LBNL-41339
Preprint

ERNEST ORLANDO LAWRENCE BERKELEY NATIONAL LABORATORY

An Adaptive Projection Method for Unsteady, Low-Mach Number Combustion

Richard B. Pember, Louis H. Howell, John B. Bell,
Phillip Colella, William Y. Crutchfield,
Woodrow A. Fiveland, and J. Patrick Jessee

Computing Sciences Directorate

January 1998

Submitted to
Combustion Science and Technology

REFERENCE COPY
Does Not
Circulate
Bldg. 50 Library - Ref.
Lawrence Berkeley National Laboratory
Copy 1
LBNL-41339

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.

LBNL-41339

An Adaptive Projection Method for Unsteady, Low-Mach Number Combustion

Richard B. Pember, Louis H. Howell, John B. Bell,
Phillip Colella, William Y. Crutchfield,
Woodrow A. Fiveland, and J. Patrick Jessee

Computing Sciences Directorate
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

January 1998

This work was supported by the Office of Energy Research, Applied Mathematical Sciences Program, Office of Computational and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

An Adaptive Projection Method for Unsteady, Low-Mach Number Combustion

Richard B. Pember [1,*] *Louis H. Howell* [2], *John B. Bell,*
Phillip Colella, and William Y. Crutchfield
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

Woodrow A. Fiveland and J. Patrick Jessee
McDermott Technology, Inc.
Alliance, OH 44601

Abstract

We present an adaptive projection method for modeling unsteady, low-Mach reacting flow in an unconfined region. The equations are based on a model for low-Mach number combustion that consists of evolution equations coupled with a constraint on the divergence of the flow. The algorithm is based on a projection methodology in which we first advance the evolution equations and then solve an elliptic equation to enforce the divergence constraint. The adaptive mesh refinement (AMR) scheme uses a time-varying hierarchy of rectangular grids. The integration scheme is a recursive procedure in which coarse grids are advanced, fine grids are advanced to the same time as the coarse grids, and the coarse and fine grid data are then synchronized.

The method is currently implemented for laminar, axisymmetric flames with a reduced kinetics mechanism and a Lewis number of unity. Three methane-air flames, two steady and one flickering, are presented as numerical examples.

Keywords: Laminar Diffusion Flames, Unsteady Combustion, Fluid Dynamic Aspects in Combustion, Numerical Modeling.

1 Introduction

The computational modeling of reacting flows with limited computer resources can be made difficult by the presence of multiple length scales and by the large number of species in a sufficiently detailed reaction mechanism. The problem of limited resources has generally been overcome in combustion modeling by using globally refined, nonuniform structured grids or by using unstructured grids.

In this paper we present a method based on a different approach, a structured grid, local adaptive mesh refinement (AMR) scheme. We develop an AMR algorithm to solve a system of equations for unsteady low-Mach number reacting flow in an unconfined region. This system is based on a generalization of the low-Mach number combustion model in Rehm and Baum (1978) and Majda and Sethian (1985). The system includes evolution equations

[1,*] Corresponding author. Correspondence information and current affiliation: L-560, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, pember@llnl.gov, (510) 422-4549, FAX: (510) 422-6287

[2] Current affiliation: Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

for density, velocity, enthalpy, and species concentrations, coupled with a constraint on the divergence of the flow.

Our approach to AMR uses a hierarchical-grid, structured approach first developed by Berger and Oliger (1984) and Berger and Colella (1989) for hyperbolic conservation laws. 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 coarse grids are advanced, fine grids are advanced multiple steps to reach the same time as the coarse grids, and the coarse and fine grids are synchronized. The method is valid for multiple grids on each level and for multiple levels of refinement.

The methodology presented here is based on a single grid algorithm developed by Pember *et al.* (1995; 1996). The single grid method is a fractional step scheme in which we first advance the evolution equations and then solve an elliptic equation to enforce the divergence constraint and update pressure. The solution of the evolution equations essentially follows the approach described due to Almgren *et al.* (1996; 1998). In order that the method be second-order accurate in time for nonlinear differential equations with source terms, however, a sequential, predictor-corrector treatment of the equations is used. The sequential approach ensures that all implicit finite difference equations are linear and can be solved by standard multigrid techniques (Wesseling, 1992), while the predictor-corrector formulation guarantees second-order accuracy in time. A simple extension of the second-order approximate projection algorithm in IAMR to low-Mach number compressible flows is employed to enforce the divergence constraint and update the pressure.

The single grid algorithm is coupled to an extension of the conservative adaptive mesh refinement scheme for variable density, constant viscosity incompressible flow (IAMR) developed by Almgren *et al.* (1995; 1998). In the present paper the IAMR algorithm is extended to account for the thermal expansion of the flow due to heat transfer and combustion, i.e., the non-zero divergence of the velocity. Additional enhancements ensure that the various relationships among the state quantities, in particular, density, enthalpy, temperature, and species concentrations, are always satisfied by the numerical solution. The treatment of scalars is also extended to account for evolution equations such as those for enthalpy and species concentrations. These two sets of extensions ensure that the method is freestream preserving with respect to primitive quantities as well as conservative and freestream preserving with respect to conserved quantities. Spatial and temporal variation of viscosity and of thermal and mass diffusivity are also accounted for.

The method is currently implemented for laminar, axisymmetric flames with a reduced kinetics mechanism and a Lewis number of unity. Results from three numerical examples, a steady methane-air diffusion flame (Smooke *et al.*, 1989), a steady methane-air diffusion flame in which the fuel is diluted with N_2 (Smooke *et al.*, 1992; Xu *et al.*, 1993; Smooke *et al.*, 1996; Bennett, 1997; Bennett and Smooke, 1997), and a flickering methane-air flame (Smyth *et al.*, 1993; Yam *et al.*, 1995; Smyth, 1997), are presented.

There are numerous references to the use of globally refined, non-uniform grids in combustion modeling. We refer the reader to Bennett (1997), Bennett and Smooke (1997), and the references therein. Local adaptive mesh refinement and local rectangular refinement

methods have been used to model steady, low-Mach number combustion. In addition to the two references above, see Coelho and Pereira (1993), de Lange and de Goey (1994), Mallens *et al.* (1995), Smooke *et al.* (1988), and Somers and de Goey (1995). The authors are unaware of any previous work using local adaptive mesh refinement to model unsteady low-Mach number combustion. Projection methods without mesh refinement have been developed for the unsteady case; see Dwyer (1990), Lai (1993), Lai *et al.* (1993), Najm (1996a; 1996b), Yam *et al.* (1995), and Hilditch and Colella (1996).

The remainder of this paper is organized as follows. In §2, we discuss the model for low-Mach number combustion and the governing equations solved with our approach. We describe the single grid algorithm in §3 and the adaptive algorithm in §4. Numerical results are shown in §5.

2 Model for Low-Mach Number Combustion and Governing Equations

The system of equations for reacting flow considered here is based on a model for low-Mach number combustion (Rehm and Baum, 1978; Majda and Sethian, 1985), which we now briefly review.

For flow in a spatially open domain, the underlying assumption in the low-Mach number model is that M is sufficiently small (say $M < .3$) that the pressure p can be written as the sum of a temporally and spatially constant part p_0 and a dynamic part π,

$$p(r,z,t) = p_0 + \pi(r,z,t), \tag{2.1}$$

where $\pi/p_0 = O\left(M^2\right)$. All thermodynamic quantities are considered to be independent of π. The perfect gas law for a multi-component gas in a flow satisfying the low-Mach number assumption is then

$$\rho = p_0/\left(TR\right) = p_0/\left(T\mathcal{R}/W\right) = p_0/\left(T\mathcal{R}\sum_l \left(Y_l/W_l\right)\right). \tag{2.2}$$

Differentiating (2.2) with respect to time and using continuity, the following constraint on the divergence of the velocity is obtained:

$$\nabla \cdot U = \frac{1}{T}\frac{DT}{Dt} + W\sum_l \frac{1}{W_l}\frac{DY_l}{Dt} \equiv S. \tag{2.3}$$

We consider flows that are axisymmetric without swirl. In addition, we assume a Lewis number of unity and neglect radiative heat transfer. The system of governing differential equations thus consists of the divergence constraint (2.3) and the following evolution equations for density, velocity, enthalpy, temperature, and species concentrations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho U = 0 \tag{2.4}$$

$$\rho\frac{DU}{Dt} = -\rho\left(0,g\right)^T - \nabla p + \nabla \cdot \tau \tag{2.5}$$

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot \rho U h = \nabla \cdot (\lambda / c_p) \nabla h \tag{2.6}$$

$$\rho c_p \frac{DT}{Dt} = \nabla \cdot \lambda \nabla T + \sum_l \rho D \nabla Y_l \cdot \nabla h_l(T) - \sum_l \omega_l h_l(T) \tag{2.7}$$

$$\frac{\partial \rho Y_l}{\partial t} + \nabla \cdot \rho U Y_l = \nabla \cdot \rho D \nabla Y_l + \omega_l. \tag{2.8}$$

The above system of equations is overdetermined in three ways. We retain these redundancies numerically in order to either ensure conservation or to simplify the solution strategy. Equations (2.6) and (2.7) are redundant because the enthalpy h is defined by

$$h = \sum_l Y_l h_l(T). \tag{2.9}$$

Equation (2.9) is used only to define the initial and inlet values of h; otherwise, h is found as the solution of (2.6) to ensure conservation of enthalpy. Moreover, equation (2.7) is used solely to define intermediate values of T; otherwise, T is computed using h, Y_l, and (2.9). The specific heat of the gas mixture c_p is found by

$$c_p = \sum_l Y_l c_{p,l}(T). \tag{2.10}$$

Equations (2.4) and (2.8) are also overdetermined because $\rho = \sum_l \rho Y_l$. We account for this redundancy by computing $\nabla \cdot \rho U$ as $\sum_l \nabla \cdot \rho U Y_l$. We can then advance ρ prior to updating the mass fractions. This allows us to use a simpler discretization of (2.8) and thereby to use a simpler solution strategy.

Equations (2.4) and (2.2) represent the last redundancy. The use of (2.4) ensures conservation of mass. The sequential approach makes it impossible, in general, to simultaneously satisfy the continuity equation and the equation of state. A pressure relaxation term is added to the numerical representation of the divergence constraint to account for this; see §3.1 for further discussion.

The diffusivities μ, D, and λ are in general considered to be functions of ρ, T, and Y_l. For the calculations shown in this paper, the viscosity μ is computed by the curve fit $\mu = \mu_0 (T/T_0)^{.7}$ (Kanuary, 1982), where $\mu_0 = 1.85 \times 10^{-5}$ kg/m-sec and $T_0 = 298$K. ρD and λ / c_p are determined from μ by $\rho D = \lambda / c_p = \mu / \mathrm{Pr}$. Following Smooke *et al.* (1989), we use $\mathrm{Pr} = .75$.

3 Single Grid Algorithm

The algorithm used to advance the solution from time t^n to $t^n + \Delta t = t^{n+1}$ on a single grid follows the general approach used in Pember *et al.* (1995) for the case of simple boundaries and incorporates many of the details of the single grid algorithm used in IAMR (Almgren *et al.*, 1998). The reader is referred to earlier works (Chorin, 1969; Bell *et al.*, 1989; Bell *et al.*, 1991; Bell and Marcus, 1992; Almgren *et al.*, 1996; Pember *et al.*, 1996) for additional discussion. We use a uniform grid of rectangular cells with widths Δr and Δz indexed by i and j. At the beginning of the time step, the numerical solution, except for

pressure, represents the flow at time t^n at cell centers. The solution for pressure, $p^{n-1/2}_{i+1/2,j+1/2}$, represents the pressure at the previous half-time step, $t^{n-1/2}$, on cell corners.

The method is essentially a second-order projection method (Bell *et al.*, 1989). The overall approach, then, is that of a fractional step scheme. In the first step (which we refer to as the convection-diffusion-reaction step), values of h, T, and Y_l are computed at time t^{n+1} using a higher-order upwind method for the convective terms and Crank-Nicholson differencing for the diffusive and the reactive terms. In addition, values of U, denoted by U^* or (u^*, v^*), are computed in this step which do not necessarily satisfy the divergence constraint at t^{n+1}. In the second step (the projection step), the divergence constraint is imposed on the velocity via a node-based projection (Almgren *et al.*, 1996). This step yields U^{n+1} and $p^{n+1/2}_{i+1/2,j+1/2}$, the pressure at $t^{n+1/2}$.

The first step uses a predictor-corrector formulation and consists of the following steps:

(1) Compute Δt:

$$\Delta t = \sigma \min_{ij} \left(\frac{\Delta r}{u^n_{ij}}, \frac{\Delta z}{v^n_{ij}}, \sqrt{\frac{2 \min(\Delta r, \Delta z)\, \rho}{\left|(0,-g)^T - (Gp)_{i,j}\right|}} \right) \tag{3.1}$$

where the Courant number σ satisfies $\sigma < 1$.

(2) Compute discrete approximations of the convective terms in the governing equations at time $t^n + \Delta t/2$ with an explicit higher-order upwind method:

$$(\nabla \cdot \rho U \varphi)^{n+1/2}_{ij} \text{ for } \varphi = h, Y_l \text{ and}$$
$$(U \cdot \nabla \varphi)^{n+1/2}_{ij} \text{ for } \varphi = u, v, T.$$

(3) Compute

$$\rho^{n+1}_{ij} = \rho^n_{ij} - \Delta t \sum_l (\nabla \cdot \rho U Y_l)^{n+1/2}_{ij} \tag{3.2}$$

and $\rho^{n+1/2}_{ij} = \left(\rho^n_{ij} + \rho^{n+1}_{ij}\right)/2$.

(4) Compute predicted values $\varphi^{n+1,p}$ of the solution at t^{n+1} for the flow quantities $\varphi = Y_l, T$, and h using Crank-Nicholson temporal differencing of the diffusion terms in conjunction with the time-centered convective terms found in step (2). In this step, diffusivities and thermochemical properties at time $n+1$ are evaluated using the state at time n.

(5) Compute corrected values of Y_l, T, and h and values of (u^*, v^*) to provide the solution at time t^{n+1}, again using Crank-Nicholson differencing. Properties at time $n+1$ are evaluated here using the predicted state found in step (4).

In step (2), a MAC projection (Harlow and Welch, 1964) is performed so that the edge velocities used to form the convective derivatives satisfy the divergence constraint. In steps (4) and (5) the equations for each of the flow quantities Y_l, h, T, and (u^*, v^*) are solved sequentially so that only linear systems of equations result from the Crank-Nicholson differencing. The update for (u^*, v^*) is a coupled solve due to the tensor nature of τ. Note that the velocity is not predicted in step (4) because predicted values of the velocity are not needed in step (5). In the predictor step, T is advanced using (2.7); this approach is

typically less computationally expensive than solving (2.9) for $T^{n+1,p}$. In the corrector step, T^{n+1} is found by solving (2.9) for T.

The species update is itself performed sequentially in two steps, one accounting for convection and diffusion and the other for kinetics, in order to facilitate the use of complex kinetics mechanisms. In the kinetics update, the system of equations $\partial \rho Y_l / \partial t = \omega_l$ is integrated with an implicit difference scheme.

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 (Wesseling, 1992; Almgren *et al.*, 1998). 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.

In the remainder of this section, we present details of the above algorithm.

We note here that the details of the algorithm are modified for the first time step. We follow the procedure used in IAMR; in particular, before any time steps are taken, the initial velocity field is projected to ensure that it satisfies the divergence constraint.

3.1 Numerical divergence constraint

The right hand sides of equations (2.7) and (2.8) can be used to obtain the following expression for S:

$$\begin{aligned} S &= \frac{1}{\rho c_p T}\left(\nabla \cdot \lambda \nabla T + \sum_l \rho D \nabla Y_l \cdot \nabla h_l\right) + \\ &\quad \frac{W}{\rho}\sum_l \frac{1}{W_l}\nabla \cdot D\rho\nabla Y_l + \frac{1}{\rho}\sum_l \left(\frac{W}{W_l} - \frac{h_l(T)}{c_p T}\right)\omega_l. \end{aligned} \tag{3.3}$$

Numerically, ω_l/ρ is approximated by $\Delta Y_l/\Delta t$, where ΔY_l is the change in Y_l due to chemical reactions during the time step. The other terms are approximated by central differences.

If equation (3.3) is used without modification, however, the algorithm may suffer from a mild instability arising because the sequential approach cannot simultaneously conserve mass and enforce the constraint $p_0 = \rho RT$; at the very least, the solution drifts from this constraint. (Analytically, this is not an issue; the equation of state and the continuity equation (2.4) are equivalent (Majda and Sethian, 1985).) In our approach, expression (3.2) guarantees conservation of mass. To stabilize the method, we add an extra term to the discrete form of the divergence constraint (3.3) which accounts for the discrepancy between the value of ρ found by continuity and that found using the equation of state. The value of the right hand side of the divergence constraint used numerically, $\tilde{S}$, is found by incrementing S as follows,

$$\tilde{S}_{ij} = S_{ij} + f\left(\tilde{p}_{ij} - p_0\right)\frac{c_{p,ij} - R_{ij}}{\Delta t c_{p,ij}\tilde{p}_{ij}} \tag{3.4}$$

where $\tilde{p}_{ij} = R_{ij}\rho_{ij}T_{ij}$ and f is a constant satisfying $f < 1.0$. The extra term in the numerical divergence constraint is found by approximating Dp/Dt in the enthalpy equation for non-isobaric flow (Kuo, 1986) by $\left(\tilde{p}_{ij} - p_0\right)/\Delta t$, rewriting the resultant equation in terms of T,

and using (2.3). The term $f(\tilde{p}_{ij} - p_0)/\Delta t$ acts to drive the solution back to the constraint $\tilde{p}_{ij} = p_0$. Similar treatments have been used in numerical petroleum reservoir simulation (Trangenstein and Bell, 1989).

Equation (3.4) is evaluated once per time step, immediately prior to the projection step, to determine $\tilde{S}^{n+1}$. $\tilde{S}^n$ is used whenever an evaluation of $\nabla \cdot U^n$ is needed.

For the MAC projection, we also need an estimate of $\partial\tilde{S}/\partial t$ in order to approximate $\tilde{S}$ at $t^{n+1/2}$. We use

$$\left(\frac{\partial \tilde{S}}{\partial t}\right)^n_{ij} \approx \frac{\tilde{S}^n_{ij} - \tilde{S}^{n-1}_{ij}}{\Delta t}. \tag{3.5}$$

3.2 Convection-Diffusion-Reaction Step

3.2.1 Computation of convective derivatives

The approximation of the convective derivatives generally follows the approach used in IAMR (Almgren *et al.*, 1998); see Bell *et al.* (1991) for additional discussion. There are two primary components to this computation: a higher-order upwind scheme (Colella, 1990) to determine edge states and a MAC projection (Harlow and Welch, 1964) to enforce the divergence constraint on the edge velocities.

The general procedure can be summarized as follows:

(1) Compute values of $u^{n+1/2}_{i+1/2,j}$ and $v^{n+1/2}_{i,j+1/2}$, on all r- and z- cell edges, respectively, using the higher-order upwind scheme.

(2) Compute advection velocities $u^{ADV}_{i+1/2,j}$ and $v^{ADV}_{i,j+1/2}$ by projecting the edge velocities found in (1) so that they satisfy the divergence constraint.

(3) Recompute $u^{n+1/2}_{i+1/2,j}$ and $v^{n+1/2}_{i,j+1/2}$, and compute $v^{n+1/2}_{i+1/2,j}$, $u^{n+1/2}_{i,j+1/2}$, $T^{n+1/2}_{i+1/2,j}$, $T^{n+1/2}_{i,j+1/2}$, $(\rho Y_l)^{n+1/2}_{i+1/2,j}$, $(\rho Y_l)^{n+1/2}_{i,j+1/2}$, $(\rho h)^{n+1/2}_{i+1/2,j}$, and $(\rho h)^{n+1/2}_{i,j+1/2}$ using the higher-order upwind scheme.

(4) Form discrete approximations of convective terms.

The first step follows the approach in IAMR. First, time-centered left and right edge states, $u^{n+1/2}_{i+1/2,j,L}$ and $u^{n+1/2}_{i+1/2,j,R}$, at all r-cell faces and bottom and top edge states, $v^{n+1/2}_{i,j+1/2,L}$ and $v^{n+1/2}_{i,j+1/2,R}$, at all z-cell faces are found with Taylor expansions that use monotonicity-limited approximations to the spatial derivatives in the convective terms. (Other spatial derivatives are evaluated by standard central difference approximations.) The time-centered edge states $u^{n+1/2}_{i+1/2,j}$ at all r-cell faces and $v^{n+1/2}_{i,j+1/2}$ at all z-cell faces are then found by an upwinding procedure.

In step (2), we use a MAC projection to enforce the divergence constraint (3.4). The equation

$$\left(D^{MAC}\frac{1}{\rho^n}G^{MAC}\phi\right)_{ij} = \left(D^{MAC}U^{n+1/2}\right)_{ij} - \left(\tilde{S}^n_{ij} + \frac{\Delta t}{2}\frac{\partial \tilde{S}^n}{\partial t}_{ij}\right) \tag{3.6}$$

is solved for ϕ, where $\tilde{S}^n$ and $\partial\tilde{S}/\partial t^n$ are given by (3.4) and (3.5), and D^{MAC} and G^{MAC} are the standard discretizations of the divergence and gradient operators on a staggered

MAC grid (Almgren *et al.*, 1998). The advection velocities are then computed by

$$\begin{aligned} u^{ADV}_{i+1/2,j} &= u^{n+1/2}_{i+1/2,j} - \frac{1}{\rho^n_{i+1/2,j}}(G^{MAC}\phi)^r_{i+1/2,j} \\ v^{ADV}_{i,j+1/2} &= v^{n+1/2}_{i,j+1/2} - \frac{1}{\rho^n_{i,j+1/2}}(G^{MAC}\phi)^z_{i,j+1/2}, \end{aligned} \tag{3.7}$$

where the edge values of ρ are averages of the adjacent cell centered values.

In step (3), we recompute $u^{n+1/2}_{i+1/2,j}$ and $v^{n+1/2}_{i,j+1/2}$, and compute $v^{n+1/2}_{i+1/2,j}$, $u^{n+1/2}_{i,j+1/2}$, $T^{n+1/2}_{i+1/2,j}$, $T^{n+1/2}_{i,j+1/2}$, $(\rho Y)^{n+1/2}_{i+1/2,j}$, and $(\rho Y)^{n+1/2}_{i,j+1/2}$, again using the approach in IAMR. In this step, the upwind states are found using the MAC projected edge velocities from step (2).

$(\rho h)^{n+1/2}_{i+1/2,j}$ and $(\rho h)^{n+1/2}_{i,j+1/2}$ are computed in a slightly different manner. The edge values of T are used to compute edge values of $h_l(T)$ for all species l. These values of h_l and the edge values of ρY_l are then used to compute edge values of ρh using (2.9).

In step (4), the convective derivatives are approximated by

$$\begin{aligned} (\nabla \cdot U\rho\varphi)^{n+1/2}_{ij} &= \frac{r_{i+1/2}u^{ADV}_{i+1/2,j}\,(\rho\varphi)^{n+1/2}_{i+1/2,j} - r_{i-1/2}u^{ADV}_{i-1/2,j}\,(\rho\varphi)^{n+1/2}_{i-1/2,j}}{r_i\Delta r} + \\ &\quad \frac{v^{ADV}_{i,j+1/2}\,(\rho\varphi)^{n+1/2}_{i,j+1/2} - v^{ADV}_{i,j-1/2}\,(\rho\varphi)^{n+1/2}_{i,j-1/2}}{\Delta z} \text{ for } \varphi = h, Y_l \text{ and} \end{aligned} \tag{3.8}$$

$$\begin{aligned} (U \cdot \nabla\varphi)^{n+1/2}_{ij} &= \frac{\left(u^{ADV}_{i+1/2,j} + u^{ADV}_{i-1/2,j}\right)\left(\varphi^{n+1/2}_{i+1/2,j} - \varphi^{n+1/2}_{i-1/2,j}\right)}{2\Delta r} + \\ &\quad \frac{\left(v^{ADV}_{i,j+1/2} + v^{ADV}_{i,j+1/2}\right)\left(\varphi^{n+1/2}_{i,j+1/2} - \varphi^{n+1/2}_{i,j-1/2}\right)}{2\Delta z} \text{ for } \varphi = u, v, T. \end{aligned} \tag{3.9}$$

The higher-order upwind scheme used in steps (1) and (3) uses a second-order Taylor series expansion in time and space about (r_i, z_j, t^n) to determine left and right (bottom and top) states at time $t^{n+1/2}$ at $r-$ $(z-)$ edges. The time derivative in the Taylor expansion is expressed in terms of the spatial derivatives and lower order terms by using a quasilinear form of the appropriate governing equation. The particular form of the quasilinear equation for a given state variable φ depends on whether we compute $\rho\varphi$ or φ at edges. In the former case, $\rho\varphi$ is computed directly – there is not a separate computation of ρ – and in the quasilinear equation, $\nabla \cdot \rho U\varphi$ is expressed as $U \cdot \nabla(\rho\varphi) + \rho\varphi\nabla \cdot U$. Note that in the case of ρY_l, we omit the ω_l term from the quasilinear equation because of the operator split treatment of the kinetics.

The edge values of ρh are computed in the manner described above to ensure that the numerical scheme is freestream preserving with respect to temperature in the presence of multiple species. The convection scheme uses van Leer slope limiting (van Leer, 1979) in the approximation of the first-order spatial derivatives. The scheme is hence monotonicity preserving but also necessarily nonlinear (LeVeque, 1990). In particular, then, if the edge values of ρh were computed in the same manner as ρY_l, edge values of ρY_l and ρh would not necessarily satisfy (2.9) under isothermal conditions; the scheme might then incorrectly generate a non-constant temperature field.

3.2.2 Crank-Nicholson differencing

In steps (4) and (5) of the convection-diffusion-reaction step we solve difference equations obtained by applying the Crank-Nicholson method to the governing equations. The difference equations are solved using standard multigrid techniques (Wesseling, 1992). By using a sequential approach and a predictor-corrector formulation, these difference equations are linear and uncoupled in the sense that we can solve for T, h, Y_l, and (u^*, v^*) separately. In step (4), we compute predicted values of temperature, species mass fractions, and enthalpy at time $n+1$. Note that we do not need to find predicted values of (u^*, v^*) because the equations have no coupled or nonlinear dependencies on the velocity; in particular, we do not need predicted values of the velocity to compute predicted values of μ, D, and λ. In step (5), we compute corrected values of T, Y_l, and h, as well as (u^*, v^*). In the corrector step, T^{n+1} is found directly by solving (2.9) given values of h^{n+1} and Y_l^{n+1}.

We now summarize the difference equations for Y_l, h, T and U; the cell indices ij are suppressed. The details of the discretizations of the divergence and gradient operators, except in the case of of $\nabla\cdot\tau$, are discussed in Almgren *et al.* (1998). The discretization of $\nabla\cdot\tau$ uses similar strategies and is discussed in Appendix A. Note that in all the discretizations, edge-based values of the appropriate diffusivity are needed. These are found by simple averages of the cell-based values.

The discretization of the evolution equation for Y_l used in the corrector is

$$\frac{\rho^{n+1}Y_l^{n+1} - \rho^n Y_l^n}{\Delta t} + (\nabla\cdot\rho U Y_l)^{n+1/2} = \frac{1}{2}\left(\nabla\cdot(\rho D)^n \nabla Y_l^n + (\rho D)^{n+1,p}\nabla Y_l^{n+1}\right).$$

In the predictor, $(\rho D)^n$ is used instead of $(\rho D)^{n+1,p}$. Note that ω is not included because of the operator split treatment of kinetics. The discretizations of the enthalpy equation have a similar form. The form of the difference equation for temperature used in the predictor is slightly different because of the terms accounting for enthalpy transport due to interdiffusion of species in (2.7):

$$\rho^{n+1/2}c_p^n\left(\frac{T^{n+1,p}-T^n}{\Delta t} + (U\cdot\nabla T)^{n+1/2}\right) = \frac{1}{2}\left(\nabla\cdot\nabla\lambda^n\nabla T^n\right.$$
$$\left.+\nabla\cdot\nabla\lambda^n\nabla T^{n+1,p}\right) + (\rho D)^n\sum_l \nabla h_l\left(T^n\right)\cdot\nabla Y_l^n.$$

As in the case of the species equation, ω is not included. Finally, the discretization of the momentum equation is a coupled difference equation for $U^* = (u^*, v^*)$:

$$\rho^{n+1/2}\frac{U^*-U^n}{\Delta t} = \frac{1}{2}\left((\nabla\cdot\tau)^n + (\nabla\cdot\tau)^{n+1}\right) - \rho^{n+1/2}(U\cdot\nabla U)^{n+1/2} - (\nabla p)^{n-1/2}. \quad (3.10)$$

The viscosities in $(\nabla\cdot\tau)^n$ and $(\nabla\cdot\tau)^{n+1}$ are evaluated using T^n and $T^{n+1,p}$, respectively. Note that the pressure gradient is lagged.

3.3 Projection Step

A projection (Almgren *et al.*, 1996) is now used to approximately enforce the divergence constraint (3.4) and determine $p^{n+1/2}$. In the convection-diffusion-reaction step, we use

(3.10) and a time-lagged pressure gradient to compute a velocity that does not necessarily satisfy the divergence constraint (3.4). In the projection we enforce

$$\begin{aligned} \rho_{ij}^{n+1/2}\frac{U_{ij}^{n+1}-U_{ij}^{n}}{\Delta t} &= \frac{1}{2}\left((\nabla\cdot\tau)^{n}+(\nabla\cdot\tau)^{n+1}\right)-\rho^{n+1/2}\left(U\cdot\nabla U\right)_{ij}^{n+1/2}-(\nabla p)_{ij}^{n+1/2} \\ (\nabla\cdot U)_{ij}^{n+1} &= \tilde{S}_{ij}^{n+1}. \end{aligned} \tag{3.11}$$

From (3.10) and (3.11), we see that

$$\frac{U_{ij}^{n+1}-U_{ij}^{n}}{\Delta t}+\frac{1}{\rho_{ij}^{n+1/2}}\left(\nabla\delta\right)_{ij}=\frac{U_{ij}^{*}-U_{ij}^{n}}{\Delta t} \tag{3.12}$$

where $\delta_{i+1/2,j+1/2}=p_{i+1/2,j+1/2}^{n+1/2}-p_{i+1/2,j+1/2}^{n-1/2}$. Taking the divergence of (3.12), we obtain the following equation,

$$\nabla\cdot\left(\frac{1}{\rho_{ij}^{n+1/2}}\left(\nabla\delta\right)_{ij}\right)=\nabla\cdot\left(\frac{U_{ij}^{*,n+1}-U_{ij}^{n}}{\Delta t}\right)-\frac{\tilde{S}_{ij}^{n+1}-\tilde{S}_{ij}^{n}}{\Delta t}, \tag{3.13}$$

which we solve using a standard finite-element bilinear discretization. U^{n+1} and $p^{n+1/2}$ are then found by

$$\begin{aligned} U_{ij}^{n+1} &= U_{ij}^{*}-\frac{\Delta t}{\rho_{ij}^{n+1/2}}\left(\bar{G}\delta\right)_{ij} \\ p_{i+1/2,j+1/2}^{n+1/2} &= p_{i+1/2,j+1/2}^{n-1/2}+\delta_{i+1/2,j+1/2} \end{aligned} \tag{3.14}$$

where $\left(\bar{G}\delta\right)_{ij}$ represents the cell average of $G\delta$ over cell ij.

An additional step is needed because S may be underresolved, in particular, if there are extremely steep gradients in the temperature field or in any of the species concentration fields, or if the flame is very thin with respect to the grid spacing. In such situations, the velocity found above may contain spurious modes in the regions where S is underresolved. The modes can persist in time even after the underresolved gradients have dissipated; in particular, $\nabla\cdot U$ may be non-zero in a region where S is uniformly zero but where it was underresolved at an earlier time. We believe this problem arises due to the approximate nature of the projection. To correct it, we modify the value of U found in (3.14) by using the following filter,

$$U_{ij}^{n+1}:=U_{ij}^{n+1}+f\Delta r\Delta z\nabla\left((\nabla\cdot U)_{ij}^{n+1}-\tilde{S}_{ij}^{n+1}\right), \tag{3.15}$$

where f is a constant satisfying $f<1.0$. This update has the effect of relaxing U back to the constraint $\nabla\cdot U=S$. We use (3.15) in all computational cells.

4 Extension to Adaptive Mesh Refinement

In this section we describe the extension of the single grid algorithm to an adaptive hierarchy of nested rectangular grids. The methodology is based on the IAMR algorithm described

by Almgren *et al.* (1998). Many of details of the present algorithm are identical, or very nearly so, to those of the IAMR algorithm. The reader is referred to the above reference for these. In the following subsections, we review the features common to both algorithms to provide context but otherwise emphasize those that are specific to the modeling of low-Mach number reacting flow.

4.1 Grid Hierarchy and Overview of Time-Stepping Procedure

The adaptive mesh refinement (AMR) algorithm uses a hierarchical grid structure, which changes dynamically, composed of rectangular, uniform grids of varying resolution. The collection of grids at a given resolution is referred to as a level. By definition, level 0 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 a 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 levels ℓ and $\ell + 2$. (See Figure 1).

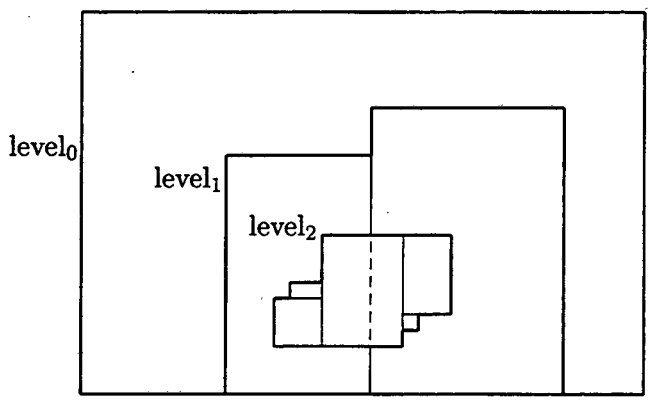

Figure 1: A properly nested hierarchy of grids

On the full adaptive mesh, the AMR 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. Figure 2 shows a space-time diagram of a single 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 new level $\ell+1$ grids to cover this region, and transfer data to new grids (using conservative interpolation from level ℓ if necessary).

In the remainder of this section, we refer to steps 1 and 2 as a complete coarse level advance or time step; step 1 is referred to as a level advance or a level ℓ advance.

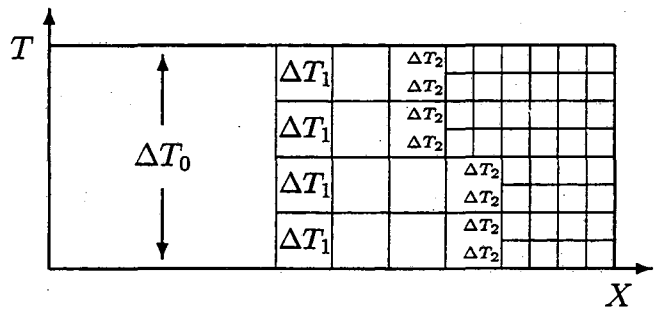

Figure 2: Multilevel timestep structure

The algorithm to advance a single level uses the same sequence of steps as the single grid algorithm presented in §3. Note that the MAC projection, the Crank-Nicholson solves, and the nodal projection must be done on all grids in a level simultaneously.

A detailed treatment of boundary conditions for the level advance is presented in Almgren *et al.* (1998). For our purposes, we need only mention that boundary conditions for the convection and the Crank-Nicholson steps are essentially implemented by filling ghost cells of the grids. The ghost cells which are interior to the problem domain but exterior to all of the level grids are filled by conservative interpolation from the underlying coarser level grids.

4.2 Managing the Grid Hierarchy

In the adaptive algorithm, the flow quantities whose values must persist from one time step to the next are the dependent variables in the evolution equations, in particular, ρ, U, T, ρh and ρY_l, and the pressure p. (T could be recomputed at the beginning of each step; we let the value of T persist simply to avoid an extra solution of (2.9).)

The variables $\tilde{S}$ and $\partial\tilde{S}/\partial t$ are also treated as persistent. The values of these at a given level ℓ are computed by (3.3) and (3.5) only before the projection step during the level advance. Otherwise, they are computed by averaging down (at the end of a complete level ℓ time step in cells covered by level $\ell+1$ cells) or by conservative interpolation to level ℓ cells (in level ℓ cells that are newly created by regridding or that are ghost cells not contained within existing level ℓ grids.) Values of $\partial\tilde{S}/\partial t$ are persistent simply because computing $\partial\tilde{S}/\partial t$ at time n requires values of $\tilde{S}$ at t^{n-1} as well as t^n. Within a single level, $\tilde{S}$ could be recomputed at the beginning of each time step. To do so, however, would require a reevaluation of the reaction rates used in the previous time step; we wish to avoid this computation since it can be expensive. For fine grid cells that are newly created during

regridding and for coarse grid cells that underlay fine grid cells, the same argument applies. We note that at the beginning of a time step, the velocity U may not satisfy $\nabla \cdot U = \tilde{S}$ in newly created fine grid cells and in underlying coarse grid cells. However, during the subsequent time step, the divergence of U is driven toward $\tilde{S}$ by the filter (3.15).

The treatment of the primitive quantities T, Y_l, and h also requires discussion. Whenever ρh and ρY_l have been defined by conservative interpolation or redefined by synchronization, T is recomputed according to (2.9). Within a given level, Y_l and h are defined in the obvious way. In ghost cells completely exterior to a level, Y_l and h are defined by first conservatively interpolating ρ, ρY_l, and ρh.

The conservative interpolation of the quantities ρ, ρY_l, and ρh is the final area requiring general discussion. As in the single level convection step, the conservative interpolation algorithm uses van Leer slope limiting (van Leer, 1979) in the approximation of spatial derivatives. For the same reasons discussed in §3.2.1, if the conservative interpolation scheme were used without modification, interpolated values of ρh and ρY would not necessarily satisfy (2.9) under isothermal conditions. Further, interpolated values of ρ and ρY_l might not satisfy $\rho = \sum_l \rho Y_l$. In order to overcome these shortcomings, we modify the slope calculation procedure used in the interpolation scheme. In a given cell, we compute van Leer-limited slopes and unlimited central-difference slopes of ρ, ρY_l, and ρh. We then compute the minimum of the ratios of the limited slopes to the unlimited slopes, where the ratio is defined to be one if the slope is zero. The slopes $\delta\varphi$, $\varphi = \rho, \rho h, \rho Y_l$, used in interpolation are then defined to be this minimum ratio times the unlimited slopes, i.e.,

$$\delta\varphi = \min\left(\frac{\delta_{\text{lim}}\rho h}{\delta_{\text{unlim}}\rho h}, \frac{\delta_{\text{lim}}\rho}{\delta_{\text{unlim}}\rho}, \min_l\left(\frac{\delta_{\text{lim}}\rho Y_l}{\delta_{\text{unlim}}\rho Y_l}\right)\right)\delta_{\text{unlim}}\varphi, \; for\ \varphi = \rho, \rho h, \rho Y_l, \quad (4.1)$$

where δ_{lim} and δ_{unlim} denote the van Leer limited and the unlimited slopes. In the synchronization step, corrections for ρ, ρh, and ρY_l at a given level may need to be interpolated to finer levels. The interpolation of these corrections follows the same strategy.

4.3 Synchronization

The general synchronization issues for the present algorithm are roughly the same as those for IAMR (Almgren *et al.*, 1998). Before discussing details specific to low-Mach number combustion, we briefly review these.

The advance of a single level entails a number of convective and diffusive solves as well as projections. During the advance of a given fine level, we use Dirichlet boundary data for each such operation from the next coarser level at coarse-fine interfaces. Even though the solution within each level is consistent, there is a mismatch at the coarse-fine interface at the end of a complete coarse grid advance prior to the synchronization step. Specifically, there are four mismatches between a coarse and a fine level after a complete coarse level time step (we adopt the notation from Almgren *et al.* (1998)):

(M.1) The solution in coarse cells underlaying fine grid cells is not synchronized with the overlying fine grid solution.

(M.2) The composite advection velocity, properly defined, does not satisfy a properly defined composite divergence constraint at the coarse-fine interface.

(M.3) The convective and diffusive fluxes from the coarse and the fine levels do not agree along the coarse-fine interface.

(M.4) The coarse and fine cell-centered velocity do not satisfy a properly defined composite divergence constraint at the coarse-fine interface.

The purpose of the synchronization step is to correct the effects of each mismatch. We use the notation (S.n) to refer to the correction for mismatch (M.n). In the remainder of this section we discuss the correction strategies.

(M.1) is corrected by averaging the fine grid data onto the coarse grid data as in IAMR. Note that here we average $\tilde{S}$ and $\partial\tilde{S}/\partial t$ onto the coarse grid as well. We also average T onto the coarse grid to provide the temperature used to compute diffusivities in (S.3).

Mismatch (M.2) is corrected with the same approach used in IAMR. During the coarse and fine grid level advances, the difference between the coarse and the fine grid advection velocities at a given cell edge along the interface are accumulated in a time and area weighted fashion.

In (S.2), the accumulated differences appear as the right hand side of a MAC sync solve whose result is a correction to all the coarse grid advection velocities. Because the coarse and fine grid velocities both satisfy the divergence constraint within their respective levels, the velocity correction is divergence free; hence, the elliptic equation that is solved in this step is identical to that solved in IAMR. for incompressible flow. Because the advection velocities used in the original coarse level advance did not contain this correction, we repeat the coarse level convection step to generate flux corrections that account for the convective transport due to the advective velocity corrections. Note that in this computation, which we call the MAC sync convection step, we follow the same prescription for ρh as was used in §3.2.1.

The correction for (M.3) uses the same general approach as in IAMR. There are, however, a number of modifications and additional details. For a given coarse cell edge along the coarse-fine interface, the differences between the coarse and fine level fluxes (both convective and diffusive) are accumulated. A cell-centered correction field is defined on the coarse grid cells by combining the accumulated flux differences, which are associated with the coarse cells along the interface outside the fine grids, and the advection updates arising from the corrections to the advection velocities in the MAC sync convection step.

Unlike (S.1), (S.3) affects the solution at the entire coarse level and all finer levels. We first define the coarse grid corrections to the scalar fields. We denote the scalar correction fields by $RHS_\rho, RHS_{\rho h}$, and $RHS_{\rho Y_l}$. The values of the state quantities after (S.1) but prior to (S.3) are denoted by $(\cdot)^{n+1,\mathrm{S.1}}$. First, we redefine RHS_ρ to be $\sum_l RHS_{\rho Y_l}$. ρ^{n+1} is then found by

$$\rho^{n+1} = \rho^{n+1,\mathrm{S.1}} + RHS_\rho.$$

For $\varphi = h, Y_l$, we can write

$$(\rho\varphi)^{n+1} - (\rho\varphi)^{n+1,\mathrm{S.1}} = RHS_{\rho\varphi}$$

$$= \rho^{n+1}\left(\varphi^{n+1} - \varphi^{n+1,\mathrm{S.1}}\right) + \varphi^{n+1,\mathrm{S.1}}\left(\rho^{n+1} - \rho^{n+1,\mathrm{S.1}}\right). \quad (4.2)$$

We see that there are two components to the correction to $\rho\varphi$: a correction to ρ and a correction to φ. The correction to $\rho\varphi$ therefore has two steps. We first solve the difference equation

$$\rho^{n+1}\varphi_{corr} - \frac{\Delta t}{2}\nabla\cdot\frac{\mu^{n+1,\mathrm{S.1}}}{\mathrm{Pr}}\nabla\varphi_{corr} = RHS_{\rho\varphi} - \varphi^{n+1,\mathrm{S.1}}\left(\rho^{n+1} - \rho^{n+1,\mathrm{S.1}}\right) \quad (4.3)$$

for φ_{corr}, where φ_{corr} denotes $\varphi^{n+1} - \varphi^{n+1,\mathrm{S.1}}$. $(\rho\varphi)^{n+1}$ is then computed by

$$(\rho\varphi)^{n+1} = (\rho\varphi)^{n+1,\mathrm{S.1}} + \rho^{n+1}\varphi_{corr} + \varphi^{n+1,\mathrm{S.1}}\left(\rho^{n+1} - \rho^{n+1,\mathrm{S.1}}\right).$$

The coarse grid velocity correction in (S.3) follows the same approach used in IAMR, with straightforward modifications for non-constant viscosity and the tensor form of τ; see Appendix A for details. All the coarse grid corrections are conservatively interpolated to the overlying fine grid cells in all finer levels. Finally, T is recomputed on the coarse and all finer levels using equation (2.9).

The final mismatch, (M.4), is corrected with a similar approach to that used in IAMR. During the coarse and fine grid level advances, a composite residual is accumulated at the coarse nodes at the coarse-fine interface that measures the extent to which the level projections fail to satisfy the composite projection equations at the interface.

Unlike the case of the MAC projection, there is a contribution to this residual due to the compressibility of the flow. At a given coarse node at the coarse-fine interface, there is a contribution to the residual from the value of $\partial\tilde{S}/\partial t$ (3.5) in each coarse cell outside the fine grid which shares the node and each fine cell bordering any of these coarse cells. The total residual Res_{SP}^{coarse} (the "SP" subscript denotes sync projection) equals the residual $Res_{SP,\nabla\cdot U=0}^{coarse}$ for incompressible flow (Almgren *et al.*, 1998) plus the finite-element weighted contributions of $\partial\tilde{S}/\partial t$ from the coarse cells, plus the time and space averaged finite-element weighted contributions from the fine cells, i.e.,

$$\begin{aligned} Res_{SP}^{coarse} &= Res_{SP,\nabla\cdot U=0}^{coarse} + \text{coarse grid } \frac{\partial\tilde{S}}{\partial t} \text{ contributions} + \\ &\quad \frac{1}{R_{coarse}}\sum_{k=1}^{R_{coarse}} \text{fine grid } \frac{\partial\tilde{S}}{\partial t} \text{ contributions.} \end{aligned}$$

Note that the fine grid contributions are first computed at the fine nodes and then averaged to the coarse node. See Figure 3 for an example.

The remainder of (S.4) is identical to the same step in IAMR. The composite residual is combined with the divergence of the velocity corrections found in (S.3) to form the right hand side of a multilevel sync projection. Corrections to both the velocity and the pressure at the coarse and all finer levels result.

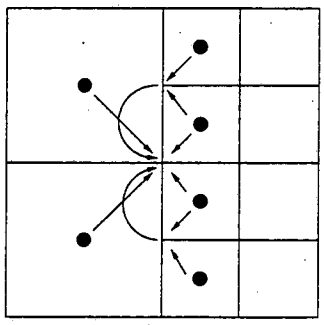

Figure 3: Schematic showing contributions of coarse and fine grid cell-centered values of $\partial\tilde{S}/\partial t$ to the node-based residual for a refinement ratio 2.

5 Computational Results

In this section we present numerical results demonstrating the methodology described above. Three methane-air flames are computed, two steady and the other flickering. These examples serve as an initial validation of the algorithm. In all cases, we use square computational cells ($\Delta r = \Delta z$) and a Courant number (see (3.1)) of .4.

For these computations, we consider two different compositional models. In the first, the gas is composed of three species:

$$CH_4, \text{ air, product.} \tag{5.1}$$

Thermochemical properties are defined by polynomial curve fits for $c_{p,ox}$, $c_{p,pr}$, (Rhine and Tucker, 1991) and $c_{p,fu}$ (Glasstone, 1947), and a heat of formation of 4.855×10^7 J/kg for natural gas (Rhine and Tucker, 1991). A one-step reaction mechanism (Khalil *et al.*, 1975) for methane oxidation is used:

$$CH_4 + 9.57 \text{ air} \rightarrow 10.57 \text{ product.} \tag{5.2}$$

The adiabatic flame temperature for this reaction is 2222 K for a base temperature of 298 K. The rate of fuel consumption is given by

$$-\omega_{fu} = \rho^2 Y_{fu} Y_{ox} A \exp\left(-E_a/\mathcal{R}T\right), \tag{5.3}$$

where $A = 10^{10}\text{m}^3/(\text{kg-sec})$ and $E_a/\mathcal{R} = 1.84 \times 10^4$ K (Khalil *et al.*, 1975). We refer to this compositional model and the accompanying reaction mechanism as model 1.

The second compositional model uses 6 species:

$$CH_4,\ O_2,\ N_2,\ CO,\ CO_2,\ H_20. \tag{5.4}$$

Enthalpies, heat capacities, and heats of formation are computed with GRI-Mech thermochemical data (Frenklach *et al.*, 1994). The following two-step reaction mechanism is used:

$$\begin{array}{lcl} CH_4 + {}^3\!/_2 O_2 & \rightarrow & CO + 2H_2O \\ CO + {}^1\!/_2 O_2 & \leftrightarrow & CO_2. \end{array} \tag{5.5}$$

We also consider a modification of this mechanism in which we neglect the reverse reaction in the CO oxidation step. The adiabatic flame temperature for the complete forward reaction is 2317 K for a base temperature of 298 K. We consider two different expressions for the rate of CH_4 oxidation, the first due to Zimont and Trushin (1969),

$$-\frac{d[CH_4]}{dt} = 10^{14.49}T^{0.5}\exp\left(-E_a/\mathcal{R}T\right)[CH_4][O_2] \text{ gmoles cm}^{-3}\text{sec}^{-1}, \tag{5.6}$$

where $E_a = 39895$cal/gmole, and the second due to Dryer and Glassman (1972),

$$-\frac{d[CH_4]}{dt} = 10^{13.2}\exp\left(-E_a/\mathcal{R}T\right)[CH_4]^{0.7}[O_2]^{0.8} \text{ gmoles cm}^{-3}\text{sec}^{-1}, \tag{5.7}$$

where $E_a = 48400$ cal/gmole. We use the following rate for the forward CO oxidation step (Dryer and Glassman, 1972),

$$-\frac{d[CO]}{dt} = 10^{14.6}\exp\left(-E_a/\mathcal{R}T\right)[CO][H_2O]^{0.5}[O_2]^{0.25} \text{ gmoles cm}^{-3}\text{sec}^{-1} \tag{5.8}$$

and the following reverse rate (Westbrook and Dryer, 1981),

$$-\frac{d[CO_2]}{dt} = 5\times 10^{8}\exp\left(-E_a/\mathcal{R}T\right)[CO_2] \text{ gmoles cm}^{-3}\text{sec}^{-1}, \tag{5.9}$$

where $E_a = 40000$ cal/gmole. We refer to the complete two-step mechanism with (5.6) as model 2 and with (5.7) as model 3. The corresponding models in which the reverse CO oxidation step is neglected are referred to as models 2n and 3n.

The five composition/mechanism/rate models are summarized in Table I.

Model	Composition	Reaction Mechanism	Reaction Rates
1	5.1	5.2	5.3
2	5.4	5.5	5.6, 5.8, 5.9
2n	5.4	5.5	5.6, 5.8, $d[CO_2]/dt = 0$
3	5.4	5.5	5.7, 5.8, 5.9
3n	5.4	5.5	5.7, 5.8, $d[CO_2]/dt = 0$

Table I: Compositional models, reaction mechanisms, and reaction rates used in the numerical examples.

In the results reported below, we follow the approach used by Smooke *et al.* (1989) and define flame length as the z-coordinate of the center of the cell along the axis of symmetry corresponding to the first temperature maximum. We use the same definition for the flame height of a lifted flame. We additionally follow the approach of Bennett and Smooke (1997) and define the lift-off height of a lifted flame as the cell-center z-coordinate of the cell closest to the inlet plane for which $T \geq 1000$ K.

5.1 Steady Laminar Methane-Air Diffusion Flame

The first example is the calculation of the steady, unconfined coflowing methane-air diffusion flame previously computed by Smooke *et al.* (1989). The experimental configuration is illustrated in Figure 4. The radius of the inner fuel jet is .2 cm and the radius of the coflowing air jet is 2.54 cm. At the inlet, the temperature is 298 K and the fuel velocity is $u = 0$, $v = 5.0$ cm/sec. The inlet air velocity is $u = 0$, $v = 25.0$ cm/sec; Re ≈ 60 for a reference length equal to the diameter of the fuel jet.

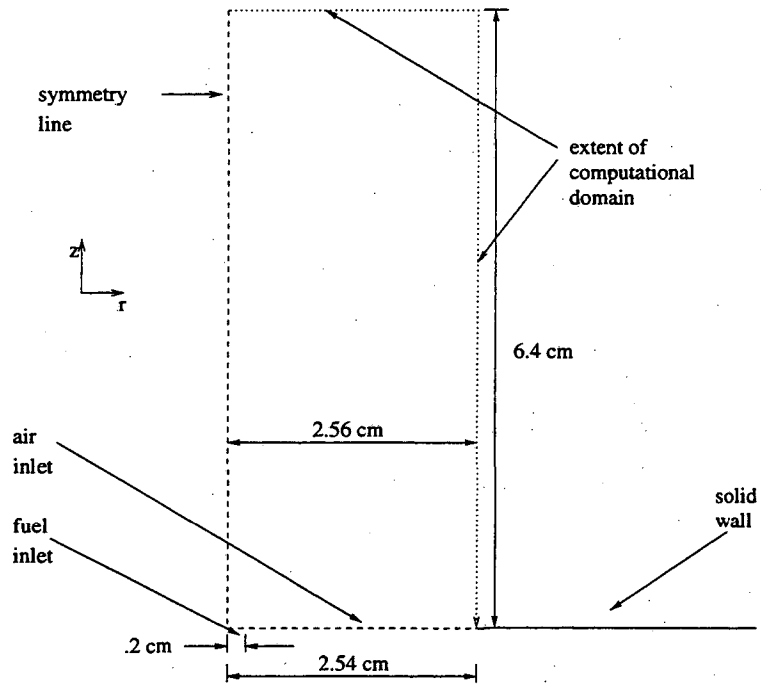

Figure 4: Sketch of specification of unconfined coflowing methane-air diffusion flame.

In our computation, the flame is ignited by a small hot patch ($T = 1500$K) next to the inlet. We use a 16×40 level 0 grid to cover a 2.56 cm by 6.4 cm problem domain. There are three additional levels of refinement. The refinement ratio $R_\ell = 2$ for $\ell = 0, 1, 2$, so that the equivalent uniform grid is 128×320. The inlet boundaries are refined to level 3 so that they align with level 3 grid lines. Additionally, the region $T > 1800$ K is refined to level 2.

We compute this flow with each of the five models in Table I. We first discuss results obtained using model 1. Figure 5 shows the early development of the flame. The unsteady phase is characterized by a vortex ring which appears as a "mushroom" shape in the plots. The ring forms due to the initial expansion of gas following ignition and ultimately rises out of the computational domain. The boundaries of the level 1, 2, and 3 grids are shown as thin lines in the plots. We note that because of the initial velocity projection and the use of a hot patch to ignite the flame, the figure is merely representative of the development of the flame at early time.

Figure 6 shows the flame at steady-state. We calculate a flame length and a maximum

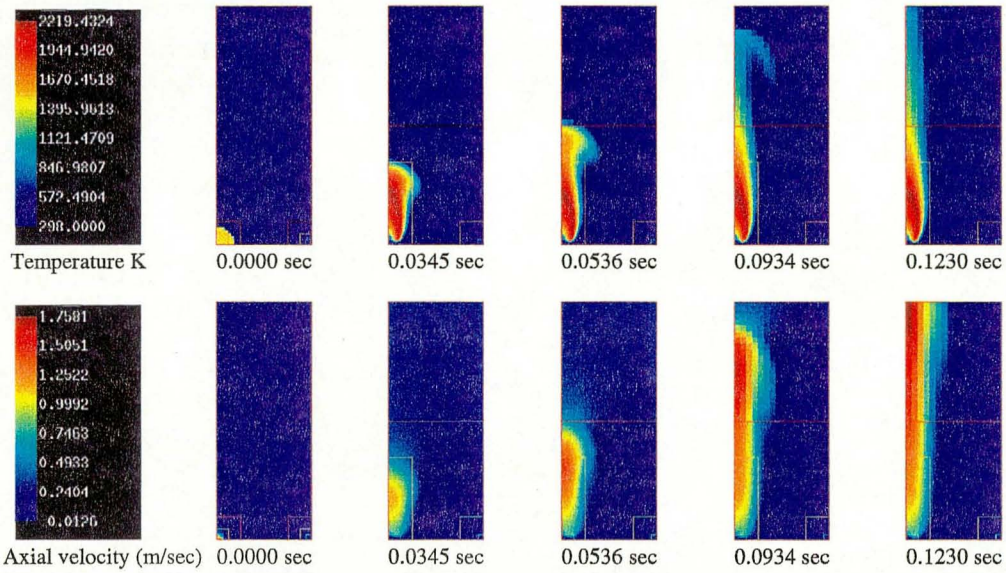

Figure 5: Unconfined coflowing methane-air laminar diffusion flame: early time. The boundaries of the level 1, 2, and 3 grids are shown as thin lines in the plots.

temperature of 1.43 cm and 2208 K, respectively; Smooke *et al.* compute values of 1.25 cm and 2053 K. Qualitatively, our calculation shows the same general flame shape and the same rapid increase of axial velocity along the centerline. We speculate that our temperatures may be higher due to using a reduced kinetics mechanism and/or species-independent mass diffusivities. Note that we have plotted ρRT to show how well the scheme meets the constraint $p_0 = \rho RT$. The two values differ significantly only along the edge of the flame, and the maximum percentage deviation from p_0 is less than 10%.

We now compare the solution obtained with model 1 with solutions for the other four models. Table II shows the values of flame length, maximum temperature, and maximum axial velocity, and the range of ρRT for each of these models. The results for models 2, 2n, 3, and 3n have higher maximum temperatures than model 1 because of the higher adiabatic flame temperature for the associated compositional model. Models 2 and 3, in turn, produce lower peak temperatures than models 2n and 3n due to the reverse CO oxidation step. The values are otherwise comparable. Figure 7 shows the temperature fields at steady state for the five models.

5.1.1 Timings

We now present timings of the code for model 1 for the steady laminar flame problem discussed above. Five cases are reported: a 16×40 base grid with three levels of refinement ($R_\ell = 2, \ell = 0, 1, 2$), a 32×80 base grid with one level ($R_0 = 4$), a 32×80 base grid with two levels ($R_\ell = 2, \ell = 0, 1$), a 64×160 base grid with one level ($R_0 = 2$), and a uniform 128×320 grid. In the adaptive cases, the inlets is refined to the finest level and the region $T > 2000$K is refined to level 2 or the finest level, whichever is smaller. The calculations are

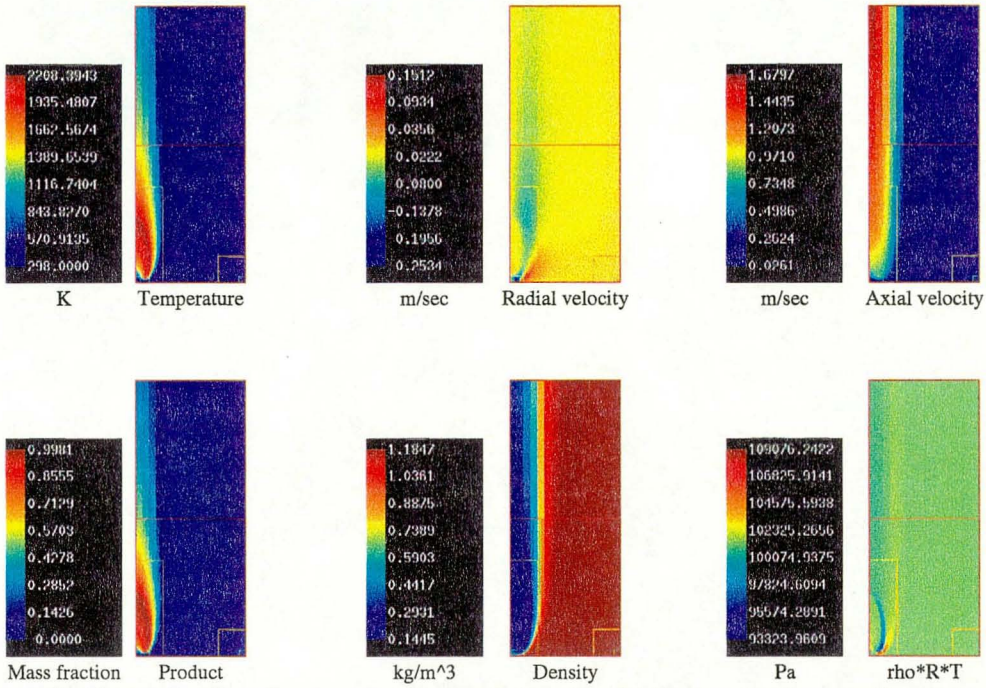

Figure 6: Unconfined coflowing methane-air laminar diffusion flame: steady state ($t = 0.442$ sec). The boundaries of the level 1, 2, and 3 grids are shown as thin lines in the plots. ρRT is plotted to show how well the scheme meets the constraint $p_0 = \rho RT$. The two values differ significantly only along the edge of the flame.

Model	T_{max} (K)	Flame length (m)	v_{max} (m/sec)	ρRT(kPa)
1	2208.4	.0143	1.680	93.3 – 109.1
2	2264.8	.0159	1.774	90.3 – 111.6
2n	2303.8	.0159	1.755	90.5 – 111.5
3	2270.5	.0143	1.703	91.5 – 109.5
3n	2310.5	.0143	1.688	91.8 – 109.6

Table II: Comparison of steady flame results for the five composition/mechanism/rate models.

all run on a single 300 MHz single processor of a four processor DEC Alpha workstation to a final time of .10412 sec. Table III shows the CPU time used to complete the calculation, the total number of cells advanced, the CPU time per cell, 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. The numbers show that the adaptive mesh refinement scheme can reduce the computational cost in terms of both CPU time and memory usage. For the examples run, however, the CPU time per cell does increase with the number of levels of refinement; the time for the level three case is nearly triple that of the level zero case. The results suggest that the refinement strategy used must be judicious; if too large a portion of the domain were refined, grid refinement would not lower the computational cost.

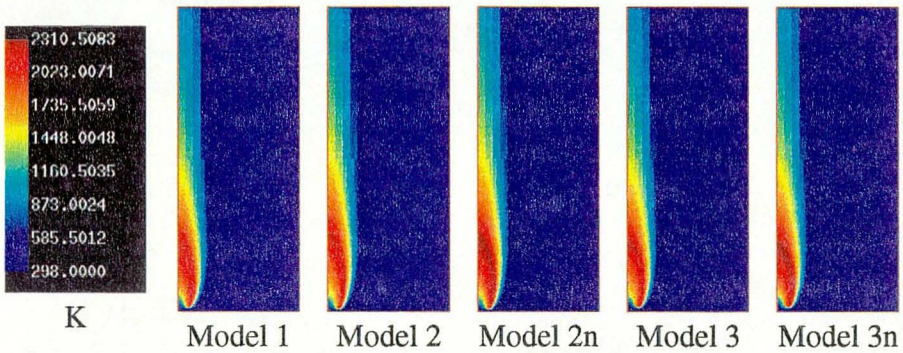

Figure 7: Steady state temperature of the unconfined coflowing methane-air laminar diffusion flame for the five composition/mechanism/rate models. Grid boundaries are not shown here.

Gridding	CPU Time		Cells Advanced	Peak Memory Usage
	Total(s)	μs/cell	Number	Mb
128×320, uniform	45810	615	74547200	33
64×160, $R_0 = 2$	13410	1004	13363072	16
32×80, $R_{0,1} = 2, 2$	7633	1151	6631680	10
32×80, $R_0 = 4$	5297	892	5941248	10
16×40, $R_{0,1,2} = 2, 2, 2$	3593	1716	2093568	9

Table III: Timings for uniform grid and refined grid calculations on a single processor of a four-processor DEC Alpha for the steady laminar flame problem presented in Section 5.1.

5.2 Steady Methane-Air Diffusion Flame with N_2-diluted Fuel

The second example is a steady, unconfined coflowing methane-air diffusion flame in which the fuel jet is composed of methane diluted with nitrogen. The experimental configuration is again illustrated by Figure 4. The fuel jet molar composition is 65% CH_4 and 35% N_2. The radius of the inner fuel jet is .2 cm and the radius of the coflowing air jet is 2.5 cm. At the inlet, the temperature is 298 K. The velocity of both inlet streams is $u = 0.0$, $v = 35.0$ cm/sec. Re ≈ 83 for a reference length equal to the diameter of the fuel jet.

This flow has been previously studied both experimentally (Smooke *et al.*, 1992) and computationally (Smooke *et al.*, 1992; Xu *et al.*, 1993; Smooke *et al.*, 1996; Bennett, 1997; Bennett and Smooke, 1997). Experimentally determined values for the flame include a liftoff height of .4 cm, a flame height of approximately 3.4 cm, and a maximum temperature of approximately 1949 K (Bennett, 1997). Computed values of the liftoff height vary from .34 cm (Bennett and Smooke, 1997) to nearly 1 cm (Smooke *et al.*, 1992) depending on, among other factors, the detailed reaction mechanism used. Additionally, the liftoff height is seen to depend on the resolution of the calculation (Bennett and Smooke, 1997). Maximum computed temperatures are roughly 1940 K (Smooke *et al.*, 1996; Bennett and Smooke, 1997) if radiative losses are accounted for, but jump to approximately 2040 K if these losses are neglected (Smooke *et al.*, 1992; Xu *et al.*, 1993). The computed flame heights are all approximately 3-3.5 cm.

In our computation, we neglect radiative losses. The flame is ignited by a small hot patch ($T = 1500$K) next to the inlet. We use a 16×48 level 0 grid to cover a 3.2 cm by 9.6 cm problem domain. There are three additional levels of refinement. The refinement ratio $R_\ell = 2$ for $\ell = 0, 1, 2$, so that the equivalent uniform grid is 128×384. The inlet boundaries are refined to level 3 so that they align with level 3 grid lines. The region $T > 1800$ K is also refined to level 3.

We compute the steady flame with models 2, 2n, 3, and 3n (see Table I) by timestepping to a steady state. The results are summarized in Table IV. The temperature fields for the four models are shown in Figure 8. Note the "wishbone"-like structure of the peak temperature region (we show half of the "wishbone"). Except for the maximum temperature, the results for models 2 and 2n, in particular, the flame shape and the liftoff and flame heights, agree better with the experimental results than those for models 3 and 3n. Figure 9 shows the mass fractions fields for O_2, H_2O, CO_2, and CO obtained with models 2 and 2n. The general structure of the mass fraction fields for O_2 and H_2O obtained with both models shows fairly good agreement with the reported experimental results (Bennett, 1997), although the values themselves show better agreement for model 2n. The CO_2 field for model 2n and the CO field for model 2 likewise compare favorably with experiment; the corresponding fields for models 2 and 2n, respectively, do not. For comparison, the ranges of the mass fractions found experimentally for O_2, H_2O, CO_2, and CO are 0.016-0.2304, 0.0007-0.1007, 0.0010-0.1477, and 0.000312-0.043998, respectively (Bennett, 1997).

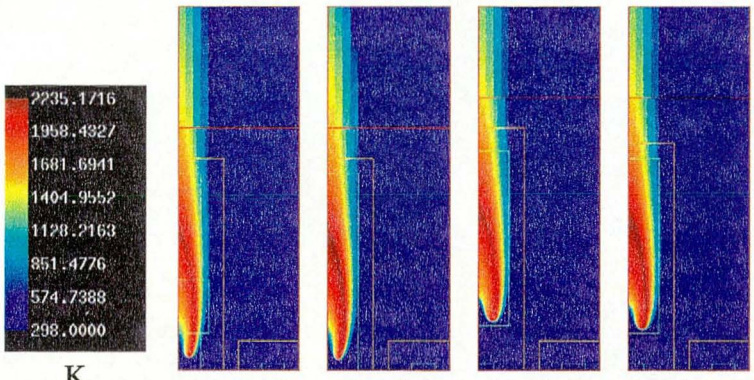

Figure 8: Steady state temperature of the unconfined coflowing N_2-diluted fuel, methane-air laminar diffusion flame for models 2, 2n, 3, and 3n. Grid boundaries are shown as thin lines.

Note that we have modeled the wall separating the fuel and air streams as having zero thickness. We performed additional calculations accounting for a finite wall thickness of .38 cm (Bennett, 1997). There were not significant differences in the two sets of results.

5.3 Flickering Methane-Air Diffusion Flame

The last example is the calculation of a flickering, unconfined coflowing methane-air diffusion flame. The computation models the coannular burner used by Smyth *et al.* (1993; 1994;

Model	T_{max} (K)	Flame liftoff height (m)	Flame height (m)
2	2196.2	0.00338	0.03538
2n	2231.5	0.00288	0.03463
3	2194.7	0.01288	0.04113
3n	2235.2	0.01288	0.03988

Table IV: Comparison of steady flame results for the four composition/mechanism/rate models.

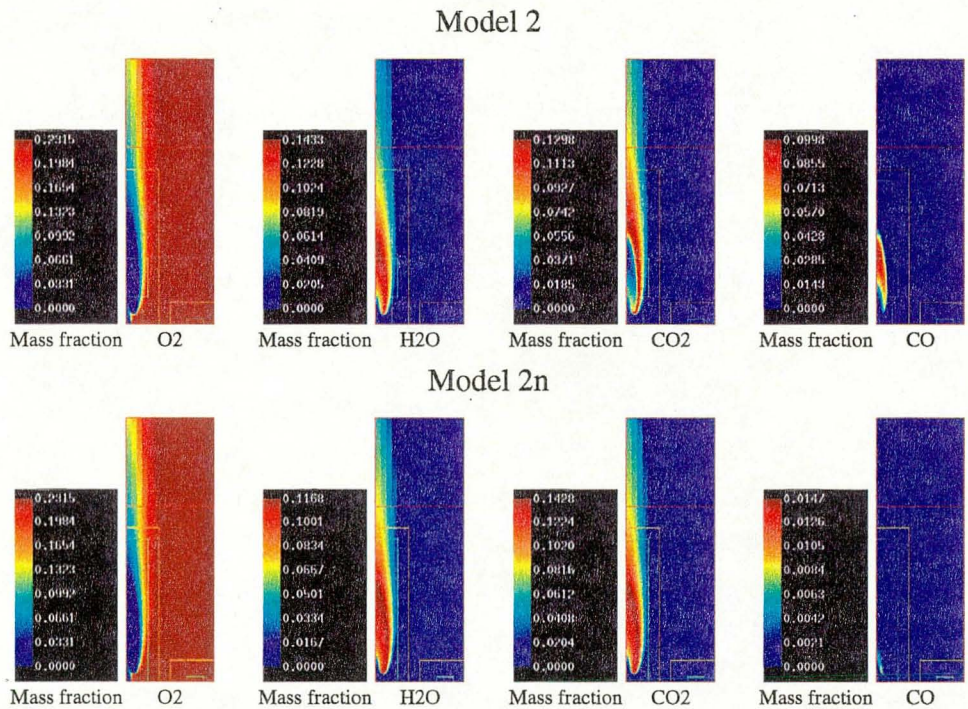

Figure 9: Steady state mass fraction fields of the unconfined coflowing N_2-diluted-methane/air laminar diffusion flame for models 2 and 2n. Grid boundaries are shown as thin lines.

1997) in a flame study performed to help develop better models of soot formation. They report results that include the effect of acoustic forcing (Smyth *et al.*, 1993) and those that do not (Smyth, 1994; Smyth, 1997). The latter case is the one computed here. Yam *et al.* (1995) have also simulated this flow using a single grid projection method.

The experimental configuration is conceptually similar to those modeled in the previous two sections. The coannular burner consists of a fuel inlet with a radius of .55 cm surrounded by an annulus of coflowing air with an outer radius of 5.1 cm. The velocity of both inlet streams is 7.9 cm/sec. Re $\approx$ 260 for a reference length equal to the diameter of the fuel jet. The flow for this configuration can be summarized as follows. During its early development, the flame grows in length and oscillates in a non-periodic manner. After a short time, the flame reaches a "steady-state" in which it exhibits a periodic oscillatory behavior best

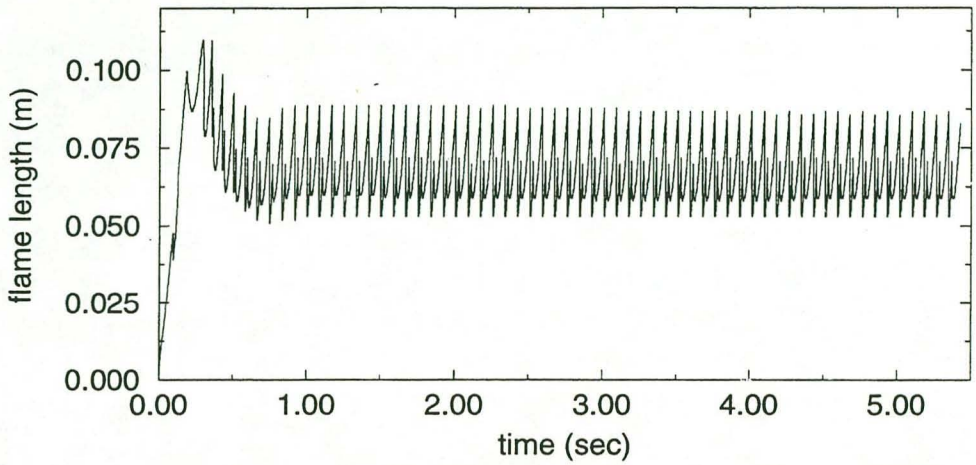

Figure 10: Axial position of the maximum temperature of the flickering flame along the centerline axis as a function of time.

described as flickering. The flame oscillations are caused by a buoyancy-induced Kelvin-Helmholtz type of instability.

In our computations, the flame is ignited by a small hot patch ($T = 1500$K) next to the inlet. We use a 16×64 level 0 grid to cover a 6.4 cm by 25.6 cm problem domain. There are three additional levels of refinement. The refinement ratio $R_\ell = 2$ for $\ell = 0, 1, 2$, so that the equivalent uniform grid is 128×512. The inlet boundaries and the region $T > 1800$ K are refined to level 3. Additionally, the region in which the magnitude of the vorticity exceeds 50 sec^{-1} is refined to level 1.

We compute the flow with each of the five models in Table I. All the computed flames establish periodic flickering by $t = 1$ sec. For each computed flame, we calculate the flickering frequency and the time-averaged flame length by using the complete flickering cycles (measured peak length to peak length) between $t = 1$ sec and $t = 2.5$ sec.

We first report results for model 1. Figure 10 shows a time history of the flame length. Figure 11 displays the temperature field during a single flame oscillation. We compute a flickering frequency of 11.94 Hz; Smyth *et al.* report a value of 12 Hz (Smyth, 1994). The computed time-averaged flame height is 6.66 cm; the experimental value is 7.9 cm. (The flame height reported by Smyth *et al.* is the axial location of the end of the soot burnout region, which is typically beyond the maximum temperature location (Smyth, 1997).) Yam *et al.* compute values of 15.7 Hz and 5.51 cm. As in the calculation reported in the previous section, we compute temperatures that are high compared to those previously reported; see the discussion above. We also compute a larger flame height oscillation (roughly 3 cm) than do Yam *et al.* (1 cm).

We now compare the results for the other four models with those for model 1. In Table V, we compare the flickering frequencies and flame lengths obtained using the models. The average flame lengths found with models 3 and 3n are longer than those for the other three models. The results are otherwise comparable. Figure 12 compares the temperature fields for the five models at comparable times during the flickering cycle. The shapes of the flames

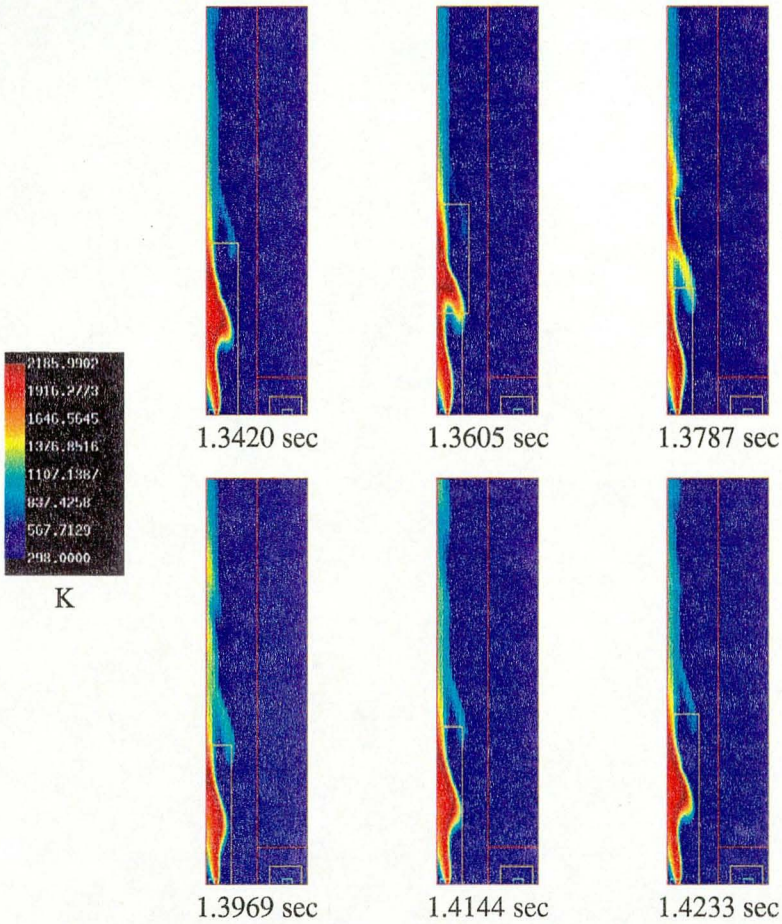

Figure 11: Temperature field of flickering flame during a single flame oscillation. The boundaries of the level 1, 2, and 3 grids are shown as thin lines in the plots.

agree fairly well. There are, however, secondary instabilities along the edges of the flame for models 2 and 3; we are uncertain why these features appear.

Model	Flickering Frequency (sec^{-1})	Flame length (m)
1	11.94	.0666
2	12.01	.0662
2n	11.92	.0664
3	12.13	.0684
3n	11.83	.0682

Table V: Comparison of flickering flame results for the five composition/mechanism/rate models.

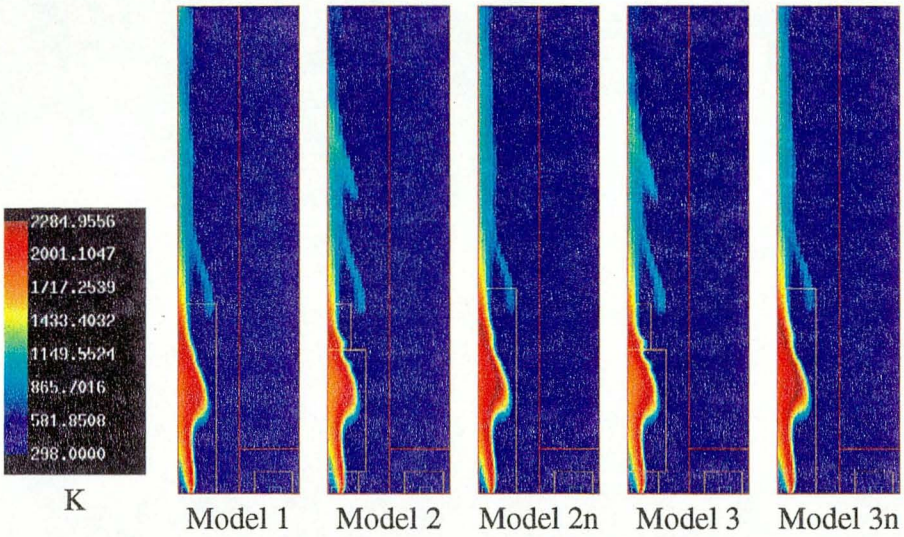

Figure 12: Temperature field of flickering flame for the five composition/mechanism/rate models at comparable times during the flickering cycle. The boundaries of the level 1, 2, and 3 grids are shown as thin lines in the plots.

6 Conclusions and Discussion

We have presented an adaptive projection method for computing unsteady, low-Mach number combustion. The adaptive mesh refinement scheme incorporates a higher-order projection methodology and uses a nested hierarchy of rectangular grids which are refined in both space and time. The algorithm is currently implemented for laminar, axisymmetric flames with a reduced kinetics mechanism and a Lewis number of unity. Numerical results for three test problems are favorable. The computed temperatures are higher, however, than those reported elsewhere for the same flows. We speculate that the high temperatures may be due to the use of a reduced kinetics mechanism and/or species-independent mass diffusivities.

Future directions for this work include developing automatic refinement criteria, incorporating detailed chemistry and species dependent mass diffusivities, accounting for radiative heat transfer (Howell *et al.*, 1998), and extending the methodology to three-dimensional and turbulent flows and to realistic engineering geometries.

Acknowledgments. 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.

The code which implements the algorithm described in this paper is derived from the IAMR code (Almgren *et al.*, 1998) and shares a large common code base with it, including BOXLIB (Crutchfield and Welcome, 1993). The authors would like to thank the following people: Ann Almgren, Vincent Beckner, Marc Day, Charles Rendleman, David Stevens, and Michael Welcome for their contributions to the development of IAMR code; Ann Almgren

for numerous enhancements to IAMR which facilitated this research; Vincent Beckner, Michael Lijewski, Charles Rendleman, and Michael Welcome for their work on BOXLIB.

NOMENCLATURE

$c_{p,l}(T)$	specific heat of species l at $p = p_0$
$c_p(T)$	specific heat of the gas mixture at $p = p_0$
D	molecular mass diffusivity
D/Dt	$\partial/\partial t + U \cdot \nabla$
E_a	activation energy in Arrhenius law
fu	subscript denoting fuel
Gp	a cell-centered gradient for a node-based pressure p
g	magnitude of acceleration due to gravity: 9.81 m/sec^2
h	enthalpy of gas mixture, $\sum_l h_l(T)Y_l$
$h_l(T)$	specific enthalpy of species l at $p = p_0$, including the heat of formation
i, j	cell indices in $r-$, $z-$ directions
l	subscript denoting species
Le	Lewis number, Sc/Pr $= \lambda/\rho D c_p$
M	Mach number
ox	subscript denoting oxidizer
p	pressure
p_0	ambient pressure: 101325 N/m^2
Pr	Prandtl number, $\mu c_p/\lambda$
pr	subscript denoting product
$\mathcal{R}$	universal gas constant
R	gas constant of mixture
R_ℓ	ratio of level $\ell + 1$ cell widths to the level ℓ cell widths
r_i	r-coordinate of center of cell ij, $i\Delta r$
$r_{i+1/2}$	r-coordinate of upper r-edge of cell ij
Re	Reynolds number, $\rho U L/\mu$
r	radial coordinate
S	right hand side of divergence constraint
$\tilde{S}$	right hand side of the numerical divergence constraint
Sc	Schmidt number, $\mu/\rho D$
T	temperature
t^n	time at the end of the n-th time step
$t^{n+1/2}$	$t^n + \Delta t/2$
U	velocity

u	radial component of velocity
v	axial component of velocity
Y_l	mass fraction of species l
z	axial coordinate

Greek symbols

Δr	cell width in r-direction
Δz	cell width in z-direction
Δt	time step used to advance solution from t^n to t^{n+1}
λ	thermal conductivity
μ	viscosity
π	dynamic pressure, $p - p_0$
ρ	density
τ	stress tensor
ω_l	specific mass production rate of species l by chemical reactions

Subscripts and superscripts

$(\cdot)_{ij}^{n}$	value at center of cell ij at time t^n or average value over cell ij at t^n
$u_{ij}^{n+1,*}, v_{ij}^{n+1,*}$	axial and radial components of velocity before enforcement of divergence constraint
$(\cdot)_{ij}^{n+1,p}$	predicted value at center of cell ij at time t^n
$(\cdot)_{ij}^{n+1/2}$	value at center of cell ij at time $t^n + \Delta t/2$
$(\cdot)_{i+1/2,j}^{n+1/2}$	value at upper r-edge of cell ij at time $t^n + \Delta t/2$
$(\cdot)_{i,j+1/2}^{n+1/2}$	value at upper z-edge of cell ij at time $t^n + \Delta t/2$
$(\cdot)_{i+1/2,j+1/2}^{n+1/2}$	value at upper corner of cell ij at time $t^n + \Delta t/2$
$(\cdot)_{ij}^{n+1}$	value at center of cell ij at time $t^n + \Delta t$

Other

$[\cdot]$	molar concentration, gmoles/cm^3

References

Almgren, A. S., Bell, J. B., Colella, P., Howell, L. H., and Welcome, M. L. (1998). A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, accepted for publication. Also as LBNL-39075, Lawrence Berkeley National Laboratory, July, 1996, and *http://www.nersc.gov/research/CCSE/publications/almgren/abchw96/paper.ps.gz.*

Almgren, A.S., Bell, J.B., Colella, P., Howell, L. H., and Welcome, M.L. (1995). A high-resolution adaptive projection method for regional atmospheric modeling. In: *Proceedings of the NGEMCOM Conference,* sponsored by the U.S. EPA, August 7–9, 1995.

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

Bell, J. B., and Marcus, D. L. (1992). A second-order projection method for variable-density flows. *J. Comput. Phys.*, **101**, 334–348.

Bell, J. B., Colella, P., and Howell, L. H. (1991). An efficient second-order projection method for viscous incompressible flow. In: *10th AIAA Computational Fluid Dynamics Conference.* Honolulu, June 24–27, 1991.

Bell, J.B., Colella, P., and Glaz, H.M. (1989). A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, **85**, 257–283.

Bennett, B.A.V. (1997). *Solution-Adaptive Gridding Methods with Application to Combustion Problems.* Ph.D. thesis, Yale University.

Bennett, B.A.V., and Smooke, M.D. (1997). Local rectangular refinement with application to fluid flow problems. Submitted to *J. Comput. Phys.* Report ME-101-97, Dept. of Mech. Eng., Yale University September, 1997.

Berger, M.J., and Colella, P. (1989). Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, **82**, 64–84.

Berger, M.J., and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, **53**, 484–512.

Chorin, A. (1969). Numerical solution of the Navier-Stokes equations. *Math. Comp.*, **22**, 745–762.

Coelho, P.J., and Pereira, J.C.F. (1993). Calculation of a confined axisymmetric laminar diffusion flame using a local grid refinement technique. *Combust. Sci. Tech.*, **92**, 243–264.

Colella, P. (1990). Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, **87**, 171–200.

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

de Lange, H.C, and de Goey, L.P.H. (1994). Numerical flow modeling in a locally refined grid. *Inter. Jour. Num. Meth. Eng.*, **37**, 497–515.

Dryer, F.L., and Glassman, I. (1972). High-temperature oxidization of co and ch_4. Pages 987–1003 of: *Fourteenth Symposium (International) on Combustion.* The Combustion Institute, 1972.

Dwyer, H. (1990). Calculation of low Mach number reacting flows. *AIAA Journal*, **28**(1), 98–105.

Frenklach, M., Wang, H., Bowman, C.T., Hanson, R.K., Smith, G.P., Golden, D.M., Gardiner, W.C., and Lissianski, V. (1994). *An Optimized Kinetics Model for for Natural Gas Combustion.* 25th International Symposium on Combustion, Irvine, California, 1994, Work-In-Progress PosterSession 3, Number 26.

Glasstone, S. (1947). *Thermodynamics for Chemists.* van Nostrand Co., Princeton.

Harlow, F., and Welch, J. (1964). Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces. *Phys.Fl.*, **8**, 2182–2189.

Hilditch, J., and Colella, P. (1996). A projection method for low Mach number fast chemistry reacting flow. AIAA 35th Aerospaces Meeting, Reno, NV, Jan., 1996.

Howell, L. H., Pember, R. B., Colella, P., Fiveland, W. A., and Jessee, J. P. (1998). A conservative adaptive-mesh algorithm for unsteady, combined-mode heat transfer using the discrete ordinates method. Submitted to *Num. Heat Trans., Pt. B.*

Kanuary, A.M. (1982). *Combustion Phenomena.* Gordon & Breach, New York.

Khalil, E.E., Spalding, D.B., and Whitelaw, J.H. (1975). The calculation of local flow properties in two-dimensional furnaces. *Int. J. Heat and Mass Transfer*, **18**, 775–791.

Kuo, K.K. (1986). *Principles of Combustion.* Wiley-Interscience, New York.

Lai, M. (1993). *A Projection Method for Reacting Flow in the Zero Mach Number Limit.* Ph.D. thesis, University of California, Berkeley. Also published as CPAM-601, University of California, Berkeley, February, 1994.

Lai, M., Bell, J.B., and Colella, P. (1993). A projection method for combustion in the zero Mach number limit. In: *Proceedings, AIAA 11nd Computational Fluid Dynamics Conference, Orlando, FL.* July 6-9, 1993, p.776-783, AIAA Paper 93-3369-CP.

LeVeque, R.J. (1990). *Numerical Methods for Conservation Laws.* Birkhauser, Boston.

Majda, A., and Sethian, J.A. (1985). The derivation and numerical solution of the equations for zero Mach number combustion. *Combust. Sci. Tech.*, **42**, 185–205.

Mallens, R.M.M., de Lange, H.C., van de Ven, C.H.J., and de Goey, L.P.H. (1995). Modeling of confined and unconfined laminar premixed flames on slit and tube burners. *Combust. Sci. Tech.*, **107**, 387–401.

Najm, H.N. (1996a). A conservative low Mach number projection method for reacting flow modeling. Pages 921–932 of: *Transport Phenomena in Combustion.* Taylor & Francis.

Najm, H.N. (1996b). A mass-conservative projection method for confined low Mach number reacting flow. Submitted to *AIAA Journal.*

Pember, R.B., Almgren, A.S., Crutchfield, W.Y., Howell, L.H., Bell, J.B., Colella, P., and Beckner, V.E. (1995). *An Embedded Boundary Method for the Modeling of Unsteady Combustion in a Gas-Fired Furnace.* Fall Meeting, Western States Section of the Combustion Institute, 1995, WSS/CI 95F-165.

Pember, R.B., Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., and Lai, M. (1996). A higher-order projection method for the simulation of unsteady turbulent nonpremixed combustion in an industrial burner. Pages 1200–1211 of: *Transport Phenomena in Combustion.* Taylor & Francis.

Rehm, R.G., and Baum, H.R. (1978). The equations of motion for thermally driven buoyant flows. *N.B.S.J.Res.*, **83**, 297–308.

Rhine, J.M., and Tucker, R.J. (1991). *Modelling of Gas-Fired Furnaces and Boilers.* British Gas, McGraw-Hill.

Smooke, M.D., Turnbull, A.A., Mitchell, R.E., and Keyes, D.E. (1988). Solution of two-dimensional axisymmetric laminar diffusion flames by adaptive boundary value methods. Pages 261–300 of: Brauner, C.-M., and Schmidt-Lainé, C. (eds), *Mathematical Modeling in Combustion and Related Topics.* Dordrecht: Martinus Nijhoff Publishers.

Smooke, M.D., Mitchell, R.E., and Keyes, D.E. (1989). Numerical solution of two-dimensional axisymmetric laminar diffusion flames. *Combust. Sci. Tech.*, **67**, 85–122.

Smooke, M.D., Xu, Y., Zurn, R.M., Lin, P., Frank, J.H., and Long, M.B. (1992). Computational and experimental study of OH and CH radicals in axisymmetric laminar diffusion flames. Pages 813–821 of: *Twenty-Fourth Symposium (International) on Combustion.* The Combustion Institute, 1992.

Smooke, M.D., Ern, A., Tanoff, Valdati, B.A., Mohammed, R.K., Marran, D.F., and Long, M.B. (1996). Computational and experimental study of NO in an axisymmetric laminar diffusion flame. Pages 813–821 of: *Twenty-Sixth Symposium (International) on Combustion.* The Combustion Institute, 1996.

Smyth, K.C. (1994). *Flickering Flames as Testing Grounds for Reliable Models of Gas Combustion.* Project Advisor Group Meeting and 1994 Contractors Review of Combustion Chemistry Team and Turbulent Flame Burners Team, GRI, Chicago, IL, May 17-18, 1994.

Smyth, K.C. (1997). Private communication.

Smyth, K.C., Harrington, J.E., Johnsson, E.L., and Pitts, W.M. (1993). Greatly enhanced soot scattering in flickering CH_4/air diffusion flames. *Combust. Flame*, **95**, 229–239.

Somers, L.T., and de Goey, L.P.H. (1995). A numerical study of a premixed flame on slit burner. *Combust. Sci. Tech.*, **108**, 121–132.

Trangenstein, J.A., and Bell, J.B. (1989). Mathematical structure of the black-oil model for petroleum reservoir simulation. *SIAM J. Appl. Math.*, **49**, 749–783.

van Leer, B. (1979). Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, **32**, 101–136.

Wesseling, P. (1992). *An Introduction to Multigrid Methods.* John Wiley & Sons.

Westbrook, C.K., and Dryer, F.L. (1981). Simplified reaction mechanisms for the oxidation of hydrocarbon fuels in flames. *Combust. Sci. Tech.*, **27**, 31–43.

Xu, Y., Smooke, M.D., Lin, P., and Long, M.B. (1993). Primitive variable modeling of multidimensional laminar flames. *Combust. Sci. Tech.*, **90**, 289–313.

Yam, C.G., Marx, K.D., Chen, J.-Y., and Chen, C.-P. (1995). *Numerical Study of Flickering Frequency and Emission Index of a Methane Diffusion Flame for Varying Gravitational Force.* Fall Meeting, Western States Section of the Combustion Institute, 1995, WSS/CI 95F-223.

Zimont, V. L., and Trushin, Y. M. (1969). Total combustion kinetics of hydrocarbon fuels. *Comb. Expl. Shock Wav.*, **5**, 391–394.

Appendix A: Details of Tensor Level Solves

We present here the details of the tensor level solve used in solving both the difference equation (3.10) during a single level advance and the corresponding difference equation used in step (S.3) during the adaptive synchronization step. The discussion below is for two-dimensional rectangular coordinates; the extension to cylindrical coordinates is straightforward.

Unlike viscous velocity solves in a homogeneous constant-temperature medium, the algorithm presented in this paper require solving a parabolic tensor equation. The goal is to solve an equation of the form

$$\alpha(x)\vec{v} - \nabla \cdot (\beta(x)\tau(\vec{v})) = rhs \tag{6.1}$$

where τ is the tensor

$$\tau(\vec{v})_{ij} = v_{i,j} + v_{j,i} \quad . \tag{6.2}$$

In practical application, $\beta(x)$ would be viscosity, which is position dependent because of temperature variations.

In most respects, this parabolic tensor equation may be solved in exact analogy with the scalar cell-centered level solves discussed in Almgren *et al.* (1998). Both are cell-centered single-level solves defined on the union of rectangles. The system is solved using standard multigrid methods (V-cycles with multi-color Gauss-Seidel relaxation). The restriction operator is volume-weighted averaging; the multigrid interpolation is piecewise constant. In the following, we will concentrate upon the single difference: the discretization of the operator near the boundaries of each individual rectangle in the union.

We use a finite-volume discretization of Equation 6.1, so that the term $\nabla \cdot (\beta(x)\tau(\vec{v}))$ is represented by differences of $\beta\tau(\vec{v})$ evaluated upon the faces of a unit cell. $\tau(\vec{v})$ contains

both derivatives which are normal to the cell face, and derivatives which are tangential to the cell face. The normal derivatives may be treated in exact analogy to the treatment in the IAMR algorithm (Almgren *et al.*, 1998) and will not be further discussed. In the interior of the rectangles, where the finite difference stencil is completely contained within the rectangle, the tangential derivative is computed with an "H-shaped" stencil, for example

$$(\frac{\partial u}{\partial y})_{i+1/2,j} = \frac{u_{i+1,j+1} + u_{i,j+1} - u_{i+1,j-1} - u_{i,j-1}}{4\Delta y} \tag{6.3}$$

Care is required in computing the tangential derivative when the "H-shaped" stencil extends outside one of the rectangles.

In IAMR, the operator is evaluated in the outer row of cells in a rectangle by placing second-order accurate values in a row of cells immediately exterior to the rectangle (ghost cells) and applying the same stencil operator as is applied in the interior. Values are provided for these "ghost" cells from one of three possible sources: 1) copying from adjacent rectangles in the union of rectangles; 2) interpolation from the next coarsest level of refinement; 3) application of physical boundary conditions. Unfortunately, the straightforward use of ghost cells will provide inconsistent values of the tangential derivatives. Figure 13 shows that using ghost cells will cause two adjoining grids to compute different values of the same tangential derivative. Suppose it is desired to compute an x-derivative at the location of the solid circle. Because this location is shared by both rectangle 1 and 2, it is necessary for consistency that both grids compute the same value for the x-derivative. The "H-stencil" will require values at the locations of the open circles. One of the open circles is not covered by either rectangle 1 or 2, and must be filled by interpolation. As explained in Almgren *et al.* (1998), computations on rectangle 1 will fill in a ghost-cell value using coarse cell values at a, b, and c, plus the fine grid values indicated with small triangles. However, computations on rectangle 2 will fill in a ghost-cell values using coarse cells values at a, d, and e, plus the fine grid values indicated with small squares. Both values for the ghost cell will be second-order accurate, but they will not, in general, be identical. This will lead to different values for the shared wall flux.

In order to maintain consistency of tangential derivatives computed on different rectangles, we will avoid ghost cells in computing tangential derivatives, and instead modify the stencil where appropriate. Our general principle is to utilize fine grid information when it is available from other rectangles. If there is not enough information to evaluate the H-stencil, the stencil will be modified to use one-sided differences which are totally contained within the union of rectangles. If there is not enough fine level data to support the one-sided differences, then derivative information is interpolated from a coarser level, or from physical boundary conditions. Mask arrays are maintained with each rectangle of the union that indicate if adjoining cells are covered by fine grid data.

Consider first cell edges which are located on the perimeter of the rectangle. The edge derivative is computed as linear interpolation of 1) a cell centered derivatives located in the cell just interior to the edge, and 2)a derivative centered exterior to the rectangle. For example,

$$\frac{\partial u}{\partial y}_{i+1/2,j} = \frac{\lambda - 1/2}{\lambda}\frac{\partial u}{\partial y}_{i,j} + \frac{1}{2\lambda}\frac{\partial u}{\partial y}_{i+\lambda,j} \tag{6.4}$$

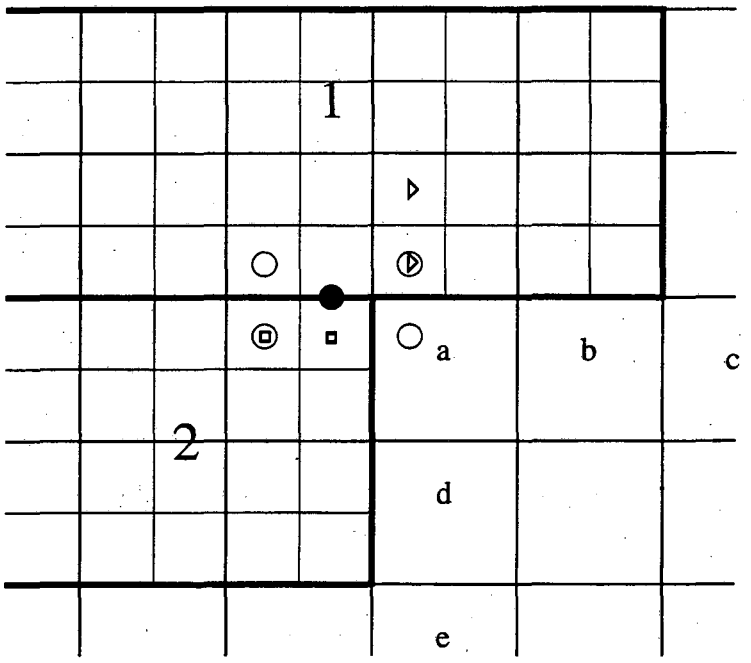

Figure 13: Where two rectangles in a level adjoin each other, each rectangle may compute a different value for a tangential derivative if ghost cells are used. In this 2D example, the horizontal derivative is needed at the cell edge indicated by the solid circle. The H-stencil of the vertical derivative requires values at locations indicated by open circles. In rectangle 1, the values at coarse grid locations a, b, and c and the fine grid locations indicated by triangles contribute to the ghost cell value. In rectangle 2, coarse grid locations a, d, and e and the fine grid locations indicated by squares contribute.

where λ parameterizes the location of the derivative centered exterior to the rectangle. When fine grid data is available exterior to the rectangle, λ would be one. When coarse level data is used, λ would be determined by the location of the coarse cell centers. With obvious meaning, we will refer to the derivatives on the right-hand side of the above as the inside and outside derivatives. Linear interpolation will provide a second order accurate approximation to the derivative if the inner and outer derivatives are second order accurate.

In the case of the inside derivative, we compute it with centered differences unless the centered stencil requires a cell value which is not found on the fine level. In the case where the centered difference cannot be used, a second-order accurate one-sided derivative whose stencil is contained within the the rectangle is used. Rectangles are not allowed to become small enough that the one-sided derivative is not covered by the rectangle. In the case of the outer derivative, we consider the same sequence of possible stencils: first the centered difference and then two possible one-sided differences. If none of these three possible stencils are usable, the outer derivative is computed by a second-order accurate interpolation from the coarse grid. In this last case, consistency of the tangential derivative is not a problem because two rectangles are not adjoining at this point.

We must also compute tangential derivatives on cell edges which are not on the perimeter of the rectangle. However, since none of these edges are shared between rectangles, the problem of consistency does not arise. It should be possible to use ghost cells in the computation of these tangential derivatives. However, to maintain consistency with the programming structure used for the tangential derivatives on the perimeter, we continue to use modified stencils for these derivatives as well.

These modified stencils produce second-order accurate approximations to the tangential derivatives, which reduce the accuracy of the parabolic operator at some of the boundary cells to first order, compared to the second order accuracy in the interior of the rectangles. However, since the first-order errors are localized at the boundary of the union of rectangles, the overall scheme is still second order.

Ernest Orlando Lawrence Berkeley National Laboratory
One Cyclotron Road | Berkeley, California 94720

Prepared for the U.S. Department of Energy under Contract No. DE-AC03-76SF00098