Numerical Simulation of Laminar Reacting Flows with Complex Chemistry

M S Day and J B Bell

Lawrence Berkeley National Laboratory,
Berkeley, CA, 94720

Abstract. We present an adaptive algorithm for low Mach number reacting flows with complex chemistry. Our approach uses a form of the low Mach number equations that discretely conserves both mass and energy. The discretization methodology is based on a robust projection formulation that accommodates large density contrasts. The algorithm uses an operator-split treatment of stiff reaction terms and includes effects of differential diffusion. The basic computational approach is embedded in an adaptive projection framework that uses structured hierarchical grids with subcycling in time that preserves the discrete conservation properties of the underlying single-grid algorithm. We present numerical examples illustrating the performance of the method on both premixed and non-premixed flames.

PACS numbers: 47.40.Fw, 82.40.Py

Submitted to: *Combust. Theory Modelling*

1. Introduction

Detailed modelling of time-dependent reacting flows with realistic chemical mechanisms places severe demands on computational resources. One approach to reducing this computational cost is the use local adaptive mesh refinement to reduce the total number of computational zones that must be advanced for a specific problem. Local refinement for steady combustion has been discussed by a number of authors. See for example, Smooke *et al* [36], Coelho and Pereira [7], de Lange and de Goey [10], Mallens *et al* [22] Somers and de Goey [34] and Bennett and Smooke [4] and the references cited in these works.

For time dependent flows, Najm *et al* [26] couple a local refinement algorithm for species and temperature with a vortex method for momentum. Pember *et al* [28] present an adaptive projection algorithm for time-dependent low-Mach number combustion using simplified kinetics and an assumption of Lewis number of one. The methodology in Pember *et al* [28] uses a hierarchical structured refinement approach based on the local adaptive projection methodology developed by Almgren *et al* [3]. The method presented here represents a generalization of the Pember *et al* methodology to incorporate complex chemistry and the effects of differential diffusion.

A key consideration in developing a structured refinement algorithm is the choice of single-grid algorithm. For incompressible flows, projection-based fractional step methods have proven to provide an efficient approach that is well suited to adaptive refinement. For low Mach number reacting flows, two different projection-based sequential algorithms have been proposed. One of these approaches has been used by Knio *et al* [18] and Najm *et al* [25] to model flows with complex chemistry. Their approach is based on a computational framework for low-Mach number reacting flows developed by McMurtry *et al* [23] and Rutland and Ferziger [32].

The single-grid algorithm that forms the basis for our adaptive algorithm uses an alternative numerical approach that is similar to the method used by Pember *et al* [28]. The basis for the approach was first introduced by Lai [19] and Lai *et al* [20]. Related implementations or extensions include Hilditch and Colella [13] and Pember *et al* [29, 27]. The key elements in the extension of the algorithm presented in Pember *et al* [28] to the current setting are the use of a symmetric operator-split approach that is second-order accurate in time and the treatment of differential diffusion.

Although our approach is similar in spirit to the approach of Knio *et al* and Najm *et al.*, there are two major differences. First, in the approach of Knio *et al* and Najm *et al.*, an evolution equation for density is derived by differentiating the equation of state in time and using the temperature and species evolution equations to replace the temporal derivative of those quantities by spatial operators. Temperature is then computed directly from the equation of state. In our approach, we solve for species mass densities and enthalpy

as primary unknowns. We construct a conservative discretization for mass and energy but allow the approximate solution to drift off the constraint imposed by the equation of state at fixed ambient pressure. In section 2, we discuss how this drift is controlled.

The second major difference between our approach and that of Knio and Najm relates to the projection step of the algorithm. The first step of the algorithm computes an intermediate velocity field that does not satisfy the constraint. The projection step corrects the velocity so that the low Mach number divergence constraint is satisfied. Knio and Najm formulate a projection algorithm in terms of a constant-coefficient pressure-Poisson equation to be solved at each time step. Here, we define the projection as an orthogonal decomposition of velocity in a density-weighted space. While our approach leads to a somewhat more expensive variable-coefficient elliptic solve, it has the advantage that it easily accommodates large density contrasts (800:1). (see Almgren *et al* [1] and Sussman *et al* [35]).

The remainder of the paper is organized as follows. In section 2, we introduce the mathematical model and describe the computational framework for the projection methodology. We will not discuss in detail the Godunov-type upwind advection algorithm for computing advective derivatives (this aspect of algorithm is identical to those detailed in Almgren *et al.*[1] and Pember *et al.*[28]). In section 3, we present the second-order time discretization of the species and enthalpy equations, including the operator-split treatment of detailed chemistry and the discretization of the terms due to differential diffusion. In section 4, we give an overview of our adaptive methodology, focusing primarily on modifications to the methodology presented in [1, 28] required for the present application. In section 5, we present results of applying this numerical solution algorithm to two problems: the evolution of a strong vortex interacting in two-dimensions with a planar premixed H_2-O_2 flame, and the response of a co-flow axisymmetric laminar CH_4-O_2 flame to a periodically modulated fuel stream. The final section presents some conclusions and discusses future generalizations of the methodology.

2. Mathematical model and computational framework

The model presented here is based on the model for low Mach number combustion introduced by Rehm and Baum [30] and rigorously derived from low Mach number asymptotic analysis by Majda and Sethian [21]. We consider a gaseous mixture ignoring Soret and Dufort effects, body forces and radiative heat transfer, and assume a mixture model for species diffusion[37, 17]. For low-speed flow in an unconfined domain, we can write

$$p(x,t) = p_0 + \pi(x,t)$$

where p_0 is the ambient thermodynamic pressure. The low Mach number model assumes that the perturbational pressure field, $\pi/p_0 \sim \mathcal{O}(M^2)$, where M is the Mach number, and that all thermodynamic quantities are independent of π. In the low Mach number limit,

the equations describing momentum transport and conservation of species and enthalpy are given by

$$\rho \frac{DU}{Dt} = -\nabla \pi + \nabla \cdot \tau, \tag{1}$$

$$\frac{\partial \rho Y_m}{\partial t} + \nabla \cdot U \rho Y_m = \nabla \cdot \rho \mathcal{D}_m \nabla Y_m - \dot{\omega}_m, \tag{2}$$

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot U \rho h = \nabla \cdot \lambda \nabla T + \sum_m \nabla \cdot \rho h_m(T) \mathcal{D}_m \nabla Y_m \tag{3}$$

where ρ is the density, U is the velocity, Y_m is the mass fraction of species m, h is the enthalpy of the gas mixture, and T is the temperature. Here, $\dot{\omega}_m$, the destruction rate for ρY_m due to chemical reactions, is specified via a collection of fundamental reactions using a CHEMKIN-III[16] compatible database. The stress tensor is given by

$$\tau = \mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \nabla \cdot U \right)$$

where $\mu(Y_m, T)$ is the viscosity, $\mathcal{D}_m$ are the species mixture-averaged diffusion coefficients[14], λ is the thermal conductivity and $h_m(T)$ is the enthalpy of species m. These evolution equations are supplemented by an equation of state:

$$p_0 = \rho R_{mix} T = \rho \mathcal{R} T \sum_m \frac{Y_m}{W_m} \tag{4}$$

where W_m is the molecular weight of species m, and by a relationship between enthalpy, species and temperature:

$$h = \sum_m Y_m h_m(T) \qquad . \tag{5}$$

Neither species diffusion nor reactions redistribute total mass; hence, we have $\sum_m \mathcal{D}_m \nabla Y_m = 0$ and $\sum_m \dot{\omega}_m = 0$. Summing the species equations and noting that $\sum_m Y_m = 1$, we see that (2) implies the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho U = 0 \tag{6}$$

The evolution specified by (1-3) is subject to the constraint on velocity that

$$\begin{aligned} \nabla \cdot U = & \frac{1}{\rho c_p T} \left(\nabla \cdot \lambda \nabla T + \sum_m \rho \mathcal{D}_m \nabla Y_m \cdot \nabla h_m \right) \\ & + \frac{1}{\rho} \sum_m \frac{W}{W_m} \nabla \cdot \rho \mathcal{D}_m \nabla Y_m \; + \frac{1}{\rho} \sum_m \left(\frac{h_m(T)}{c_{p,mix} T} - \frac{W}{W_m} \right) \dot{\omega}_m \equiv S \end{aligned} \tag{7}$$

where $W = (\sum_m Y_m / W_m)^{-1}$ and $c_{p,mix} = \sum_m Y_m dh_m/dT$. The constraint (7) is obtained by differentiating the equation of state along particle paths and replacing the Lagrangian derivatives by expressions obtained from (2, 3, 5).

As discussed in the introduction, we have written the low Mach number equations in a form that directly expresses conservation of both species and enthalpy. In a fractional step approach it is impossible to numerically conserve species and enthalpy while satisfying the equation of state (4). In our approach we discretely conserve both species and energy so that the evolution of these quantities does not satisfy the equation of state (with T determined from (5)). In order to prevent the algorithm from drifting too far off the equation of state we add a correction term to the constraint (7); namely, we use the modified constraint

$$\nabla \cdot U = S + f \; \frac{c_{p,mix} - R}{\Delta t \; c_{p,mix} \hat{p}} \; (\hat{p} - p_0) \equiv \hat{S} \tag{8}$$

where $\hat{p}$ is the thermodynamic pressure, defined by (4) as a function of the species densities and enthalpy and f is a numerical damping factor, $f < 1$. This additional term serves to damp the system back onto the ambient equation of state if the solution drifts off that constraint. (See Pember *et al* [28] for a discussion of this term and a heuristic for its derivation.)

Computational framework

In this section we give an overview of the projection formulation for solving (1–3) subject to the modified constraint (8). The reader is referred to Pember *et al* [28], and Almgren *et al* [1] for details of the algorithm. For the methodology described in this paper, the velocity, species densities and enthalpy are all defined at cell centers, denoted by a subscript–ij. The perturbational pressure, π, is a nodal quantity with values denoted by subscripts–$i + 1/2, j + 1/2$ that is defined at $1/2$ time levels. π is used to compute a cell-centered gradient, $\nabla\pi$ at $t^{n+1/2}$ in the momentum equation (1).

The overall projection formulation is a multistep process. In the first step, we use an unsplit second-order Godunov procedure to predict a time-centered advection velocity, $U^{ADV,*}$, using the cell-centered data at t^n and the lagged pressure gradient from the interval centered at $t^{n-1/2}$. The provisional field, $U^{ADV,*}$, represents a normal velocity on cell edges analogous to a MAC-type staggered grid discretization of the Navier-Stokes equations (see [12], for example). However, $U^{ADV,*}$ fails to satisfy the time-centered divergence constraint. We apply a discrete projection by solving the elliptic equation

$$D^{MAC} \frac{1}{\rho^n} G^{MAC} \phi^{MAC} = D^{MAC} U^{ADV,*} - \left(\hat{S}^n + \frac{\Delta t^{\,n}}{2} \; \frac{\hat{S}^n - \hat{S}^{n-1}}{\Delta t^{\,n-1}} \right) \tag{9}$$

for ϕ^{MAC}, where D^{MAC} represents a centered approximation to a cell-based divergence from edge-based velocities, and G^{MAC} represents a centered approximation to edge-based gradients from cell-centered data. The solution, ϕ^{MAC}, is then used to define

$$U^{ADV} = U^{ADV,*} - \frac{1}{\rho^n} G^{MAC} \phi.$$

U^{ADV} is a second-order accurate, staggered-grid vector field at $t^{n+1/2}$ that discretely satisfies the constraint (8), and is used for computing the time-explicit advective derivatives for U, ρh and ρY_m.

In the next step of the algorithm we advance the advection-reaction-diffusion system for $U, \rho h$, and ρY_m. First we advance ρh and ρY_m to t^{n+1}. This procedure is itself a complex multistep process that is discussed in detail in the next section. Given the new-time values for ρh, ρY_m and $\rho = \sum_m \rho Y_m$ we compute the temperature and fluid viscosity at t^{n+1} and evaluate $\hat{S}^{n+1}$ using finite difference approximations.

We then compute an intermediate velocity field, $U^{n+1,*}$ using the lagged pressure gradient, by solving

$$\rho^{n+1/2}\frac{U^{n+1,*} - U^n}{\Delta t} + [(U^{ADV} \cdot \nabla)U]^{n+1/2} = \frac{1}{2}(\nabla \cdot \tau^n + \nabla \cdot \tau^{n+1,*}) - \nabla \pi^{n-1/2}$$

where $\tau^{n+1,*} = \mu^{n+1}((\nabla + \nabla^T)U^{n+1,*} - \frac{2}{3}\delta_{ij}\hat{S}^{n+1})$ and $\rho^{n+1/2} = \frac{1}{2}(\rho^n + \rho^{n+1})$. At this point, the intermediate velocity field $U^{n+1,*}$ does not satisfy the constraint. We apply an approximate projection to update the pressure and to project $U^{n+1,*}$ onto the constraint surface. In particular, we solve

$$L^\rho \phi = D(U^{n+1,*} + \Delta t\, G\pi^{n-1/2}) - \hat{S}^{n+1} \tag{10}$$

for nodal values of ϕ, where L^ρ is a finite element approximation to $\nabla \cdot \frac{1}{\rho}\nabla$ with ρ evaluated at $t^{n+1/2}$. In this step, D is a discrete second-order operator that approximates the divergence at nodes from cell-centered data, and $G = -D^T$ approximates a cell-centered gradient from nodal data.

In the projection formulation ϕ satisfies Neumann boundary conditions at solid walls and inflow boundaries. At outflow boundaries, Dirichlet conditions are generated to suppress any tangential accelerations on the fluid leaving the domain. We compute nodal values for $\hat{S}^{n+1}$ for the solution of (10) using a volume-weighted average of cell-centered values. Finally, we determine the new-time cell-centered velocity field from

$$U^{n+1} = U^{n+1,*} - \frac{1}{\rho}G\phi$$

and the new time-centered pressure from

$$\pi^{n+1/2} = \phi.$$

We note that this represents an improved version of the approximate projection algorithm over the approach used by Pember *et al* [28] (see Almgren *et al* [2]). This completes the description of the basic projection scheme.

3. Numerical integration of species and enthalpy equations

In this section we describe the numerical integration procedure for the species and enthalpy equations. We base the approach on the assumption that the reactions are "stiff" relative to the time scales associated with the fluid dynamics, and that accurately tracking the chemical kinetics will require subcycling in time with respect to a time step based on CFL considerations. In the algorithm presented here we use an alternate form of enthalpy equation that emphasizes deviation from unity Lewis number,

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot U\rho h = \nabla \cdot \frac{\lambda}{c_{p,mix}} \nabla h + \sum_m \nabla \cdot h_m (\rho \mathcal{D}_m - \frac{\lambda}{c_{p,mix}}) \nabla Y_m \tag{11}$$

As an aside, the case of unity Lewis number corresponds to $\rho \mathcal{D}_m = \frac{\lambda}{c_{p,mix}}$ for all m so that the second term on the right hand side of (11) vanishes. By substituting the definition of enthalpy (5) into the enthalpy equation, (3), and evaluating species derivatives using the species evolution equations, (2), we obtain the following evolution equation for temperature

$$\rho c_{p,mix} \frac{DT}{Dt} = \nabla \cdot \lambda \nabla T + \sum_m \rho \mathcal{D}_m \nabla h_m(T) \cdot \nabla Y_m + \sum_m \dot{\omega}_m h_m(T) \tag{12}$$

We use this equation for predicting a provisional temperature to evaluate fluid properties at the new time level prior to completing the predictor step for species and enthalpy. This equation is also used to predict the temperature at the cell edges in the Godunov step discussed below. In all other cases, the temperature is computed by inverting (5).

Operator-split formulation

We advance (2, 11) using a splitting scheme that computationally decouples the procedure into two independent operators, representing the chemistry and CFD components. We define the pointwise chemistry operator, $\mathcal{H}^C_{dt}$, such that $\mathcal{H}^C_{dt}(Y^n_m, T^n) \rightarrow (Y^{n+1}_m, T^{n+1})$ represents a discrete chemistry evolution of the reaction kinetics from t^n to t^{n+1}, using

$$\frac{\partial \rho Y_m}{\partial t} = -\dot{\omega}_m$$

$$\rho c_{p,mix} \frac{\partial T}{\partial t} = \sum_m \dot{\omega}_m h_m(T)$$

under adiabatic, fixed-volume conditions. We note that although the kinetics integration modifies the temperature field, there are no net changes in enthalpy. The changes in temperature that result from the chemistry step are consistent with the change in mixture composition associated with the reactions and a constant enthalpy within the cell.

In our implementation, we integrate the chemistry component, $\mathcal{H}^C$, using time-implicit backward difference methods, as implemented in VODE[5], a general-purpose stiff ODE integration software package. VODE utilizes adaptivity in order of accuracy and subcycled

time-step size so that an absolute error tolerance of 10^{-16} in mass fractions is maintained throughout. Typically, the resulting scheme is between third and fifth order convergent in Δt .

After ρY_m and T have been advanced from t^n to $t^{n+1/2}$ by $\mathcal{H}^C$, we integrate the system (2,5,11) using a sequential predictor-corrector Crank-Nicolson scheme over an interval Δt, setting $\dot{\omega}_m = 0$. We represent this component of the solution algorithm by the operator, $\mathcal{H}^{A-D}_{dt}$. We complete the time step for the species and energy equations by performing a second chemistry evolution, $\mathcal{H}^C_{dt/2}$. This combined process defines a symmetric Strang-type splitting,

$$\mathcal{H}_{dt}(Y^n_m, h^n) = \mathcal{H}^C_{dt/2}\mathcal{H}^{A-D}_{dt}\mathcal{H}^C_{dt/2} \rightarrow (Y^{n+1}_m, h^{n+1})$$

which constitutes formally a second-order temporal discretization of the complete system. Note that because the second step of this three-step process communicates with the boundary, we must apply $\mathcal{H}^C_{dt/2}$ as well to data at the boundary of the computational region during the first chemistry substep. Once the evolution of the species and energy equations are complete, we can evaluate the modified constraint (8) using the state and transport/thermodynamic properties at $(\rho, Y_m, T, h)^{n+1}$ and an effective $\dot{\omega}_m$ averaged over the time step.

Advection–Diffusion step

There are several possible approaches to solving the species diffusion and enthalpy equations within the Godunov advection-diffusion framework. One approach is to compute advective derivatives and then compute the diffusive transport for all of the equations simultaneously using a non-linear iterative scheme. There are a variety of other approaches that iterate the system with varying degrees of coupling. The scheme used here is based on a sequential approach in which the equations are solved individually in a specified order, updating properties as appropriate. This scheme, which follows that of Pember *et al.*, is, essentially, a non-linear Gauss-Seidel iteration scheme for the complete system. The basic approach is to solve the equations with lagged fluid properties to determine predicted values of the solution at the new time level. These predicted values can then be used to recompute values of the fluid properties required for evaluating the solution at the new time level to second-order accuracy.

For the advection-diffusion step we begin with data obtained by advancing the chemistry with $\mathcal{H}^C_{dt/2}$. We refer to this data with a superscript n in the remainder of the section. As with the velocity equation discussed before, we first compute edge-centered states for ρY_m and T at $t + \Delta t/2$ using a second-order Godunov procedure. Time-centered edge values of ρY_m and T are used to compute $t^{n+1/2}$ values for ρ and ρh using $\sum_m \rho Y_m = \rho$ and (5).

Because neither reactions or species diffusions transport mass, we can now update ρ using the discrete form of the continuity equation

$$\rho^{n+1} = \rho^n - \Delta t \ \nabla \cdot \left(\sum_m U^{ADV} \rho Y_m^{n+1/2} \right).$$

Given ρ^{n+1} and fluid properties $(\lambda, c_{p,mix}, h_m)^n$ computed after the chemistry step, we predict a preliminary new-time temperature, $\tilde{T}^{n+1,*}$ step using a Crank-Nicolson discretization of the temperature equation

$$\rho^{n+1/2} c_{p,mix}^n \left(\frac{\tilde{T}^{n+1,*} - T^n}{\Delta t} + \left(U^{ADV} \cdot \nabla T \right)^{n+1/2} \right) = \frac{1}{2} \left(\nabla \lambda^n \nabla T^n + \nabla \lambda^n \nabla \tilde{T}^{n+1,*} \right) + \sum_m \rho \mathcal{D}_m^n \nabla h_m(T^n) \cdot \nabla Y_m^n \tag{13}$$

where $\rho^{n+1/2} = 1/2(\rho^{n+1} + \rho^n)$. Next, we approximate the mixture-averaged diffusivities at t^{n+1} using Y_m^n and the provisional predicted temperature, $\tilde{T}^{n+1,*}$. We denote these diffusivities as $\mathcal{D}_m^{n+1,*} = \mathcal{D}_m^{n+1,*}(Y_m^n, \tilde{T}^{n+1,*})$, and use them to predict a preliminary $\tilde{Y}_m^{n+1,*}$ from

$$\frac{\rho^{n+1} \tilde{Y}_m^{n+1,*} - \rho Y_m}{\Delta t} + \left(\nabla \cdot U^{ADV} \rho Y_m \right)^{n+1/2} = \frac{1}{2} \nabla \cdot \left(\rho^{n+1} \mathcal{D}_m^{n+1,*} \nabla \tilde{Y}_m^{n+1,*} - \Gamma_m^n \right) \tag{14}$$

Here, Γ_m^n represents the species diffusion fluxes at t^n. Ideally, these diffusion fluxes would be defined by $\Gamma_m^n = -\rho^n \mathcal{D}_m^n \nabla Y_m^n$. However, the mixture-averaged formulation for $\mathcal{D}_m$ does not preserve mass balance, i.e. numerically, $\sum_m \Gamma_m^n \neq 0$. We want to modify these diffusive fluxes, Γ_m^n, to sum discretely to zero. Here, we consider two approaches for making this modification. For both algorithms we want to avoid the selection of a single "dominant" species for the entire problem whose diffusive flux is defined by the requirement that the diffusive fluxes sum to zero.

The first approach we consider is to identify the locally dominant species and adjust its diffusive flux. When $\sum_m \tilde{\Gamma}_m$ on an edge is non-zero, the diffusion fluxes on that edge are transporting net mass across that edge. Based on the sign of $\sum_m \tilde{\Gamma}_m$, we identify, for each face, the corresponding cell-center that is receiving excess mass. For concreteness, we consider the case in which $\sum_m \tilde{\Gamma}_m > 0$ at edge-$i + 1/2, j, k$. In this case, the diffusion fluxes are transporting net mass from cell-i, j, k to cell-$i + 1, j, k$. We then modify Γ_m for the species, m, in cell-$i + 1, j, k$ with the largest mass fraction in that cell so that the diffusive fluxes sum to zero. For cases in which there is a dominant species this approach reduces to the traditional treatment such as that found in the PREMIX code [15]. This is a robust and effective approach; however, it is not free-stream preserving for the chemical species. In particular, the differential equations have the property that a chemical species with an initially constant mass fraction that does not participate in any chemical reactions should remain constant. The approach outlined above, as well as traditional treatments, do not satisfy this property; they will introduce an erroneous variation in species concentrations.

We propose a second approach to modifying diffusive fluxes, Γ_m^n, so that they are free-stream preserving. In this approach, we modify Γ_m^n using a re-distribution idea originally introduced by Chern and Colella[6] to correct conservation errors in a volume-of-fluid shock tracking algorithm. In particular, given the preliminary species diffusion fluxes, $\tilde{\Gamma}_m = -\rho \mathcal{D}_m \nabla Y_m$, we identify "excess" flux and redistribute the excess in a mass-weighted manner so that the adjusted species diffusion fluxes, Γ_m, satisfy $\sum_m \Gamma_m \equiv 0$ on each cell edge.

As before, when $\sum_m \tilde{\Gamma}_m$ on an edge is non-zero, the diffusion fluxes on that edge are transporting net mass across that edge. Based on the sign of $\sum_m \tilde{\Gamma}_m$, we identify for each face the corresponding cell-center that is receiving excess mass. To modify the fluxes so that they sum to zero, we want to reduce the fluxes that contribute to that net mass flux while leaving the fluxes of the opposite sign alone.

We again consider the case in which $\sum_m \tilde{\Gamma}_m > 0$ at edge-$i+\frac{1}{2}, j, k$ so that the diffusion fluxes are transporting net mass from cell-i, j, k to cell-$i+1, j, k$. We group the fluxes based on whether their contribution to the update is positive or negative. We denote the respective groups by $\tilde{\Gamma}_k^+$ and $\tilde{\Gamma}_l^-$. Then we have

$$\sum_k \tilde{\Gamma}_k^+ + \sum_n \tilde{\Gamma}_l^- > 0 \quad .$$

We want to determine reduction factors α_k for the positive fluxes so that

$$\sum_k \alpha_k \tilde{\Gamma}_k^+ + \sum_n \tilde{\Gamma}_l^- = 0 \quad . \tag{15}$$

with $0 < \alpha_k < 1$. However, reducing the flux $\tilde{\Gamma}_k^+$ will also reduce the amount of species k in cell-$i+1, j, k$. We want balance the flux reduction amongst the different species , with a greater reduction for species that have a larger mass density. Thus, we want the flux reduction for species k to be proportional to the mass fraction of that species that is present; i.e.,

$$(1 - \alpha_k)\, \tilde{\Gamma} = \beta \tilde{Y_k} \tag{16}$$

for some β where Y_k corresponds to the mass fraction in the cell-$i+1, j, k$. The equations (15) and (16) define a linear system for the α_k and β. It is easy to show that for resulting α_k are between zero and one. Thus, if we define $\Gamma_k^+ = \alpha_k \tilde{\Gamma}_k^+$ and $\Gamma_l^- = \tilde{\Gamma}_l^-$ then $\sum_m \Gamma_m = 0$.

After the solution of the species diffusion equation (14) the initial new time fluxes, $\tilde{\Gamma}_n^{n+1,*} \equiv -\rho^{n+1} \mathcal{D}_m^{n+1,*} \nabla \tilde{Y}_m^{n+1,*}$ also do not sum to zero. We again apply the redistribution procedure to modify these fluxes to obtain $\Gamma_m^{n+1,*}$ that sum to zero. These corrected fluxes are then used to compute the predicted species mass fractions $Y^{n+1,*}$ using

$$\frac{\rho^{n+1} Y_m^{n+1,*} - \rho^n Y_m^n}{\Delta t} + \left(\nabla \cdot U^{ADV} \rho Y_m\right)^{n+\frac{1}{2}} = -\frac{1}{2} \nabla \cdot \left(\Gamma_m^{n+1,*} + \Gamma_m^n\right)$$

We now evaluate the fluid properties $(\lambda, c_{p,mix}, h_m)^{n+1,*}$ with the predicted species mass fractions $Y_m^{n+1,*}$ and preliminary predicted temperature $\tilde{T}^{n+1,*}$, and use them to compute the predicted new-time enthalpy $h^{n+1,*}$ using

$$\frac{\rho^{n+1}h^{n+1,*} - \rho^n h^n}{\Delta t} = \left(\nabla \cdot U^{ADV}\rho h\right)^{n+1/2} + \frac{1}{2}\left[\nabla \cdot \frac{\lambda^{n+1,*}}{c_{p,mix}^{n+1,*}}\nabla h^{n+1,*} + \nabla \cdot \frac{\lambda^n}{c_{p,mix}^n}\nabla h^n\right] \tag{17}$$
$$- \frac{1}{2}\sum_m \nabla \cdot \left[h_m^{n+1,*}\left(\Gamma_m^{n+1,*} + \frac{\lambda^{n+1,*}}{c_{p,mix}^{n+1,*}}\nabla Y_m^{n+1,*}\right) + h_m^n\left(\Gamma_m^n + \frac{\lambda^n}{c_{p,mix}^n}\nabla Y^n m\right)\right]$$

Finally, we update the predicted new-time temperature, $T^{n+1,*}$ using (5). This completes the predictor step of the advection diffusion algorithm.

Prior to the corrector step of the algorithm, the provisional new-time values of $(Y_m, h, T)^{n+1,*}$ are used to recompute new-time fluid properties, $(\lambda, h_m, \mathcal{D}_m, c_{p,mix})^{n+1}$. We then recompute $(Y_m, h)^{n+1}$. Y_m^{n+1} is obtained by solving

$$\frac{\rho^{n+1}\tilde{Y}_m^{n+1} - \rho^n Y_m^n}{\Delta t} + \left(\nabla \cdot U^{ADV}\rho Y_m\right)^{n+1/2} = \frac{1}{2}\nabla \cdot \left(\rho^{n+1}\mathcal{D}_m^{n+1}\nabla \tilde{Y}_m^{n+1} - \Gamma_m^n\right) \tag{18}$$

for the provisional $\tilde{Y}_m^{n+1}$. As before, the species diffusion fluxes computed from the solution of (18) do not sum to zero. We apply our flux redistribution scheme to obtain the adjusted species fluxes, Γ_m^{n+1} and then update Y_m^{n+1} using

$$\frac{\rho^{n+1}Y_m^{n+1} - \rho^n Y_m^n}{\Delta t} + \left(\nabla \cdot U^{ADV}\rho Y_m\right)^{n+1/2} = -\frac{1}{2}\nabla \cdot \left(\Gamma_m^{n+1} + \Gamma_m^n\right)$$

The final new-time enthalpy is then given by

$$\frac{\rho^{n+1}h^{n+1} - \rho^n h^n}{\Delta t} = \left(\nabla \cdot U^{ADV}\rho h\right)^{n+1/2} + \frac{1}{2}\left[\nabla \cdot \frac{\lambda^{n+1}}{c_{p,mix}^{n+1}}\nabla h^{n+1} + \nabla \cdot \frac{\lambda^n}{c_{p,mix}^n}\nabla h^n\right] \tag{19}$$
$$- \frac{1}{2}\sum_m \nabla \cdot \left[h_m^{n+1}\left(\Gamma_m^{n+1} + \frac{\lambda^{n+1}}{c_{p,mix}^{n+1}}\nabla Y_m^{n+1}\right) + h_m^n\left(\Gamma_m^n + \frac{\lambda^n}{c_{p,mix}^n}\nabla Y^n m\right)\right]$$

The final new-time temperature, T^{n+1} is computed by once again inverting (5), with h^{n+1} and Y_m^{n+1}.

Before discussing the incorporation of this methodology in an adaptive mesh refinement algorithm, we note some of the properties of the algorithm. First, we emphasize that the temperature equation is used only in an auxiliary capacity in the algorithm. The energy is evolved using the numerically conservative discretized enthalpy equation, (17) and (19). Second, since we explicitly ensure that $\sum_m \Gamma_m \equiv 0$ and guarantee that $\sum Y_m \equiv 1$, the sum of discrete species equations yields a conservative discretization of the continuity equation. As noted earlier, although the scheme rigorously satisfies conservation of mass and enthalpy, the evolution does not strictly maintain the equation of state at ambient pressure. The degree to which the equation of state is not satisfied will be illustrated in the results section. Since the low Mach number asymptotics used to derive the governing equation shows that

the thermodynamic pressure only satisfies (4) to $\mathcal{O}(M^2)$, relaxing the imposition of (4) is a reasonable way of dealing with the overdetermined system.

Our scheme also satisfies certain free-stream preservation properties. First, the method is designed so that for non-reacting isothermal flows the temperature remains constant independent of the velocity field and species distribution. The key issue in enforcing this property is the construction of the advective enthalpy flux at cell edges. As noted above, the Godunov edge states are constructed for temperature and species densities, then enthalpy flux is constructed from these values. The algorithm is also free-stream preserving for species mass fractions for the second, redistribution-based procedure for adjusting diffusive fluxes. In this case, if a mass fraction is initially constant and does not participate in any chemical reactions the algorithm will preserve that constant regardless of the velocity field or chemical reactions among the other species. Specifically, the flux redistribution scheme does not modify diffusion fluxes that were initially zero, so there is no mechanism to generate structure in the profile of that species. Note that this treatment departs from more traditional procedures employed to overcome the issue of discrete mass conservation, such as defining a designated "excess" species, or a "conservation diffusion velocity"[8], neither of which can be applied without potentially generating spurious signals in constant fields.

4. Local adaptive mesh refinement

In this section we present an overview of the adaptive projection algorithm. The framework is the same as that developed in Almgren *et al.*[1], extended to low Mach number combustion by Pember *et al.*[28]. As in the discussion of the single-grid time advance, we will focus primarily on modifications to the base algorithm that are required to incorporate differential diffusion and complex chemistry. We refer the reader to the above papers for more details of the basic algorithm.

Our implementation of AMR is based on a sequence of nested grids with successively finer spacing in both time and space. In this approach, fine grids are formed by evenly dividing coarse cells by a refinement ratio, r, in each direction. Increasingly finer grids are recursively embedded in coarse grids until features of the solution are adequately resolved. An error estimation procedure based on user-specified criteria evaluates where additional refinement is needed and grid generation procedures dynamically create or remove rectangular fine grid patches as resolution requirements change.

The adaptive integration algorithm advances grids at different levels using time steps appropriate to that level, based on CFL considerations. The multi-level procedure can most easily be thought of as a recursive algorithm in which, to advance level ℓ, $0 \le \ell \le \ell_{max}$, the following steps are taken:

- Advance level ℓ in time as if it is the only level. If $\ell > 0$, obtain boundary data using

time-interpolated data from the grids at $\ell - 1$, as well as physical boundary conditions, where appropriate.

- If $\ell < \ell_{max}$
  - Advance level $(\ell + 1)$ for r time steps, $\Delta t^{\ell+1} = \frac{1}{r}\Delta t^{\ell}$, using level-$\ell$ data and the physical boundary conditions.
  - Synchronize the data between levels ℓ and $\ell+1$, and interpolate corrections to finer levels $[\ell + 2, \ldots, \ell_{max}]$.

The adaptive algorithm, as outlined above, performs operations to advance the grids at each level independent of other levels in the hierarchy (except for boundary conditions) and then computes a correction to synchronize the levels. Loosely speaking, the objective in this synchronization step is to compute the modifications to the coarse grid that reflect the change in the coarse grid solution due to the presence of the fine grid. More specifically, when solving on a fine grid, we supply Dirichlet boundary conditions from the coarse grid. This leads to a mismatch in the associated fluxes at the coarse fine interface that is corrected by the synchronization.

For the adaptive projection methodology presented here there are three basic steps in the synchronization. First, the values obtained for U, ρY_m and ρh are averaged from the fine grid onto the underlying coarse grid. We view the resulting data as defining a preliminary composite grid solution that is consistent between levels. We will denote this preliminary solution with a p superscript in the remainder of the section. To complete the synchronization we need to correct inconsistencies arising from the use of Dirichlet boundary conditions at coarse-fine boundaries. First, we compute increments to ρY_m and ρh that correct the flux mismatches at coarse–fine interfaces. Finally, we correct the velocity field to satisfy a divergence constraint over the composite grid system.

There are two components that contribute to flux mismatch. First, U^{ADV}, the edge-based advection velocity satisfies the constraint on the coarse level and the fine level separately. However, since we only satisfy the Dirichlet matching condition for ϕ^{MAC} in (9), the value of U^{ADV} computed on the coarse level does not match the average value on the fine grid. We define the mismatch in advection velocities by

$$\delta U^{ADV,\ell} = -U^{ADV,\ell,n+\frac{1}{2}} + \frac{1}{r^2}\sum_{k=0}^{r-1}\sum_{edges} U^{ADV,\ell+1,n+k+\frac{1}{2}}$$

along the coarse–fine boundary. We then solve the elliptic equation

$$D^{MAC}\frac{1}{\rho}G^{MAC}\delta e^{\ell} = D^{MAC}\delta U^{ADV,\ell}$$

and compute

$$U^{ADV,\ell,corr} = -\frac{1}{\rho}G^{MAC}\delta e^{\ell}$$

which is the correction needed for U^{ADV} to satisfy the constraint and matching conditions on the composite $(\ell, \ell+1)$ grid hierarchy. This correction field is used to compute a modification to the advective fluxes for species and enthalpy that reflects an advection velocity field that satisfies the constraint on the composite grid.

The second part of the mismatch arises because the advective and diffusive fluxes on the coarse grid were computed without explicitly accounting for the fine grid, while on the fine grid the fluxes were computed using coarse-grid Dirichlet boundary data. We define the flux discrepancies

$$\delta F_{\rho h} = \Delta t^{\ell} \left(-F_{\rho h}^{\ell,n+1/2} + \frac{1}{r^2} \sum_{k=0}^{r-1} \sum_{edges} F_{\rho h}^{\ell+1,n+k+1/2} \right)$$

and

$$\delta F_{\rho Y_m} = \Delta t^{\ell} \left(-F_{\rho Y_m}^{\ell,n+1/2} + \frac{1}{r^2} \sum_{k=0}^{r-1} \sum_{edges} F_{\rho Y_m}^{\ell+1,n+k+1/2} \right)$$

where F is to total (advective+diffusive) flux through a given interface prior to these synchronization operations. Since mass is conserved, corrections to density, $\delta\rho^{sync}$, on the coarse grid associated with mismatched advection fluxes may be computed explicitly

$$\delta\rho^{sync} = -D^{MAC} \left(\sum_m U^{ADV,corr} \rho Y_m \right)^{n+1/2} + \sum_m \delta F_{\rho Y_m}$$

The post-sync new-time value of density, $\rho^{n+1} = \rho^{n+1,p} + \delta\rho^{sync}$. Given the corrected density ρ^{n+1} we can decompose the corrections for Y_m and h into

$$\delta\left(\rho Y_m\right)^{sync} = Y_m^{n+1,p} \delta\rho^{sync} + \rho^{n+1} \delta Y^{sync}$$

and

$$\delta\left(\rho h\right)^{sync} = h^{n+1,p} \delta\rho^{sync} + \rho^{n+1} \delta h^{sync}$$

Computing δY_m^{sync} and δh^{sync} requires solution of a linear system, since the flux mismatch contains implicit diffusion fluxes from the Crank-Nicolson discretization in $\mathcal{H}_{dt}^{A-D}$. The provisional correction $\delta\tilde{Y}_m^{sync}$ on the coarse level-ℓ grids is obtained by solving

$$\left(\rho^{n+1} - \frac{\Delta t}{2} \nabla \rho^{n+1} \mathcal{D}_m^{n+1} \nabla \right) \delta\tilde{Y}_m^{sync} = -D^{MAC} \left(U^{ADV,corr} \rho Y_m \right)^{n+1/2} + \delta F_{\rho Y_m} \tag{20}$$

However, as in the single-level algorithm, the species correction fluxes must sum to zero to preserve mass conservation. We compute the adjusted species correction flux, $\delta\Gamma^{sync}$ from $\delta\tilde{\Gamma}_m^{sync} \equiv -\rho^{n+1} \mathcal{D}_m^{n+1} \nabla \delta\tilde{Y}_m^{sync}$ so that $\sum_m \delta\Gamma_m^{sync} \equiv 0$ using one of the procedures discussed above. This adjusted flux is used to compute the correction, δY_m^{sync} needed to update Y_m and

to form the forcing term due to non-unity Lewis number effects in the correction equation for ρh:

$$\left(\rho^{n+1} - \frac{\Delta t}{2}\nabla\rho^{n+1}\frac{\lambda^{n+1}}{c_{p,mix}^{n+1}}\nabla\right)\delta h^{sync} = -D^{MAC}\left(U^{ADV,corr}\rho h\right)^{n+1/2} + \delta F_{\rho h}$$
$$+\nabla\cdot\sum_m h_m(T^{n+1,p})\left(\frac{\lambda^{n+1}}{c_{p,mix}^{n+1}}\nabla\delta Y_m^{sync} + \delta\Gamma_m^{sync}\right) \quad (21)$$

The corrections, δY_m^{sync}, and δh^{sync} are added to the coarse field at level-ℓ, and interpolated to all finer levels. Finally, a new temperature field is computed using (5) on all affected levels.

A similar process is also used to generate a correction to the velocity field. However, the velocity flux correction must be projected to obtain the component satisfying the constraint that updates U and the component that updates π. At this point there are two additional corrections needed for the composite velocity field:

- A correction arising because the projection at level $\ell+1$ used Dirichlet data from level ℓ, leading to a mismatch in normal derivative at coarse-fine boundaries
- The temperature and species adjustment in the first part of the synchronization leads to an increment in the computed $\hat{S}$ field.

Since the projection is linear, both of these corrections as well as the projection of the velocity flux correction can be combined into a single, multi-level node-based synchronization solve performed at the end of a coarse-grid time step.

5. Numerical results

In this section we present computational results from two different combustion regimes to validate the numerical method presented above. In the first example, we model a premixed hydrogen flame, using the results to illustrate the convergence properties of the single-grid algorithm and to demonstrate that the adaptive algorithm effectively computes the same solution as a uniform fine grid. In the second example, we illustrate the behavior of the method on a steady and an unsteady methane diffusion flame.

Premixed Flame

As an initial test of the methodology we apply our algorithm without refinement to the evolution of a one-dimensional premixed hydrogen flame. The chemistry mechanism (consisting of 9 species and 27 reactions), and associated thermodynamic and transport databases were generated for this case by stripping the carbon chemistry from the GRI-Mech-1.2 [11] distribution. The solution is initialized with a refined steady solution computed using

Table 1. Convergence rates, R^N, for L^N norm of error, measured as deviation from the PREMIX steady solution. Rates for the local dominant species method appear in the left side of the table; rates for the redistribution method appear on the right side.

	Dominant Species			Redistribution		
Quantity	R^1	R^2	R^∞	R^1	R^2	R^∞
T	2.12	1.98	1.57	1.98	1.90	1.50
V	2.42	2.42	2.42	2.14	2.15	2.16
H	2.49	2.32	2.06	1.62	1.48	1.39
ρ	2.13	1.86	1.34	1.80	1.66	1.16
Y_{H2}	2.00	1.98	1.76	1.64	1.98	2.22
Y_H	3.04	2.76	2.43	2.92	2.67	2.37
Y_O	2.53	2.47	2.25	2.46	2.41	2.13
Y_{O2}	1.98	2.08	1.87	1.99	2.15	1.98
Y_{OH}	2.83	2.78	2.09	2.87	2.89	2.07
Y_{H2O}	1.98	2.04	1.69	1.45	1.58	1.38
Y_{HO2}	1.35	1.08	0.68	1.40	1.11	0.71
Y_{H2O2}	1.64	1.67	1.36	1.68	1.71	1.39
Y_{N2}	1.97	1.73	1.33	0.40	0.47	0.65

the PREMIX code from the CHEMKIN-III library [15]. The inlet stream (mole fractions $X(\mathrm{H_2} : \mathrm{O_2} : \mathrm{N_2}) = (0.1909 : 0.0910 : 0.7181)$) enters at 79.675 cm s^{-1} and 298K. The initial profile exhibits a peak $X_{\mathrm{H_2O_2}} = 5.83 \times 10^{-5}$ centered at 0.57 cm in a 1.6 cm domain, and is evolved for 80μs on uniform grids of 64, 128 and 512 cells. The time step used in each case was $\Delta t = 4, 2$ and 0.5μs, respectively.

In Table 1 we give rates of convergence in L^1, L^2, and L^∞, computed by comparing errors over the 64- and 128-point resolutions to the (steady) PREMIX solution used to initialize the problem. Results are presented for both approaches to adjusting diffusive fluxes, as described above. For the first flux adjustment approach, based on a locally dominant species, each quantity is converging with second-order accuracy in L^1, with the exception of trace species in the flame zone. For the redistribution-based flux adjustment algorithm, apparent accuracy degradation is more evident, most notably the behavior of $Y_{\mathrm{N_2}}$. This reflects the difference in treatment of the two flux adjustment approaches. Although $Y_{\mathrm{N_2}}$ should remain constant for this problem, $\mathrm{N_2}$ is the dominant species throughout the domain and absorbs the entire corrective flux in the PREMIX algorithm as well as in our first approach.

We also present, in Table 2, convergence data for each approach compared to a high-resolution solution computed on a 512-point grid. The convergence rate in max norm is 1.35 or higher with the exception of $\mathrm{HO_2}$ which forms a narrow spike in the solution. We also observe that both approaches to correcting the diffusive fluxes from the mixture model seem to be effective with neither approach emerging as a clear preference. A more detailed

Table 2. Convergence rates, R^N, for L^N norm of error, measured as deviation from a fine-grid solution. Rates for the local dominant species method appear in the left side of the table; rates for the redistribution method appear on the right side.

	Dominant Species			Redistribution		
Quantity	R^1	R^2	R^∞	R^1	R^2	R^∞
T	2.25	2.02	1.58	2.24	2.01	1.56
V	2.18	2.18	2.19	2.13	2.13	2.14
H	2.47	2.39	2.10	2.29	2.21	2.09
ρ	2.21	1.85	1.40	2.16	1.85	1.35
Y_{H2}	2.31	2.23	1.89	2.51	2.28	1.93
Y_H	2.62	2.57	2.48	2.61	2.56	2.44
Y_O	2.59	2.54	2.22	2.59	2.54	2.18
Y_{O2}	2.26	2.25	2.02	2.27	2.25	2.04
Y_{OH}	2.83	2.80	2.18	2.85	2.83	2.20
Y_{H2O}	2.12	2.12	1.75	1.99	2.05	1.76
Y_{HO2}	1.43	1.15	0.74	1.45	1.16	0.76
Y_{H2O2}	1.74	1.74	1.38	1.76	1.75	1.40
Y_{N2}	2.83	2.44	1.88	2.45	2.20	1.95

examination of these approaches will appear in a future work.

As a more interesting test of the methodology, we use the same premixed hydrogen flame as the initial condition for a two-dimensional vortex flame interaction. The problem specification is similar to problems considered by Najm *et al* [26]. Specifically we extend the laminar flame solution to two space dimensions and superimpose the velocity field induced by a countersign vortex pair (see Figure 1). The vortices have Gaussian cores with centers 0.2 cm apart and generate a maximum rotational velocity of 14.6 m s^{-1}. The vortex pair induces a self-propagation velocity of 3.1 m s^{-1}. We exploit symmetry and only compute on the left half of the domain which is 0.4 cm wide and 1.6 cm high.

First, we solve the problem on a uniform grid with mesh spacing of 128×512. We then repeat the problem with adaptive refinement on a 32×128 base grid with two levels of refinement by a factor of two each. We trigger the finest grid on the flame zone using H_2O_2 as a marker for the flame. We also refine regions of high vorticity with one factor of two refinement. In Figure 2 we show the enthalpy, temperature and $X_{\mathrm{H_2O_2}}$ at 53.6μs. For each figure we show the adaptive solution on the left and the (reflected) uniform solution on the right. The results are in excellent agreement suggesting that the adaptive algorithm is producing essentially the same result as a uniform fine grid. Figure 3 presents a similar comparison of uniform and adaptive solutions at 85.3μs. Again the results are in good agreement. At the later time we do note that the top of the flame above the vortex is slightly lower in the adaptive case. This shows that the resolution of the vortex, which is

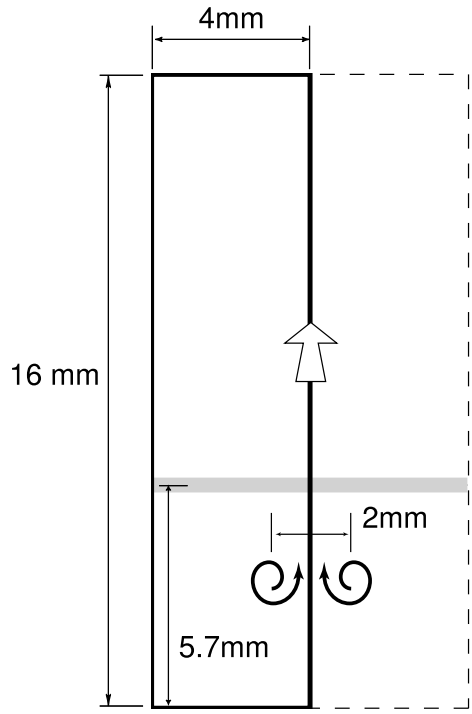

Figure 1. Schematic of the premixed hydrogen flame/vortex problem. The shaded line represents the position of the flame. Cold unburnt fuel enters the bottom of the domain and products exit the top boundary. The swirl lines represent vortical flow due to the counter rotating vortex pair. The modelled domain is 0.4×1.6 cm; symmetry is imposed along the sides to avoid modelling the dotted region.

represented on a coarser grid, does not quite match the fine grid solution.

Diffusion flame

As a second example, we apply our methodology to a laminar diffusion flame modelled by Mohammed *et al* [24]. In their work, an axisymmetric diffusion flame is modelled using a 26-species, 83-reaction methane mechanism presented by Smooke *et al* [33]. We incorporate the databases provided with CHEMKIIN-III [16] for evaluating thermodynamic and transport properties. The problem geometry is specified in Figure 4. Following the work of Mohammed *et al.*, we consider both a steady flame and a transient flame where the transient behavior is induced by a 20Hz perturbation in the fuel inflow velocity. In particular, the fuel (CH_4, diluted with 35% N_2 by volume) is injected with the velocity profile

$$v = 70 \left(1 - \frac{r^2}{R_1^2}\right) \left[1 + \epsilon \sin(\omega t)\right] \quad \text{cm s}^{-1}. \tag{22}$$

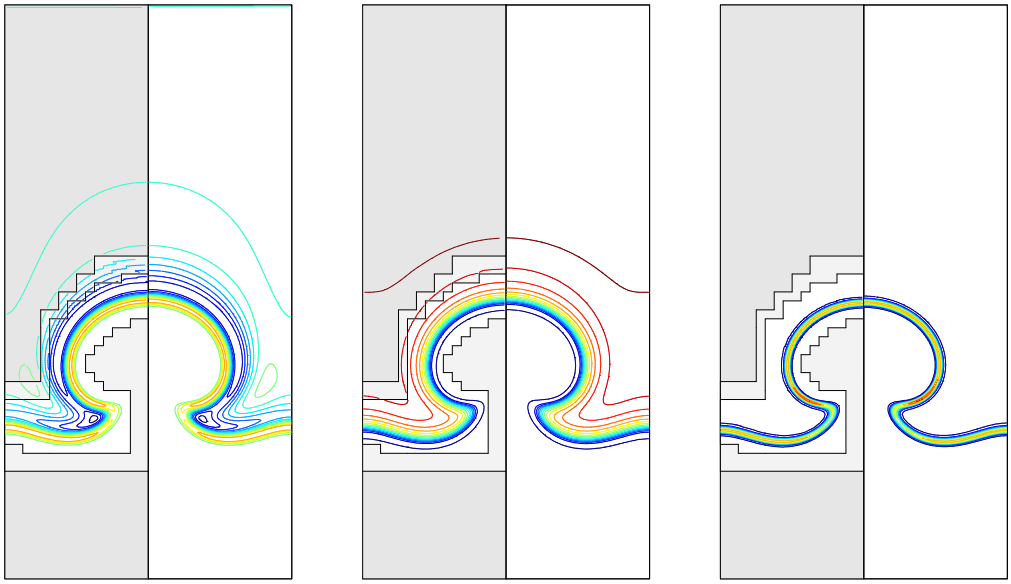

Figure 2. Enthalpy, Temperature, and $X_{\mathrm{H_2O_2}}$ contours at $t = 53.6\mu$s. In the AMR results (left), regions covered by fine grid ($\Delta x = 31.25\mu m$) are unshaded, while less refined areas ($\Delta x = 62.5, 125\mu m$) are shaded progressively darker. The single-grid results have been reflected about $x = 0.4cm$ to ease comparison.

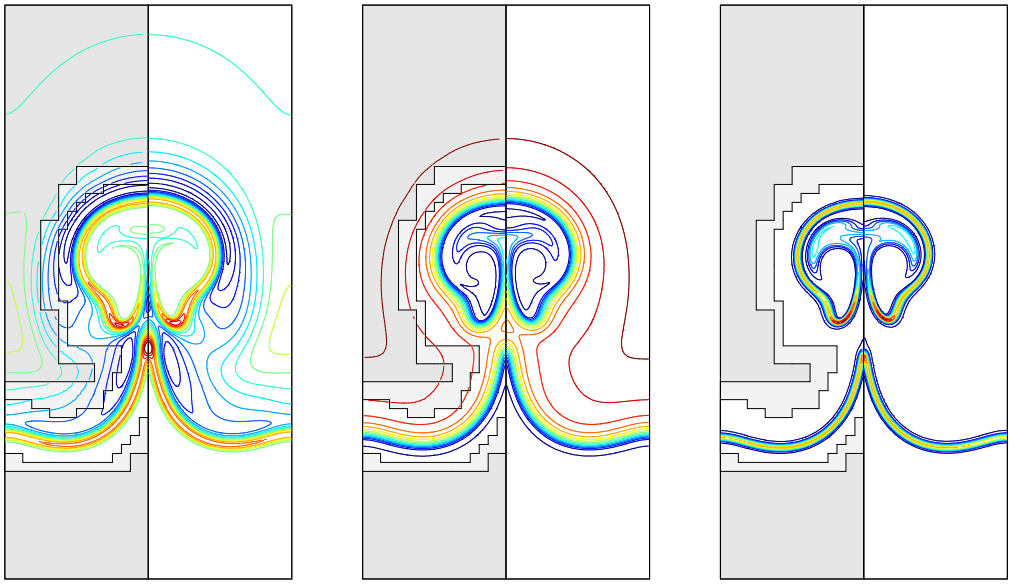

Figure 3. Enthalpy, Temperature, and X_{H2O2} contours at $t = 85.3\mu$s. In the AMR results (left), regions of fine grid ($\Delta x = 31.25\mu m$) are unshaded, while less refined areas ($\Delta x = 62.5, 125\mu m$) are shaded progressively darker. The single-grid results have been reflected about $x = 0.4cm$ to ease comparison.

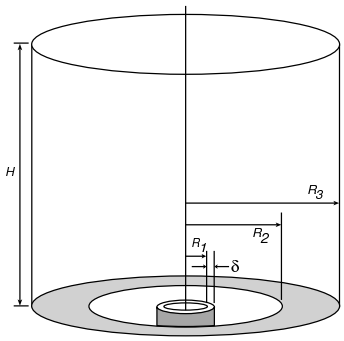

Figure 4. Schematic of the diffusion flame setup. Here, $R_1 = 2$ mm, $\delta = 0.38$ mm, $R_2 = 2.5$ cm, $R_3 = 5.12$ cm, and $H = 10.24$ cm.

Coflow air is injected at 35 cm s^{-1} through the annular region between $R_1 + \delta$ and R_2. The region between R_2 and R_3, and the vertical sides of the chamber are bounded by a wall at 298K, and the top is open to atmospheric pressure. Note the outer wall radius and domain height in our specification are smaller those in [24]. We found that the details of the flame were insensitive to either of these parameters. The entire computational domain is filled initially with room temperature air, except for a small hot spot spanning the fuel and air inlets to ignite the flame. We then allow the flow to evolve numerically on a coarse grid ($\Delta x = 0.08$ cm) to a stable flame pattern. Interestingly, we found that very small time steps were required during the early transient. Attempting to use larger time steps based on CFL considerations on these coarse grids led to problems with the flame blowing out. We conjecture that this behavior is related to an interplay between chemistry and diffusion that is not adequately represented with large time steps in the present operator-split framework.

For the steady case ($\epsilon = 0$ in (22)), we computed the solution with a base grid of 32×64, plus two levels of additional refinement. The first level refines the base grid by a factor of two based on temperature in order to capture the gross features of the flame. The finest level is a factor of four finer for an effective resolution of 256×512, and localizes grids around regions of high CH concentration. We studied the behavior of our solution with further refinement by performing two additional computations beginning with the steady three-level solution, and adding additional localized grid based on X_{CH}, refined by a factor of two or four, for effective resolutions of 0.01 cm. and 0.005 cm, respectively. Each of these runs were then time-advanced until their solutions again relaxed to steady state.

Raster images of temperature and X_{CH} are shown in Figures 5 and 6, respectively. For temperature, we show results only for the most refined solution, as the three cases are virtually indistinguishable. The computed results have been mirrored across $r = 0$ for ease of comparison with the experimental data presented in [24]. For these steady results we compute a flame lift-off height of 0.27 cm for the base case and 0.23 cm for the two finer

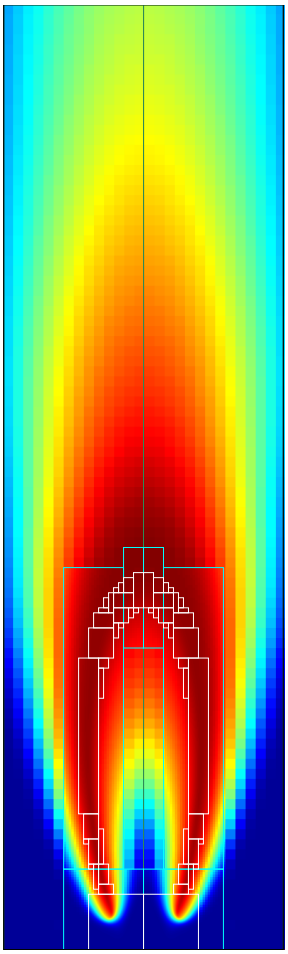

Figure 5. Refined temperature field (in K) for the steady diffusion flame, reflected about the origin. The subregion shown spans $\pm 1.12 \times 7.52$ cm. Grids bounded by white, cyan and green boxes have $\Delta x = 0.005, 0.02$ and 0.08 cm, respectively.

cases. (Following Mohammed *et al* we define the flame lift-off height as the lowest z location where the flame reaches $1000K$.) Mohammed *et al* report an experimental lift-off height of between 0.16 and 0.22 cm but they obtain a computational lift-off height of 0.66 cm. We also compute a maximum centerline temperature of $2029K$ (compared with $2025K$ reported by Mohammed *et al* without radiative losses) occurring at 3.08 cm on the base computation and 3.17 cm for the two finer cases. These data give a predicted flame length of 2.81 cm for the base case and 2.94 cm for the refined cases.

One feature of our methodology is that we conserve both enthalpy and species mass density. In doing so our computed solution does not maintain thermodynamic pressure exactly at ambient conditions. In Figure 7 we show the thermodynamic pressure computed from the species densities and enthalpy obtained from the numerical solution. For this

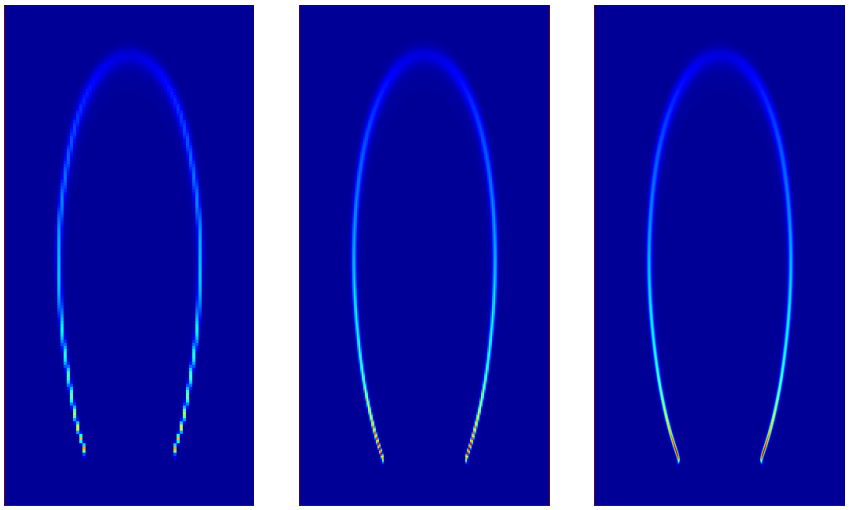

Figure 6. X_{CH} for the steady diffusion flame, reflected about the origin. The subregion shown spans $\pm 0.8 \times 3.2$ cm. The finest grids are clustered around the peak X_{CH} values, and have mesh spacings $\Delta x = 0.02, 0.01$ and 0.005 cm, in the figures from left to right, respectively.

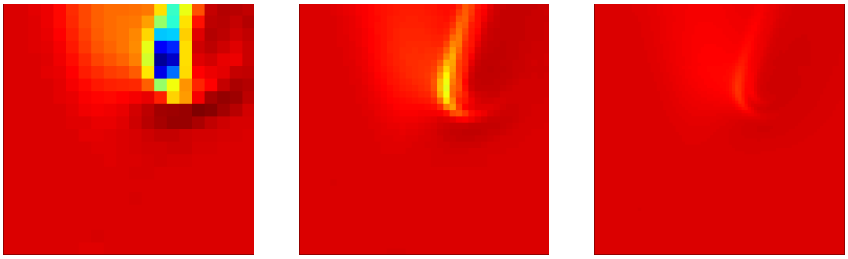

Figure 7. Computed thermodynamic pressure in Pa at the flame tip. Maximum error values with respect to atmospheric pressure, from left to right are 6075, 2107 and 556 Pa, for respective grid spacings, $\Delta x = 0.02, 0.01, 0.005$ cm.

problem the ambient pressure is constant at 10132.5 Pa. The figure shows the computed results in a small 0.4 cm square region near the inlet and the flame tip. The results show that the deviation from ambient pressure converges to zero as the mesh is refined, and does not build up over long-time integrations.

For the final example we model the transient version of the Mohammed *et al* diffusion flame. We consider two levels of the perturbation ϵ in Equation 22. These computations were performed at the same effective resolution as the coarse case in the steady simulations above (ie. $\Delta x = 0.02$ cm). In Figures 8 and 9 we show a time sequence of temperature and CH mole fraction through one period of oscillation of the flame for the weaker perturbation. In Figures 10 and 11 we show the comparable results for the stronger perturbation. For the weaker perturbation we see excellent agreement in the shape of the CH profiles with

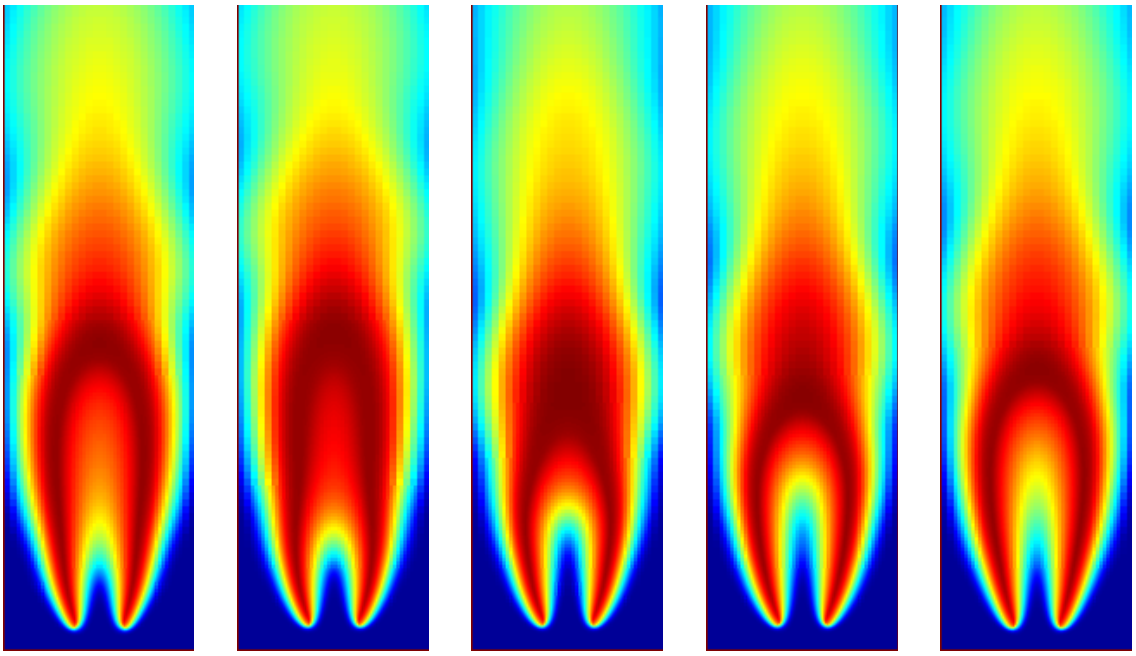

Figure 8. Temperature (in K) for $\epsilon = 0.25$, at 0.01, 0.02, 0.03, 0.04, and 0.05 s, reflected about the origin. The region shown spans $\pm 1.12 \times 7.52$ cm.

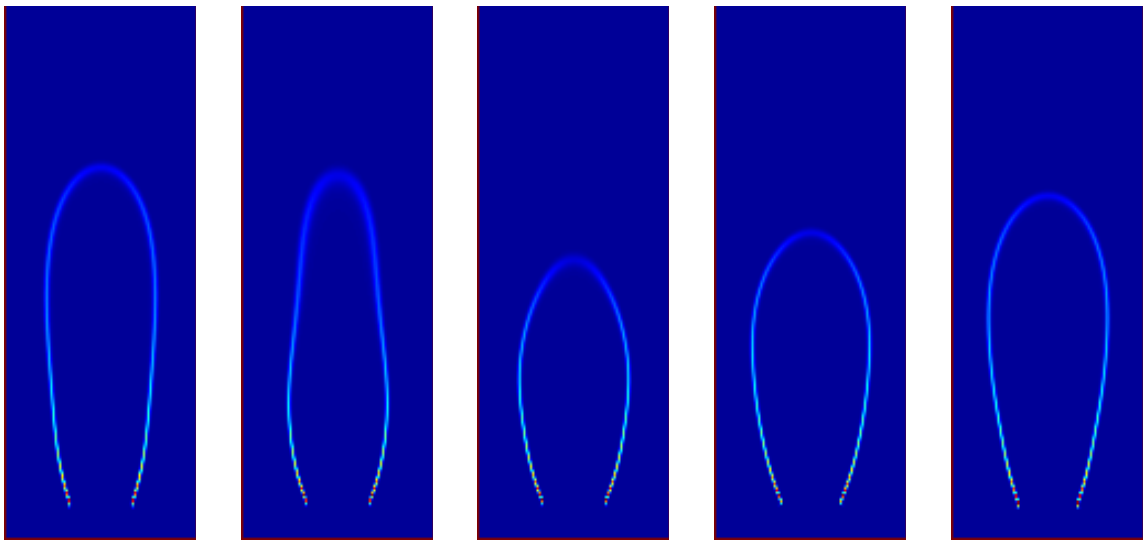

Figure 9. X_{CH} for $\epsilon = 0.25$, at 0.01, 0.02, 0.03, 0.04, and 0.05 s, reflected about the origin. The region shown spans $\pm 0.88 \times 4.88$ cm.

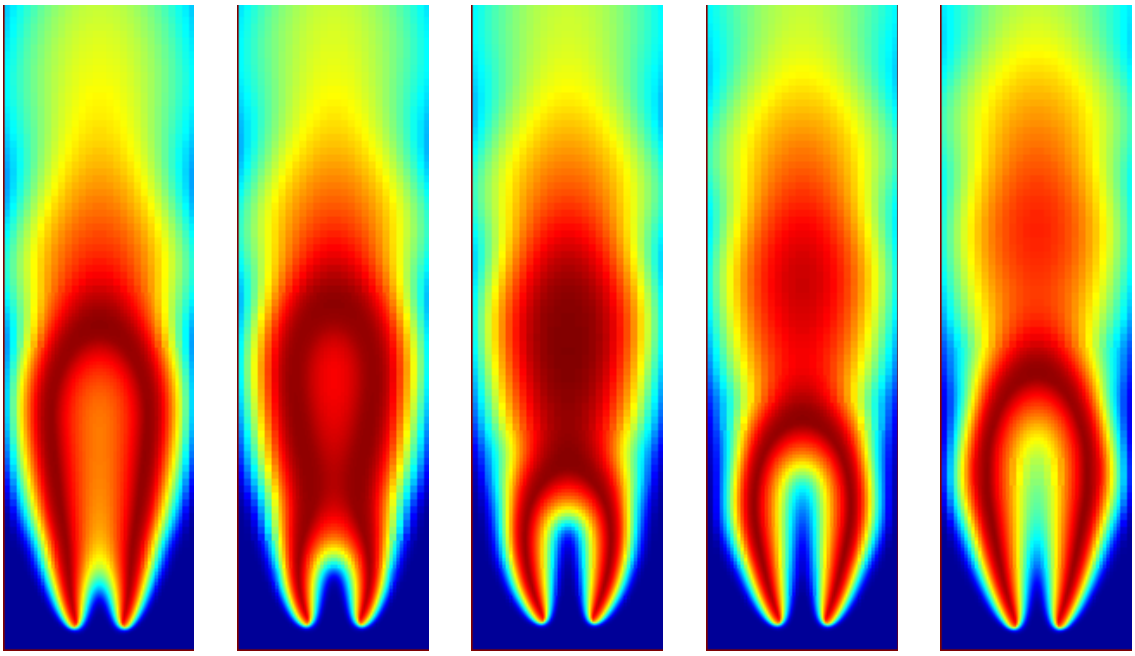

Figure 10. Temperature (in K) for $\epsilon = 0.50$, at 0.01, 0.02, 0.03, 0.04, and 0.05 s, reflected about the origin. The region shown spans $\pm 1.12 \times 7.52$ cm.

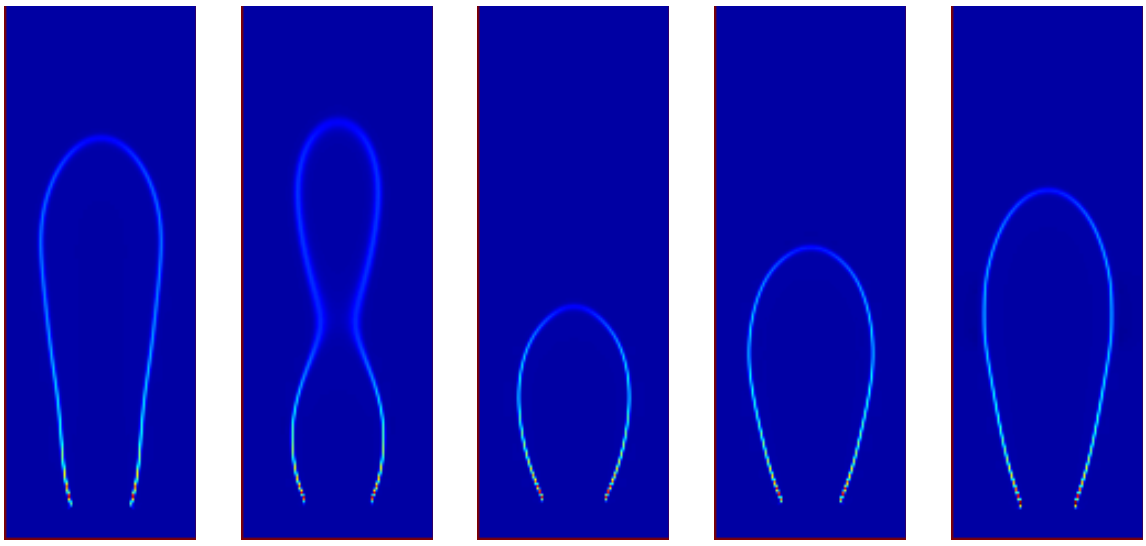

Figure 11. X_{CH} for $\epsilon = 0.50$, at 0.01, 0.02, 0.03, 0.04, and 0.05 s, reflected about the origin. The region shown spans $\pm 0.88 \times 4.88$ cm.

the experimentally observed profiles from Mohammed *et al* ; however, the overall extent of the variation is less than the experimentally observed values. The evolution for the larger perturbation is more violent than the experimental observations suggesting that the level of perturbation in the experiment is between 0.25 and 0.50. The experimental perturbation is not reported in [24].

6. Summary and conclusion

In this paper we have presented a new adaptive algorithm for solving the equations governing low Mach number combustion with complex kinetics. The overall approach is based on a projection formulation and uses a discrete form of the low Mach number equations that conserve both species and enthalpy. The extension to adaptive refinement uses subcycling in time with a synchronization step whenever coarse and fine grids reach the same time that enforces the conservation and free-stream preservation properties of the single grid algorithm.

We have demonstrated that the method converges at or near second-order for velocity, temperature and species and that the adaptive algorithm is able to compute essentially the same solution as a uniform fine grid for premixed combustion. We have also demonstrated convergence of the method for a diffusion flame and have correctly predicted the flame lift off height for a co-flowing methane/air diffusion flame.

The adaptive algorithm here is implemented using the parallel BOXLIB framework developed by Rendleman *et al* [31, 9] so that the code can run effectively on modern parallel architectures. A more detailed discussion of parallelization issues for the methodology presented here will be given in future work. We are also pursuing a number of algorithm enhancements and extensions such as models for radiation and a more complete model for transport. Finally, we are exploring a range of applications of this methodology to the study of fundamental issues in methane and hydrogen combustion.

Acknowledgments

The software implementation of the algorithms described in this paper incorporates the combined efforts of a number of people. Vincent Beckner, William Crutchfield, Charles Rendleman and Michael Lijewski provided considerable enhancements to the underlying BOXLIB, visualization and linear solver infrastructure. Our code evolved from a unity Lewis number implementation by Rick Pember, and depends heavily on software built for adaptive calculations of incompressible flow with significant contributions by Ann Almgren, Louis Howell, Michael Welcome and all the folks mentioned above. This work was carried out at the Lawrence Berkeley National Laoratory under the auspices of the US Department of Energy Office of Mathematics, Information and Computational Sciences under contract number DE-AC03-76SF00098.

[1] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, 142:1–46, 1998.

[2] A. S. Almgren, J. B. Bell, and W. Y. Crutchfield. Approximate projection methods: Part I. Inviscid analysis. Technical Report LBNL–43374, LBNL, May 1999. submitted for publication.

[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. *SIAM J. Sci. Comput.*, 17(2), March 1996.

[4] B.A.V. Benntt and M. D. Smooke. Local rectangular refinement with application to fluid flow problems. *J. Comput. Phys.*, to appear.

[5] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh. VODE: A variable coefficient ode solver. *SIAM J. Sci. Stat. Comput.*, 10:1038–1051, 1989.

[6] I.-L. Chern and P. Colella. A conservative front tracking method for hyperbolic conservations laws. Technical Report UCRL-97200, LLNL, July 1987.

[7] 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.

[8] T. P. Coffee and J. M. Heimerl. Transport algorithms for premixed laminar steady-state flames. *Comb. Flame*, 43:273, 1981.

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

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

[11] M. Frenklach, H. Wang, M. Goldenberg, G. P. Smith, D. M. Golden, C. T. Bowman, R. K. Hanson, W. C. Gardiner, and V. Lissianski. GRI-Mech—an optimized detailed chemical reaction mechanism for methane combustion. Technical Report GRI-95/0058, Gas Research Institute, 1995. http://www.me.berkeley.edu/gri_mech/.

[12] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces. *Physics of Fluids*, 8:2182–2189, 1965.

[13] J. Hilditch and P. Colella. A front tracking method for compressible flames in one dimension. *SIAM J. Sci. Comput.*, 16:755–772, 1995.

[14] R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miler. A FORTRAN computer code package for the evaluation of gas-phase multicomponent transport properties. Technical Report SAND86-8246, Sandia National Laboratories, 1986.

[15] R. J. Kee, J. F. Grcar, J. A. Miller, E. Meeks, and M. Smooke. *PREMIX Users Manual*. Reaction Design, San Diego, CA, 1998. (www.ReactionDesign.com).

[16] R. J. Kee, F. M. Ruply, E. Meeks, and J. A. Miller. CHEMKIN-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics. Technical Report SAND96-8216, Sandia National Laboratories, 1996.

[17] R. J. Kee, J. Warnatz, and J. A. Miller. A FORTRAN computer code package for the evaluation of gas-phase viscosities, conductivities, and diffusion coefficients. Technical Report SAND83-8209, Sandia National Laboratories, 1983.

[18] O.M. Knio, H. N. Najm, and P. S. Wyckoff. A semi-implicit numerical scheme for reacting flow. II. Stiff, operator split formulation. *J. Comput. Phys.*, submitted for publication.

[19] M. F. Lai. *A Projection Method for Reacting Flow in the Zero Mach Number Limit*. PhD thesis, University of California at Berkeley, 1993.

[20] M. F. Lai, J. B. Bell, and P. Colella. A projection method for combustion in the zero Mach number limit. In *Proceedings of the Eleventh AIAA Computational Fluid Dynamics Conference*, pages 776–783, 1993.

[21] 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.

[22] 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.

[23] P.A. McMurtry, W.-H. Jou, J.J. Riley, and R.W. Metcalfe. Direct numerical simulations or a reacting mixing layer with chemical heat release. *AIAA J.*, 24:962, 1986.

[24] R. H. Mohammed, M. A. Tanoff, M. D. Smooke, and A. M. Schaffer. Computational and experimental study of a forced, time-varying, axisymmetric, laminar diffusion flame. In *Twenty-Seventh Symposium (International) on Combustion*, volume 1, pages 693–702. The Combustion Institute, 1998.

[25] H. N. Najm, P. S. Wyckoff, and O.M. Knio. A semi-implicit numerical scheme for reacting flow. I. Stiff chemistry. *J. Comput. Phys.*, 143:381–402, 1998.

[26] H.N. Najm, R.W. Schefer, R.B. Milne, C.J. Mueller, K.D. Devine, and S.N. Kempka. Numerical and experimental investigation of vortical flow–flame interaction. Technical Report SAND98-8232, Sandia National Laboratory, February 1998.

[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. *Transport Phenomena in Combustion*, pages 1200–1211, 1996. Taylor and Francis.

[28] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jessee. An adaptive projection method for unsteady, low-Mach number combustion. *Comb. Sci. Tech.*, 140:123–168, 1998.

[29] Richard B. Pember, Ann S. Almgren, William Y. Crutchfiel d, Louis H. Howell, John B. Bell, Phillip Colella, and Vincent E. Beckne r. An embedded boundary method for the modeling of unsteady combus tion in an industrial gas-fired furnace. In *1995 Fall Meeting of the Western States Section of the Combustion Institute*, October 1995. WSS/CI 95F0165.

[30] 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.

[31] C. A. Rendleman, V. E. Beckner, M. Lijewski, W. Y. Crutchfield, and J. B. Bell. Parallelization of structured, hierarchical adaptive mesh refinement algorithms. Technical Report LBNL-43154, LBNL, April 1999. accepted, Computing and Visualization in Science.

[32] C.J. Rutland and J.H. Ferziger. Simulations of flame-vortex interactions. *Combust. Flame*, 84:343, 1991.

[33] M. D. Smooke, Y. Xu, R. M. Zurn, P. Lin, J. H. Frank, and M. B. Long. Computational and experimental study of oh and ch radicals in axisymmetric laminar diffusion flames. In *Twenty-Fourth Symposium (International) on Combustion*, pages 813–821. The Combustion Institute, 1992.

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

[35] M. M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. Howell, and M. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148:81–124, 1999.

[36] M.D. Smooke A.A. Turnbull, , R.E. Mitchell, and D.E. Keyes. Solution of two-dimensional axisymmetric laminar diff usion flames by adaptive boundary value methods. In *Mathematical Modeling in Combustion and Related Topics*, pages 261–300, 1988.

[37] J. Warnatz. Influence of transport models and boundary conditions on flame structure. In N. Peters and J. Warnatz, editors, *Numerical methods in flame propagation*. Friedr. Viewweg and Sohn, Wiesbaden, 1982.