LBL-38551
UC-405

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

An Adaptive Projection Method for the Modeling of 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 Jesse

Computing Sciences Directorate

October 1997

To be presented at the
Western States Section
of the Combustion Institute,
1997 Fall Meeting
Diamond Bar, CA,
October 23–24, 1997
Paper WSS/CI 97F-125

REFERENCE COPY
Does Not
Circulate
Bldg. 50 Library - Ref.
Lawrence Berkeley National Laboratory
LBL-38551
Copy 1

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.

An Adaptive Projection Method for the Modeling of Unsteady, Low-Mach Number Combustion[1]

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

Woodrow A. Fiveland and J. Patrick Jesse
Babcock & Wilcox
Alliance, OH 44601

Abstract

In this paper we present an adaptive projection method for modeling unsteady, low-Mach reacting flow in an unconfined region. The equations we solve are based on a model for low-Mach number combustion that consists of the evolution equations for density, species concentrations, enthalpy, and momentum 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, hierarchical grid structure composed of uniform rectangular grids of varying resolution. The integration scheme 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 multiple levels of refinement.

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

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 the use of globally refined, nonuniform grids.

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 low-Mach number reacting flow in an unconfined region. This system is based on a generalization of the low-Mach number combustion model in [29, 23]. The system includes

[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]Contact information: MS-50D, LBNL, 1 Cyclotron Road, Berkeley, CA 94720, rbpember@lbl.gov, (510) 486-4180

evolution equations 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 structure approach first developed by Berger and Oliger [10] and Berger and Colella [9] 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 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 methodology presented here is based on a single grid algorithm developed by Pember *et al.* [27, 28]. 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 in [3, 1]. 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, while the predictor-corrector formulation guarantees second-order accuracy in time. A simple extension of the second-order approximate projection algorithm in [3, 1] to low-Mach number compressible flows is employed to enforce the divergence constraint and update the pressure. A pressure relaxation term is added to the numerical representation of the divergence constraint to account for the fact that the sequential approach cannot simultaneously conserve mass and satisfy the equation-of-state.

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.* [2, 1]. 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 in [1] 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 two numerical examples, a steady methane-air diffusion flame [30] and a flickering methane-air flame [33, 39, 32], are presented.

There are numerous references to the use of globally refined, non-uniform grids in combustion modeling. We refer the reader to Bennett [7], Bennett and Smooke [8], and the references therein. Local adaptive mesh refinement and local rectangular refinement meth-

ods have been used to model steady, low-Mach number combustion. In addition to the two references above, see Coelho and Pereira [11], de Lange and de Goey [13], Mallens *et al.* [24], Smooke *et al.* [31], and Somers and de Goey [34]. 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 used in the unsteady case; see Dwyer [14], Lai [20], Lai *et al.* [21], Najm [25, 26], and Yam *et al.* [39].

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[29, 23], which we now briefly review. (See Table 1 for the nomenclature.)

For flow in a spatially open domain the underlying assumption in the low-Mach number model is that M is sufficiently small (say $M < .3$) so 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:

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

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

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

$c_{p,l}(T)$	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
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
l	subscript denoting species, fu (fuel), ox (oxidizer), pr (product)
Le	Lewis number, $\mathrm{Sc}/\mathrm{Pr} = \lambda/\rho D c_p$
M	Mach number
p	pressure
p_0	ambient pressure: 101325 N/m^2
Pr	Prandtl number, $\mu c_p/\lambda$
$\mathcal{R}$	universal gas constant
R	gas constant of mixture
Re	Reynolds number, $\rho U L/\mu$
r	radial coordinate
S	right hand side of divergence constraint
Sc	Schmidt number, $\mu/\rho D$
T	temperature
U	velocity
u	radial component of velocity
v	axial component of velocity
Y_l	mass fraction of species l
z	axial coordinate
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

Table 1: Nomenclature: Physical Model

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

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

In this system, equations (2.6) and (2.7) are redundant because the enthalpy h is defined by

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

Numerically, 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). Moreover, in our numerical method, 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

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

The system of equations are overdetermined in two other ways. Equations (2.4) and (2.8) are redundant because $\rho = \sum_l \rho Y_l$. We account for the redundancy numerically by computing $\nabla \cdot \rho U$ as $\sum_l \nabla \cdot \rho U Y_l$. By using (2.4) in addition to (2.8), we are able to use a simpler discretization of (2.8) and thereby use a simpler solution strategy. Equations (2.4) and (2.2) are also redundant. We use (2.4) to ensure conservation of mass. The sequential approach used numerically makes it impossible, in general, to simultaneously satisfy the continuity equation and the equation of state; see §3.1 for further discussion.

For the calculations shown in this paper, a one-step reaction model [18] for methane oxidation is used:

$$\text{natural gas} + 9.57 \text{ air} \rightarrow 10.57 \text{ product}. \tag{2.11}$$

The rate of fuel consumption is given by

$$-\omega_{fu} = \rho^2 Y_{fu} Y_{ox} A \exp\left(-E_a/\mathcal{R}T\right), \tag{2.12}$$

where $A = 10^{10}\text{m}^3/(\text{kg-sec})$ and $E_a/\mathcal{R} = 1.84 \times 10^4$ K. Polynomial curve fits are used for $c_{p,ox}$, $c_{p,pr}$, [36] and $c_{p,fu}$ [15]. We use a heat of formation of 4.855×10^7J/kg for natural gas[36]. The viscosity, μ, is computed by the curve fit $\mu = \mu_0(T/T_0)^{.7}$ [17], 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 $\rho D = \lambda/c_p = \mu/\text{Pr}$. We use a fixed Prandtl number of .7.

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 [28] for the case of simple boundaries and incorporates many of the details of the single grid algorithm described in [1]. The reader is referred to [6, 4, 5, 3, 27] for additional discussion. We use a uniform grid of rectangular cells with widths Δr and Δz indexed by i and j. (See Table 2 for the nomenclature.) 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 [6]. The overall approach, then, is that of a fractional step scheme. In the first step (which we call the advection-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

i, j	cell indices in $r-$, $z-$ directions
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}
Gp	a cell-centered gradient for a node-based pressure p
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
$\tilde{S}$	right hand side of the numerical divergence constraint
t^n	time at the end of the n-th time step
$t^{n+1/2}$	$t^n + \Delta t/2$
$(\cdot)_{ij}^n$	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 $t^n + \Delta t$

Table 2: Nomenclature: Numerical Algorithm

the second step (the projection step), the divergence constraint is imposed on the velocity via a node-based projection [3].. This step yields U^{n+1} and $p_{i+1/2,j+1/2}^{n+1/2}$, the pressure at $t^{n+1/2}$.

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

(1) Compute Δt:

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

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

(2) Compute discrete approximations of the convective terms in the governing equations at time $t^n + \Delta t/2$:

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

(3) Compute

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

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

(4) Compute predicted values $\varphi^{n+1,p}$ of the solution at t^{n+1} for the flow quantities φ, $\varphi = Y_l, T$, and h using the Crank-Nicholson method.

(5) Compute corrected the values of Y_l, T, h, and (u^*, v^*) to provide the solution at time t^{n+1} again using Crank-Nicholson differencing.

In step (2), a MAC projection [16] 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. 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 [38, 1]. 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 the above algorithm in more detail.

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 due to the fact that the sequential approach cannot simultaneously conserve mass and enforce the constraint $p_0 = \rho R T$; 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 [23].) 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,

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

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 [19] by $(\tilde{p}_{ij} - p_{amb})/\Delta t$, rewriting the resultant equation in terms of T, and using (2.3). The term $f(\tilde{p}_{ij} - p_{amb})/\Delta t$ acts to drive the solution back to the constraint $\tilde{p}_{ij} = p_{amb}$. Similar treatments have been used in numerical petroleum reservoir simulation [35].

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 Advection-Diffusion-Reaction Step

3.2.1 Computation of convective derivatives

The approximation of the convective derivatives generally follows the approach described in [1]; see [4] for additional discussion. There are two primary components to this computation: a higher-order upwind scheme [12] to determine edge states and a MAC projection [16] 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 $v^{n+1/2}_{i+1/2,j}$, and $u^{n+1/2}_{i,j+1/2}$ 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 $u^{n+1/2}_{i+1/2,j}$, $v^{n+1/2}_{i+1/2,j}$, $u^{n+1/2}_{i,j+1/2}$, and $v^{n+1/2}_{i,j+1/2}$, and compute $T^{n+1/2}_{i+1/2,j}$, $T^{n+1/2}_{i,j+1/2}$, $(\rho Y_l)^{n+1/2}_{i+1/2,j}$, $(\rho Y_l)^{n+1/2}_{i,j+1/2}$, $(\rho h)^{n+1/2}_{i+1/2,j}$, 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 [1]. 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, $U^{n+1/2}_{i,j+1/2,B}$ and $U^{n+1/2}_{i,j+1/2,T}$, 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 $U^{n+1/2}_{i,j+1/2}$ 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

grid [1]. 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 $U^{n+1/2}_{i,j+1/2}$, and compute $T^{n+1/2}_{i+1/2,j}$, $T^{n+1/2}_{i,j+1/2}$, $(\rho Y)^{n+1/2}_{i+1/2,j}$, and $(\rho Y)^{n+1/2}_{i,j+1/2}$, again using the approach in [1]. 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 value 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 to ensure that the numerical scheme is freestream preserving with respect to temperature in the presence of multiple species. The advection scheme uses van Leer slope limiting [37] in the approximation of the first-order spatial derivatives. The advection scheme is hence monotonicity preserving but also necessarily nonlinear [22]. In particular, then, if the edge values of ρh were computed in the same manner as ρY, edge values of ρY 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 advection-diffusion-reaction step we solve difference equations obtained by applying the Crank-Nicholson method to the governing equations. The difference equations are solved using the multigrid strategy outlined above. 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_{fu}, Y_{ox}, Y_{pr}, 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 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}.

The difference equations for T, h, Y_l, and U are summarized below; 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 [1]. The discretization of $\nabla \cdot \tau$ uses similar strategies and will be discussed in detail in a future paper. 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.

Temperature. In the predictor, we compute $T^{n+1,p}$ by solving the difference equation

$$\begin{aligned}\rho^{n+1/2} c_p^n \left(\frac{T^{n+1,p} - T^n}{\Delta t} + (U \cdot \nabla T)^{n+1/2} \right) &= \frac{1}{2} \left(\nabla \cdot \nabla \lambda \left(T^n\right) \nabla T^n \right. \\ &\quad \left. + \nabla \cdot \nabla \lambda \left(T^n\right) \nabla T^{n+1,p} \right) \\ &\quad + \frac{\mu\left(T^n\right)}{\mathrm{Sc}} \sum_l \nabla h_l \left(T^n\right) \cdot \nabla Y_l^n .\end{aligned}$$

Note that ω is not included because of the operator split treatment of kinetics.

Enthalpy and Species. In the corrector, the discretization of the evolution equations for $\varphi = h, Y_l$ has the form

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

where $\Lambda^{\mathcal{N}} = \mu\left(T^{\mathcal{N}}\right)/\mathrm{Sc}$ when $\varphi = Y_l$ and $\mu\left(T^{\mathcal{N}}\right) / \left(\mathrm{Pr}\ c_p^{\mathcal{N}}\right)$ when $\varphi = h$ for $\mathcal{N} = n$ or $n+1, p$. The equations used in the predictor are found by substituting $\varphi^{n+1,p}$ and Λ^n for φ^{n+1} and $\Lambda^{n+1,p}$. As was the case for the temperature equation, ω is not included for Y_l.

Velocity. The discretization of the momentum equation is a coupled difference equation for $U^* = (u^*, v^*)$:

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

The viscosities in $(\nabla \cdot \tau)^n$ and $(\nabla \cdot \tau)^{n+1}$ are evaluated using T^n and $T^{n+1,p}$, respectively.

3.3 Projection Step

A projection [3] is now used to approximately enforce the divergence constraint (3.4) and determine $p^{n+1/2}$. In the advection-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

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

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.

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.* [1]. Many of details of the present algorithm are identical, or very nearly so, to those of the IAMR algorithm. The reader is referred to [1] 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).

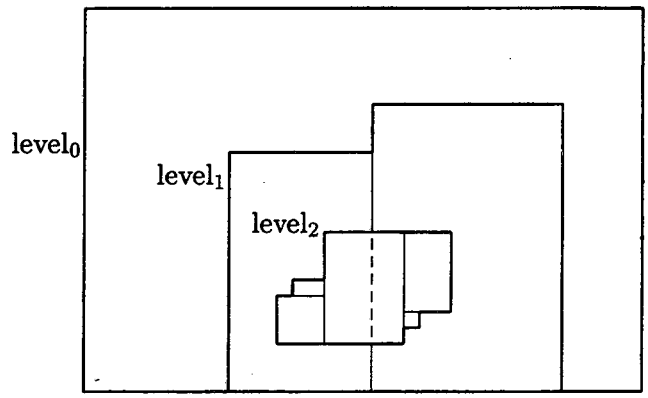

Figure 1: A properly nested hierarchy of grids

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 will involve 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.

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. The only

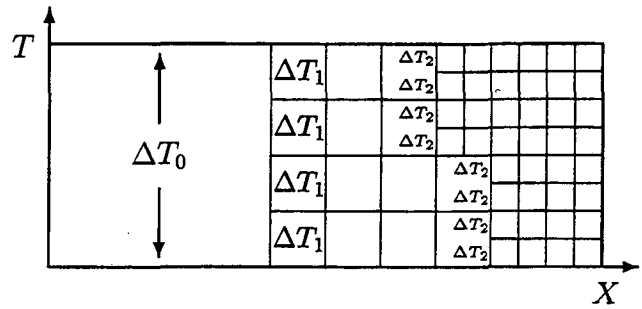

Figure 2: Multilevel timestep structure

significant modification in adapting the single grid algorithm to a level advance is that the level 0 value of Δt is always used in the pressure relaxation term in $\tilde{S}$ (3.4) regardless of the level index.

A detailed treatment of boundary conditions for the level advance is presented in [1]. For our purposes, we need only mention that boundary conditions for the advection 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 General State Variable Considerations

In the adaptive algorithm, the flow quantities whose values persist from one time step to the next are the dependent variables in the evolution equations, and, additionally, $\tilde{S}$ and $\partial\tilde{S}/\partial t$. The last two quantities are persistent for algorithmic simplicity and efficiency. The values of $\tilde{S}$ and $\partial\tilde{S}/\partial t$ 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 or that are ghost cells not contained within existing level ℓ grids.) The choice to consider $\tilde{S}$ and $\partial\tilde{S}/\partial t$ as state-like quantities was made in particular to minimize the complexity of the synchronization step.

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 (h) is defined by first conservatively interpolating ρ and ρY_l (ρh).

The conservative interpolation of the quantities ρ, ρY_l, and ρh is the final area requiring general discussion. As in the single level advection step, the conservative interpolation algorithm uses van Leer slope limiting [37] 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 [1]. 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, each such operation obtains its Dirichlet boundary data from next coarser level. 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 [1]):

(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 outline the correction strategies. The details will be presented in a future paper.

(M.1) is corrected by averaging the fine grid data onto the coarse grid data following the approach in [1]. 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 exactly the same approach as that used in [1]. 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 used in [1] for incompressible flow. Because the advection velocities used in the original coarse level advance did not contain this correction, we repeat the coarse level advection 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 advection step, we follow the same prescription for ρh that was used in §3.2.1.

The correction for (M.3) uses the same general approach as in [1]. 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 divergence of the flux corrections computed in the MAC sync advection step.

Unlike (S.1), (S.3) affects the solution at the 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

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

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

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

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 [1], with straightforward modifications for non-constant viscosity and the tensor form of τ. 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 [1]. 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 [1] 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.

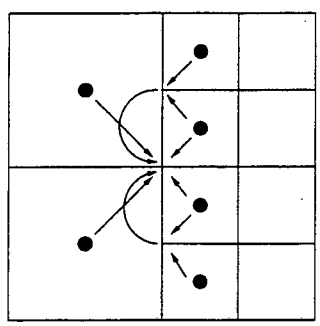

Figure 3: Schematic showing contributions of coarse and fine grid cell-centered values of $\partial\tilde{S}/\partial t$ to the node-based residual for a refinement ratio 2.

The remainder of (S.4) is identical to the case for incompressible flow reported in [1]. 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.

5 Computational Results

In this section we present two numerical examples illustrating the methodology described above. In both examples, $\Delta r = \Delta z$ and the CFL number $\sigma = .4$.

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.* [30]. 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 ≈ 760 for a reference length equal to the outer diameter of the air jet.

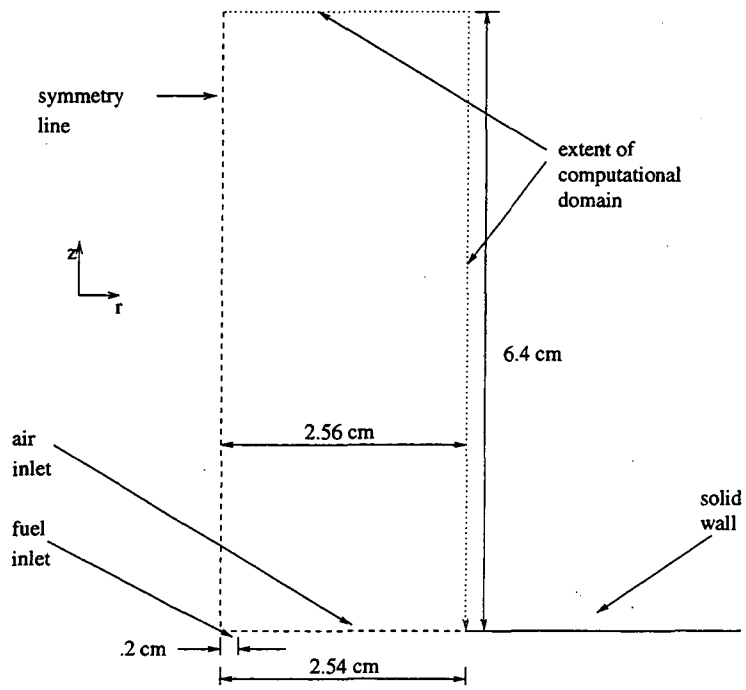

Figure 4: Sketch of specification of unconfined coflowing methane-air diffusion flame.

In our computation, the flame is ignited by a small hot patch ($T = 1200$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. The region $T > 2000$ K is refined to level 2.

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 also shown as thin lines in the plots.

Figure 6 shows the flame at steady-state. We compute a flame length and a maximum temperature of 1.68 cm and 2197 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 high

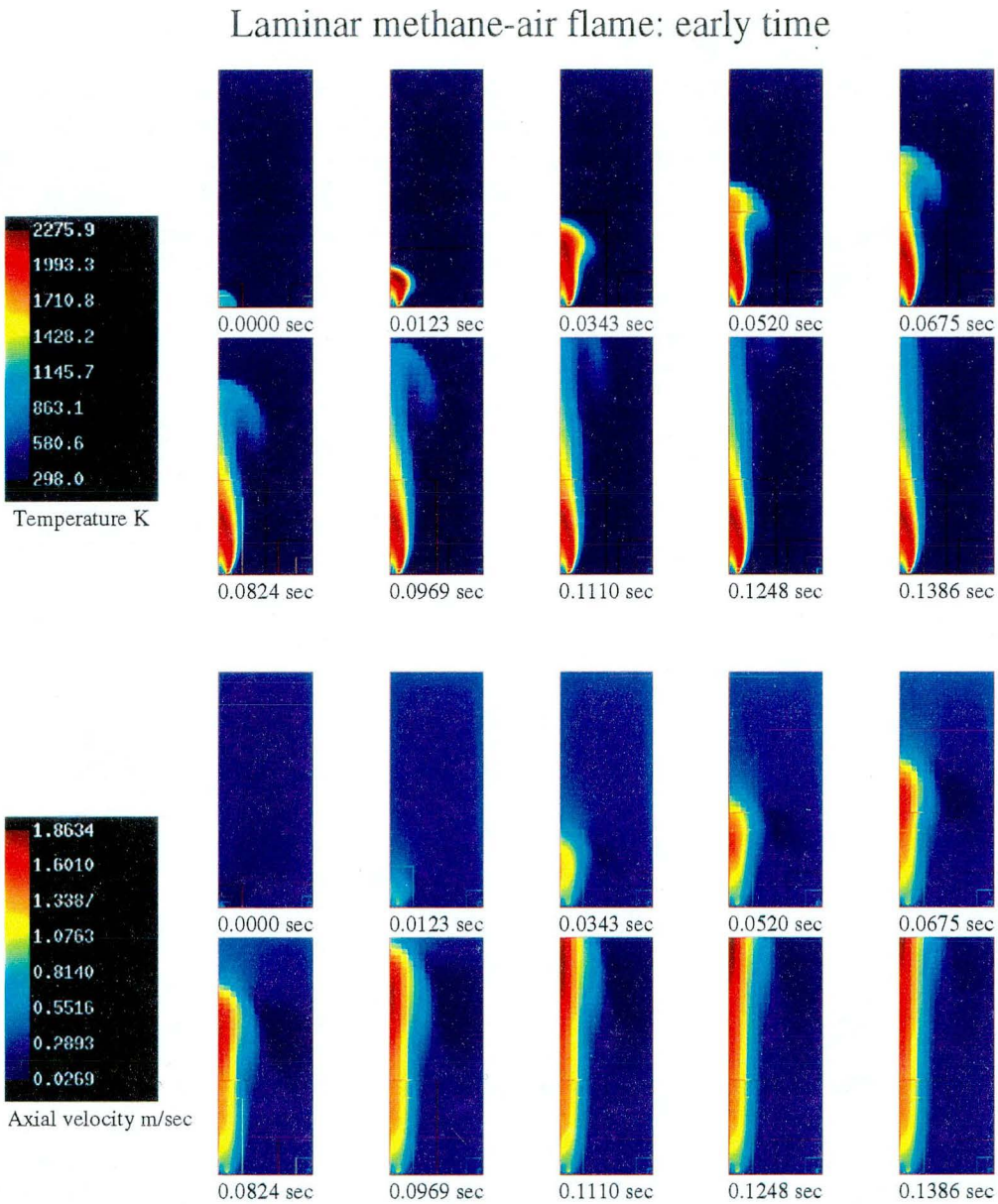

Figure 5: Unconfined coflowing methane-air laminar diffusion flame: early time

temperatures may be due to using a reduced kinetics mechanism and/or species-independent mass diffusivities.

Note that after the initial projection, the maximum axial velocity $v_{max} \approx 1$ m/sec; at steady state, $v_{max} \approx 1.8$ m/sec. The computed acceleration is consistent with the acceleration due to buoyancy. However, the use of a hot patch to ignite the flow, and the resultant large initial velocity due to the imposition of the divergence constraint, probably results in too rapid a development of the computed flow. A possible approach to computing a more realistic picture of the early time flow would be to ramp the inlet velocity in time

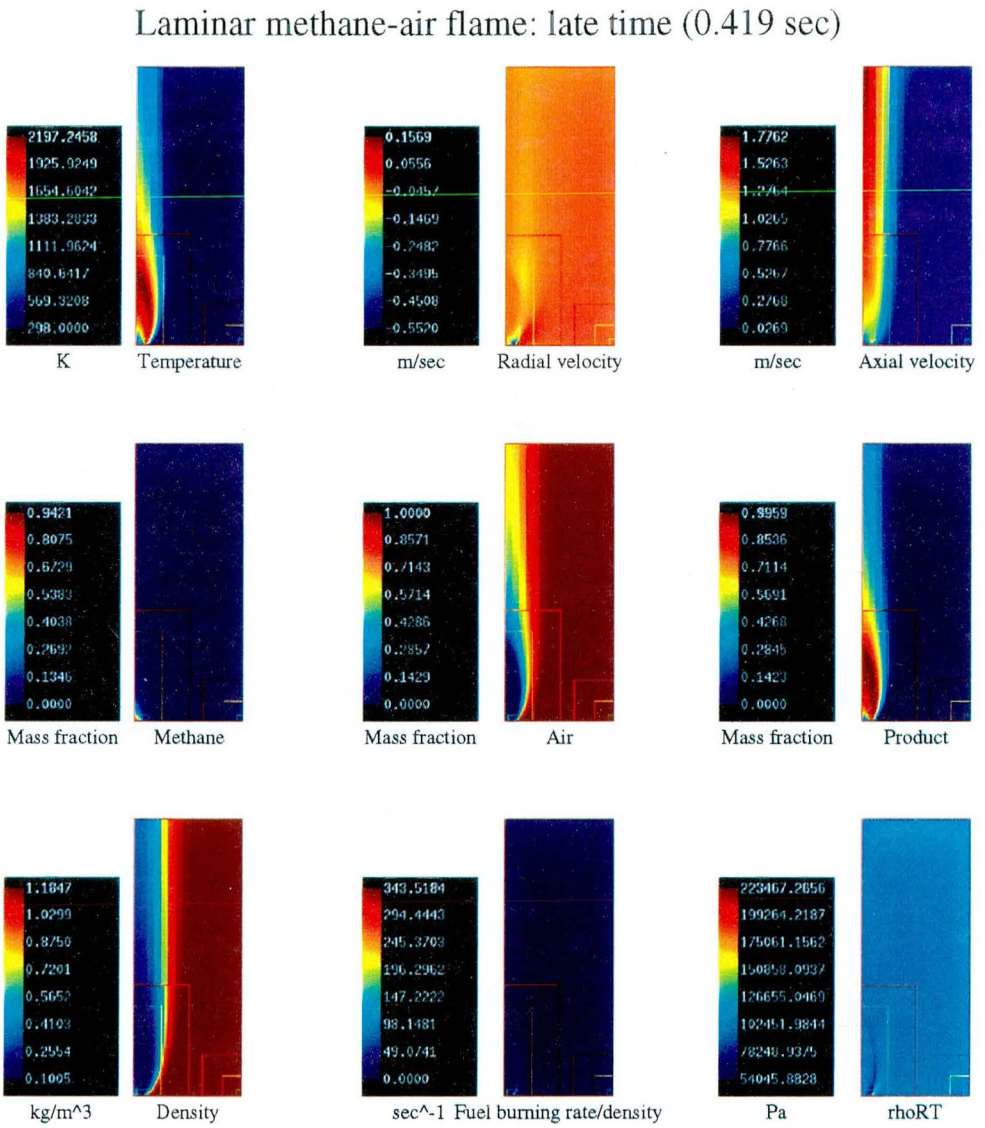

Figure 6: Unconfined coflowing methane-air laminar diffusion flame: late time (0.419 sec). ρRT is plotted to show how well the scheme meets the constraint $p_0 = \rho RT$. The two values differ significantly only along the edge of the flame.

and to model ignition as a transient source in the enthalpy equation.

5.1.1 Timings

We now present timings of the code for the steady laminar flame problem discussed above. All refinement ratios equal two. Four cases are reported: the 16×40 base grid with three levels of refinement discussed above, a 32×80 base grid with two levels, a 64×160 base grid with one level, and a uniform 128×320 grid. In the two additional refined cases, the inlets

and the region $T > 2000$K are refined to the finest level. The calculations were all run on a 300 MHz single processor DEC Alpha workstation to a final time of .10412 sec. Table 3 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 double 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.

Gridding	CPU Time		Cells Advanced	Peak Memory Usage
	Total(s)	μs/cell	Number	Mb
128×320, uniform	45810.	615	74547200	33
$64 \times 160, R = 2$	14380.	1085	13255808	16
$32 \times 80, R = 2, 2$	7260.	1171	6201536	10
$16 \times 40, R = 2, 2, 2$	8039.	1208	6654496	9

Table 3: 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 Flickering Methane-Air Diffusion Flame

The other example is the calculation of a flickering, unconfined coflowing methane-air diffusion flame. The computation models the coannular burner used by Smyth *et al.* [33, 32] in a flame study performed to help develop better models of soot formation. They report results that include the effect of acoustic forcing [33] and those that do not [32]. The latter case is the one computed here. Yam *et al.* [39] have also simulated this flow using a single grid projection method.

The experimental configuration is conceptually similar to that modeled in the previous section. 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.

In our computation, the flame is ignited by a small hot patch ($T = 1200$K) next to the inlet. We use a 32×128 level 0 grid to cover a 6.4 cm by 25.6 cm problem domain. There are two additional levels of refinement. The refinement ratio $R_\ell = 2$ for $\ell = 0, 1$, so that the equivalent uniform grid is 128×512. The inlet boundaries are refined to level 2 so that they align with level 2 grid lines. The region $T > 1950$ K is also refined to level 2.

During the early development of the flow, the flame grows in length and oscillates in a non-periodic manner; see Figure 7 for the time history of the flame length. After approximately .75 sec, the flame reaches a "steady-state" in which it exhibits a highly periodic oscillatory behavior best described as flickering. The flame oscillations are caused by a buoyancy induced Kelvin-Helmholtz type of instability. Figure 8 displays the temperature

field during a single flame oscillation. We compute a flickering frequency of 12.2 Hz; Smyth *et al.* report a value of 12 Hz. The computed time-averaged flame height is 6.94 cm; the experimental value is 7.9 cm. Yam *et al.* compute values 15.7 Hz and 5.51 cm. As in the calculation reported in the previous section, our computed temperatures are again too high; see the discussion above. We also compute a larger flame height oscillation (roughly 3 cm) at steady-state than do Yam *et al.* (1 cm).

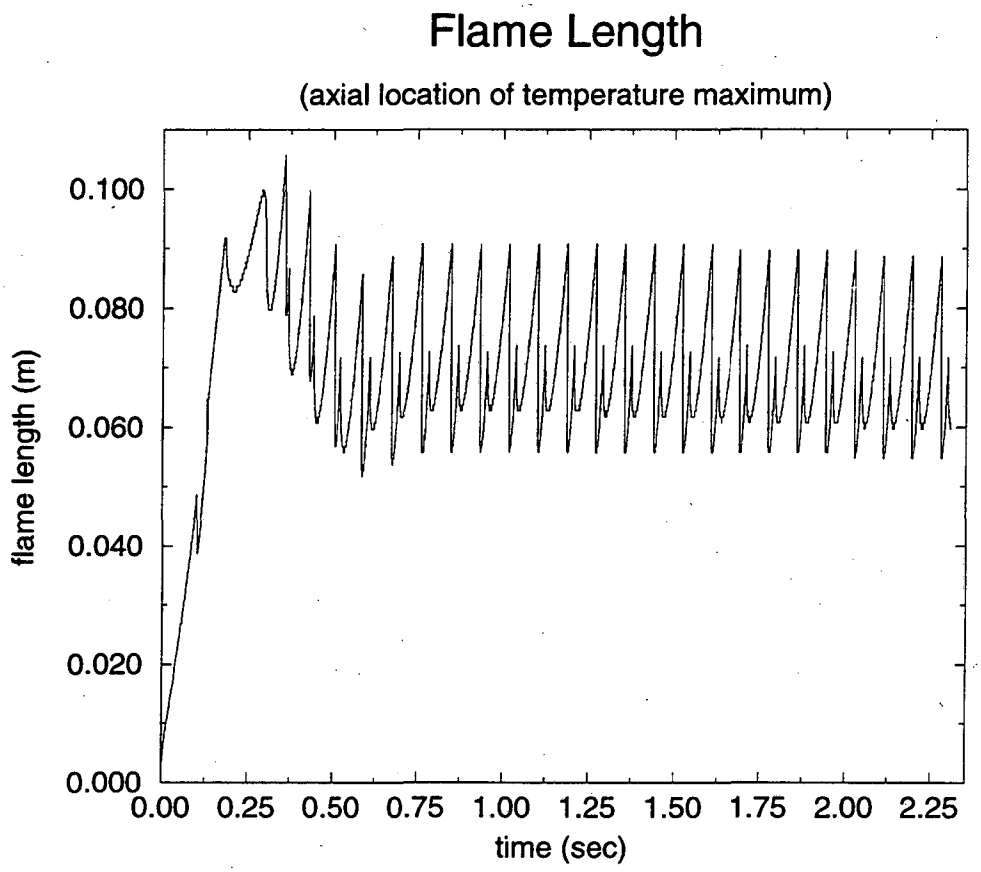

Figure 7: Axial position of the maximum temperature of the flickering flame along the centerline axis as a function of time.

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 two test problems are favorable with the exception that the computed temperatures are significantly higher than the values reported elsewhere. We speculate that the high temperatures

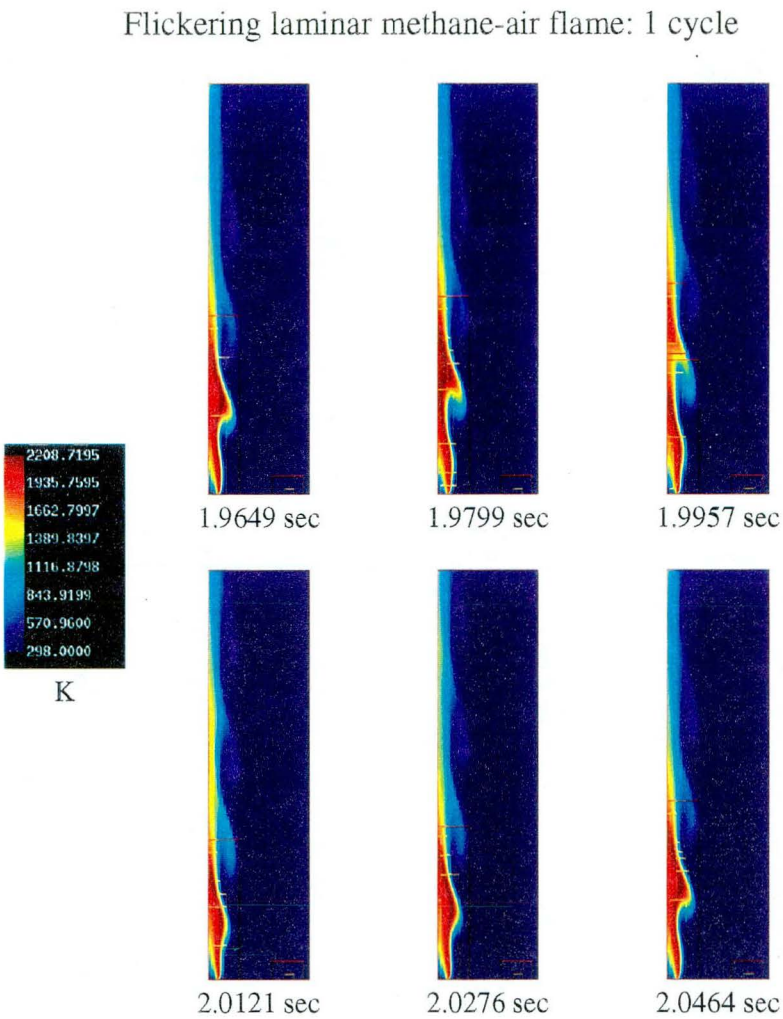

Figure 8: Temperature field of flickering flame during a single flame oscillation.

may be due to the use of a reduced kinetics mechanism and/or species-independent mass diffusivities.

Future directions for this work include incorporating detailed chemistry and species dependent mass diffusivities, and extending the methodology to three-dimensional and turbulent flows and to realistic engineering geometries.

Acknowledgments. The code which implements the algorithm described in this paper is derived from the IAMR code [1] and shares a large common code base with it, including BOXLIB. The authors would like to thank the following people: Ann Almgren, Marc Day, Chuck Rendleman, and Mike Welcome for their contributions to the development of IAMR code; Ann Almgren for numerous enhancements to IAMR which facilitated this research; and Mike Lijewski for his work on BOXLIB.

References

[1] 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. Submitted to *J. Comput. Phys.* LBNL-39075, Lawrence Berkeley National Laboratory, July, 1996.

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

[3] A.S. Almgren, J.B. Bell, and W.G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *J. Comput. Phys.*, 17(2), 1996.

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

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

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

[7] B.A.V. Bennett. *Solution-Adaptive Gridding Methods with Application to Combustion Problems.* PhD thesis, Yale University, 1997.

[8] B.A.V. Bennett and M.D. Smooke. 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.

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

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

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

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

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

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

[15] S. Glasstone. *Thermodynamics for Chemists.* van Nostrand Co., Princeton, 1947.

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

[17] A.M. Kanuary. *Combustion Phenomena.* Gordon & Breach, New York, 1982.

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

[19] K.K. Kuo. *Principles of Combustion.* Wiley-Interscience, New York, 1986.

[20] M. Lai. *A Projection Method for Reacting Flow in the Zero Mach Number Limit.* PhD thesis, University of California, Berkeley, 1993. Also published as CPAM-601, University of California, Berkeley, February, 1994.

[21] M. Lai, J.B. Bell, and P. Colella. 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.

[22] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhauser, Boston, 1990.

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

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

[25] H.N. Najm. A conservative low Mach number projection method for reacting flow modeling. In *Transport Phenomena in Combustion*, pages 921–932. Taylor & Francis, 1996.

[26] H.N. Najm. A mass-conservative projection method for confined low Mach number reacting flow. *AIAA Journal*, Submitted 1996.

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

[28] 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 a gas-fired furnace. Fall Meeting, Western States Section of the Combustion Institute, 1995, WSS/CI 95F-165.

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

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

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

[32] K.C. Smyth, 1997. Private communication.

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

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

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

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

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

[38] P. Wesseling. *An Introduction to Multigrid Methods.* John Wiley & Sons, 1992.

[39] C.G. Yam, K.D. Marx, J.-Y. Chen, and C.-P. Chen. 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.

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