derived from the IAMR code [11] and shares a large common code base with it, including BOXLIB [24]. The authors would like to thank the following people: Ann Almgren, Vincent Beckner, John Bell, Marc Day, Charles Rendleman, David Stevens, and Michael Welcome for their contributions to the development of IAMR code; and Vincent Beckner, William Crutchfield, Michael Lijewski, Charles Rendleman, and Michael Welcome for their work on BOXLIB.

References

[1] M.G. Carvalho, P.J. Coelho, A.L.N. Moreira, A.M.C. Silva, and T.F. Silva. Comparison of measurements and predictions of wall heat flux and gas composition in an oil-fired utility boiler. In *Twenty-Fifth Symposium (International) on Combustion*, pages 227–234. The Combustion Institute, 1994.

[2] W. A. Fiveland and J. P. Jessee. A mathematical model for pulverized coal combustion in axisymetric geometries. In *ASME/EPRI Joint Power Conference*, Phoenix, AZ, October 3–6 1994.

[3] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[4] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[5] J. B. Bell, M. J. Berger, J. S. Saltzman, and M. Welcome. Three dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 15(1):127–138, 1994.

[6] B. G. Carlson and K. D. Lathrop. Transport theory—the method of discrete ordinates. In H. Greenspan, C. N. Kelber, and D. Okrent, editors, *Computing Methods in Reactor Physics*, pages 171–266. Gordon and Breach, New York, 1968.

[7] E. E. Lewis and W. F. Miller, Jr. *Computational Methods of Neutron Transport.* American Nuclear Society, La Grange Park, IL, 1993.

[8] W. A. Fiveland. Discrete-ordinates solutions of the radiative transport equation for rectangular enclosures. *J. Heat Transfer*, 106(4):699–706, 1984.

[9] J. P. Jessee, W. A. Fiveland, L. H. Howell, P. Colella, and R. B. Pember. An adaptive mesh refinement algorithm for the discrete ordinates method. In *Proceedings of the 1996 National Heat Transfer Conference*, Houston, August 3–6, 1996.

[10] J. P. Jessee, W. A. Fiveland, L. H. Howell, P. Colella, and R. B. Pember. An adaptive mesh refinement algorithm for the radiative transport equation. *J. Comput. Phys.*, accepted for publication, 1997.

[11] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.,* accepted for publication, 1996. LBNL-39075, Lawrence Berkeley National Laboratory, July, 1996. Also available as *http://www.nersc.gov/research/CCSE/publications/almgren/abchw96/paper.ps.gz.*

[12] R. B. Pember, A. S. Almgren, W. Y. Crutchfield, L. H. Howell, J. B. Bell, P. Colella, and V. E. Beckner. An embedded boundary method for the modeling of unsteady combustion in an industrial gas-fired furnace. In *1995 Fall Meeting of the Western States Section of the Combustion Institute*, Stanford, CA, October 30–31 1995 (WSS/CI 95F-165).

[13] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, J. P. Jessee, and W. A. Fiveland. An adaptive projection method for the modeling of unsteady, low-Mach number combustion. In *1997 Fall Meeting of the Western States Section of the Combustion Institute*, Diamond Bar, CA, October 23–24 1997 (WSS/CI 97F-125).

[14] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

[15] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *SIAM J. Sci. Comput.*, 17:358–369, 1996.

[16] P. Wesseling. *An Introduction to Multigrid Methods.* Wiley, New York, 1992.

[17] W. A. Fiveland. Three-dimensional radiative heat-transfer solutions by the discrete-ordinates method. *J. Thermophys. Heat Trans.*, 2(4):309–316, 1988.

[18] R. Siegel. Heat transfer for laminar flow in ducts with arbitrary time variations in wall temperature. *J. Appl. Mech.*, 27:241–249, 1960.

[19] F. M. White. *Viscous Fluid Flow*, pages 132–138. McGraw Hill, New York, 1974.

[20] R. V. Churchill and J. W. Brown. *Fourier Series and Boundary Value Problems*, pages 207–210. McGraw Hill, New York, 1978.

[21] K. D. Lathrop. Use of discrete-ordinates method for solution of the photon transport problems. *Nucl. Sci. Eng.*, 24:381–388, 1966.

[22] J. P. Jessee and W. A. Fiveland. Bounded, high-resolution differencing schemes applied to the discrete ordinates method. *J. Thermophys. Heat Trans.*, 11:540–548, 1997.

[23] B. E. Pearce and A. F. Emery. Heat transfer by thermal radiation and laminar forced convection to an absorbing fluid in the entry region of a pipe. *J. Heat Transfer*, 92:221–230, 1970.

[24] W. Y. Crutchfield and M. L. Welcome. Object-oriented implementations of adaptive mesh refinement algorithms. *Sci. Prog.*, 2:145–156, 1993.