SIMULATION OF UNSTEADY COMBUSTION PHENOMENA USING COMPLEX MODELS *

T. Hagstrom†, K. Radhakrishnan‡, S. Steinberg§, and R. Zhou¶

Abstract

In this paper, we describe a numerical method for simulating flame propagation using complex physical and chemical models. A 4th order discretization is used for the spatial derivatives. For the temporal integrations, we use preconditioning to produce a highly efficient linearly implicit method with good stability properties. Spectral deferred correction is then employed to get higher order accuracy in time. Solution adapted moving grids are used to track the flame fronts. We test our method for the hydrogen-air system using plane flame solutions as initial data. The results demonstrate the accuracy, stabilty, and efficiency of the method. As a final application, we use our code to compute the quenching distance of the flame front impinging normally onto a cold wall.

1 Introduction

The propagation of laminar flames is governed by a system of partial differential equations that express the conservation of mass, momentum, energy, and species, combined, in the zero Mach number limit considered here, with some nonlinear algebraic equations, e.g. see [15]. The solutions of these partial differential-algebraic equations (PDAEs) typically have a steep front where the solution changes rapidly in a small spatial region. These equations are difficult to solve numerically because of the strong coupling among the chemical species and temperature which results in a high degree of stiffness in any accurate discrete approximation. The goal of our research is to produce a discretization with a high order of accuracy on non-uniform grids and a time-step stability constraint that is independent of the spatial discretization, because such approximations can be solved efficiently when combined with adaptive grid-generation strategies.

This paper considers the one-spatial-dimension problem for which we produce an algorithm that is fourth-order accurate and for which the time-step stability constraint, for uniform grids, is constant for sufficiently resolved grids. We provide strong numerical evidence that when this method is combined with adaptive moving-grids, the resulting simulations are accurate, stable, and efficient. As a simple application of our method, we compute the quenching distance of a flame impinging normally onto a cold wall.

2 One Dimensional Model

The one-dimensional model describes the dynamics of plane flames. The equations are derived from the general equations given in [15]. In an open system, the pressure is almost constant in space and in time, so we assume the pressure is constant and ignore the momentum conservation equation. This yields the simplified conservation equations:

$$\rho_t + (\rho v)_x = 0\,, \tag{1}$$

$$e_t + v\, e_x = -\frac{q_x}{\rho} - P\,\frac{v_x}{\rho}\,, \tag{2}$$

$$Y^i_t + v\, Y^i_x = \frac{\omega_i}{\rho} - \frac{1}{\rho}\,(\rho Y^i V^i)_x\,, \tag{3}$$

where, in the simulations described below, the independent variables satisfy $0 \le x \le 1$ (the unit of length is the centimeter) and $t \ge 0$. The dependent

*Copyright ©1999 by the authors. Published by the American Institute of Aeronautics and Astronaautics, Inc. with permission

†Department of Mathematics and Statistics, The University of New Mexico. Work supported in part by NASA Grant NRA-96-LeRC-2, NSF Grant DMS-9600146 and by ICOMP, NASA Glenn Research Center.

‡ICOMP, NASA Glenn Research Center, Cleveland, OH 44135.

§Department of Mathematics and Statistics, The University of New Mexico, Albuquerque NM 87131.

¶Department of Mathematics and Statistics, The University of New Mexico, Albuquerque NM 87131. Work supported in part by NASA Grant NRA-96-LeRC-2

variables are: e, the total energy; v, the average velocity; Y^i, the mass fraction of species i divided by its molecular weight. N_s will be used to denote the total number of species. The pressure P is assumed constant. The density ρ and the temperature T are determined from the gas law equations:

$$\frac{P}{R} = \rho T \sum_{i=1}^{N_s} Y^i , \tag{4}$$

$$e = \sum_{i=1}^{N_s} Y^i \, e_i(T) , \tag{5}$$

where, e_i are the species internal energy, and R is the gas constant. To complete the problem specification, we have to describe the computation of the reaction rates, ω_i, the diffusion velocities, V_i, the heat flux, q, and the species energy. In this work, ω_i were computed using the procedures built into the NASA Glenn kinetics code LSENS [12]. The NASA Glenn polynomial expressions [5] were used to calculate the thermodynamic properties, and the routines for the transport properties were adapted from the Sandia 1D flame code [9].

Equations (1)–(5) are a system of partial differential-algebraic equations that have index 2. Equations of index greater that one are difficult to solve numerically [1] [8]. Therefore we derive an explicit equation for the velocity v. and thus reduce the index of system of the PDAEs to 1. For convenience, define:

$$\begin{aligned}
s_1 &= \sum_{i=1}^{N_s} Y^i \, e_T^i(T) , \\
s_2 &= \sum_{i=1}^{N_s} Y^i , \\
s_3 &= \sum_{i=1}^{N_s} [\frac{\omega_i}{\rho} - \frac{1}{\rho} (\rho \, Y^i \, V^i)_x] , \\
s_4 &= \sum_{i=1}^{N_s} [\frac{\omega_i}{\rho} - \frac{1}{\rho} (\rho \, Y^i \, V^i)_x] e_i(T) .
\end{aligned}$$

Then, by differentiating (4) and (5), and using the conservation equations, we obtain the following simple first order ordinary differential equation for v

$$\alpha \, v_x = \beta , \tag{6}$$

where,

$$\alpha = 1 + \frac{P}{\rho s_1 T} .$$

$$\beta = \frac{s_3}{s_2} - \frac{s_4 + \frac{q_x}{\rho}}{s_1 \, T} .$$

Our numerical algorithm is designed to solve (2), (3), (6), combined with the algebraic equations (4) and (5). The initial and boundary conditions are problem dependent and will be discussed in Sections 6 and 8.

3 Spatial Discretization

Following the ideas of the method of lines, all spatial-derivative terms are first discretized, converting the PDE problem into a system of ODEs in time. On uniform grids, we use the standard fourth-order central differences on the interior of the grid and fourth-order lopsided schemes near the boundary to discretize the first order spatial derivatives [14]. For diffusion terms like $(a(x)u_x)_x$, we first use fourth-order differences to approximate u_x midway between grid points and then use the same formula to approximate the derivative of au_x (see [7]).

For a nonuniform grid, the mapping method [3] is used to compute the derivatives. Suppose that $x = M(s)$ is a smooth mapping and the grid is given by $x_i = M(s_i)$ with the s_i's given by a uniform grid. Application of the chain rule yields

$$\frac{df}{dx} = (\frac{df}{ds})/(\frac{dx}{ds}),$$

$$\frac{d^2 f}{dx^2} = \Big(\frac{d^2 f}{ds^2}\,\frac{dx}{ds} - \frac{df}{ds}\,\frac{d^2 x}{ds^2}\Big)/\Big(\frac{dx}{ds}\Big)^3 .$$

and then the uniform grid formulas are used to difference the s-derivatives.

4 Temporal Discretization

The spatial discretization creates a system of ordinary differential algebraic equations (DAEs) in time. In this system let y be a vector containing all values of the energy and species at the interior grid points, and let z be a vector containing all other unknown values, such as temperature and velocity at the interior grid points. Then the DAEs have the standard form

$$\begin{cases} y' &= f(y,z) , \\ 0 &= g(y,z) . \end{cases} \tag{7}$$

For our system, due to the introduction of (6), g_z is invertible, so the system has index one [1]. Because

of this, the algebraic system can be inverted and z eliminated from the system, which results in an *extremely stiff* system of ordinary differential equations (ODEs). The explicit elimination of the the z variable will destroy the sparsity structure of the system (7), so this information is only used implicitly in our computations.

Because of the stiffness of the ODEs, we use a semi-implicit or preconditioned time-step algorithm to improve the time-step stability:

$$\begin{aligned}(I - k_j\, J_j)(y^{j+1} - y^j) &= k_j\, f(y^j,\, z^j) \qquad (8)\\ 0 &= g(y^{j+1},\, z^{j+1})\,, \qquad (9)\end{aligned}$$

where k_j is the time step size and J_j is some generally rough approximation to the Jacobian of the ODEs. So each step of the algorithm consists of updating the y values followed by a solve for the z values. This discretization is first-order accurate as long as $(I - k_j\, J_j)$ is invertible and the J_j are bounded, independent of the definition of J_j.

The choice of the preconditioner J_j is based on the Jacobian of f with respect to y. For example, if $f(y, z)$ is simply a fourth order central difference of u_{xx}, then a good choice for J_j is second-order central difference approximation of u_{xx} (see [7] for more details). The advantage using this preconditioner is that the resulting coefficient matrix in (8) is block tridiagonal and thus is cheap to invert.

Since fourth order discretization is used in the spatial variable, we would also like to use higher-order discretization for the time integration. There are two distinct approaches that produce higher order methods based on a lower order method: extrapolation [8] and iterated deferred correction [2]. Here we use the second approach.

Because the algebraic equations in our PDAEs are solvable, for $t \in [a, b]$ we can write our problem in the form

$$\begin{aligned}y'(t) &= F(y(t))\,, \qquad (10)\\ y(a) &= y_0\,.\end{aligned}$$

This ODE can also be written as an integral equation,

$$y(t) = y(a) + \int_a^t F(y(s))\, ds\,, \qquad (11)$$

which is the form used in our derivation of the higher order method.

Suppose the first order approximations at the points $t_1 < t_2 < \cdots < t_m$ in $[a, b]$ are computed. We will apply the idea of deferred correction to the integral equation (11) using Gauss Quadrature on these points. Define $h_j = t_{j+1} - t_j$ and let $\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_m$ be the first-order approximation of the solution at these grid points. For the Lagrange interpolant $\tilde{y}(t)$ of these first order solution values $\tilde{y}_j$ at t_j, define the residue as:

$$\epsilon(t) = y(a) + \int_a^t F(\tilde{y}(s))\, ds - \tilde{y}(t)\,.$$

With this residue, we construct the neighboring problem

$$y(t) = y(a) + \int_a^t F(y(s))\, ds - \epsilon(t).$$

which has $\tilde{y}(t)$ as its exact solution. Although the exact solution is known, we still use the first order method described above to find it numerically. The new approximations $\bar{y}_j$ are then

$$(I - h_j\, J_j)(\bar{y}_{j+1} - \bar{y}_j) = h_j\, F(\bar{y}_j) - (\epsilon_{j+1} - \epsilon_j).$$

We expect that the error in $\bar{y}_j$ approximating $\tilde{y}_j$ is almost the same as the error for $\tilde{y}_j$ approximating the real solution $y(t_j)$. Therefore the corrected approximation

$$y_j = \tilde{y}_j + (\tilde{y}_j - \bar{y}_j)$$

should be more accurate.

The correction can be done several times, resulting in the *Deferred Correction Algorithm*:

- For $j = 0, \cdots, m$, $\bar{y}_j^0 = \tilde{y}_j^0$
- For $k = 0, 1, \cdots, k_{\max}$, $y_0^k = y(a)$

  $$\epsilon_{j+1}^k - \epsilon_j^k = \int_{t_j}^{t_{j+1}} F(\tilde{y}^k(s))\, ds - (\tilde{y}_{j+1}^k - \tilde{y}_j^k)$$

  $$\bar{y}_{j+1}^k = \bar{y}_j^k + (I - h_j\, J_j)^{-1}(h_j\, F(\bar{y}_j^k) - (\epsilon_j^k - \epsilon_j^k))$$

  $$\tilde{y}_{j+1}^{k+1} = \tilde{y}_{j+1}^0 + (\tilde{y}_{j+1}^k - \bar{y}_{j+1}^k)$$

- Extrapolate the $\tilde{y}_j^{k_{max}}$ values to the value $\tilde{y}_b$ at $t = b$.

In fact, the deferred correction algorithm does increase the order of the method, which we will prove elsewhere.

5 Other Issues

We complete our discussion of temporal integration by mentioning two other issues: the spectral integration needed in the computation of the integral appearing in the deferred correction algorithm; and the extrapolation needed when the integration nodes do not contain the end point of the interval. As mentioned above, integration in the Deferred Correction Algorithm is done by Gauss quadrature [6], as suggested in [4].

So let $s_1 < s_2 < \cdots < s_m$ be the usual m Gauss-Legendre nodes on the interval $[-1, 1]$. For a general interval $[a, b]$, the Gauss-Legendre nodes are then

$$t_i = \frac{1}{2}(b-a)s_i + \frac{1}{2}(b+a)\,,$$

and integration is done by changing variables. Let $f_1, f_2, \cdots, f_m$ be the values of f at these nodes. The Lagrange interpolant of these values as a function of s is

$$L^m(f,s) = \sum_{j=1}^{m} c_j(s)\, f_j\,, \quad c_j(s) = \prod_{i \neq j} \frac{s - s_i}{s_j - s_i}\,.$$

Define

$$g_i = \int_{-1}^{s_i} L^m(f,s)\, ds, \qquad i = 1, 2, \cdots, m\,,$$

and then

$$g_i = \sum_{j=1}^{m} \int_{-1}^{s_i} c_j(s)\, ds \cdot f_j\,.$$

If the matrix $S^m \in R^{m \times m}$ is defined by

$$S^m_{ij} = \int_{-1}^{s_i} c_j(s)\, ds, \qquad i, j = 1, 2, \cdots, m$$

then

$$g = S^m f$$

with $f = (f_1, f_2, \cdots, f_m)^T$ and $g = (g_1, g_2, \cdots, g_m)^T$. The matrix S^m is called the spectral integration matrix, and is only dependent on the number of nodes chosen.

In the temporal integration, the whole time interval is divided into subintervals, and the spectral deferred correction is done in each subinterval. Let $[t_0, t_s]$ be such an interval. If t_s is not one of the nodes, then we have not yet defined the value of the high-order approximation here. We will use extrapolation on the high-order solution y_0 and the corrected values $\tilde{y}_j$, $j = 1, \cdots, m$ to define this value, because if we only use the corrected values, the maximum order in time will be reduced. The extrapolation of order m is

$$\tilde{y}(t_s) = \sum_{j=0}^{m} \tilde{c}_j \tilde{y}_j, \quad \tilde{c}_j = \prod_{i \neq j} \frac{1 - s_i}{s_j - s_i}\,,$$

where, $s_0 = -1$, and s_j, $j = 1, \cdots, m$ are Gauss-Legendre nodes in the interval $[-1, 1]$. For $m = 4$, the coefficients are listed in the Table 1 where $\alpha_1 = -\sqrt{(3+\beta)/7}$, $\alpha_2 = -\sqrt{(3-\beta)/7}$, and $\beta = 2\sqrt{6/5}$.

$\tilde{c}_0$	$\tilde{c}_1$	$\tilde{c}_2$	$\tilde{c}_3$	$\tilde{c}_4$
1	$\frac{4+\beta}{2\beta\alpha_1}$	$\frac{-4+\beta}{2\beta\alpha_2}$	$\frac{4-\beta}{2\beta\alpha_2}$	$\frac{-4-\beta}{2\beta\alpha_1}$

Table 1: Coefficients of Extrapolation for $m = 4$

6 Numerical Tests

We tested our implemention of the fourth-order in space and time algorithm on the hydrogen-air fuel system. At the inlet, the gas is assumed to be a stoichiometric mixture at room temperature and atmospheric pressure. The model involves 13 species. The domain is truncated to the interval $[0, 1]$ with artificial boundary condition at $x = 1$. The hydrogen-air mixture is injected with the velocity $v_0 \geq 0$ at the left boundary.

At the cold boundary $x = 0$, Dirichlet boundary conditions are imposed:

$$\begin{aligned} Y^i(0,t) &= Y^i_0, \,, \\ e(0,t) &= e_0\,, \\ v(0,t) &= v_0\,. \end{aligned} \tag{12}$$

For the hot boundary $x = 1$, we use the conditions analyzed in [11] for advection-diffusion equations:

$$\begin{aligned} Y^i_{xx}(1,t) &= 0, \,, \\ e_{xx}(1,t) &= 0\,, \end{aligned} \tag{13}$$

Since the equation for v is first order, we do not need any boundary condition for v at the hot boundary.

The initial conditions at $t = 0$ are given by,

$$\begin{aligned} Y^i(x,0) &= Y_0^i(x), , \\ e(x,0) &= e_0(x) . \end{aligned} \tag{14}$$

In [7], the steady state solution for the hydrogen-air fuel was computed which has a flame front traveling to the left at some speed between 236.86 and 237.90 cm/s. So if we choose the initial data to correspond to this steady state flame and the velocity at the left boundary to be zero ($v_0 = 0$), then we expect a flame that propagates to the left at approximately this velocity.

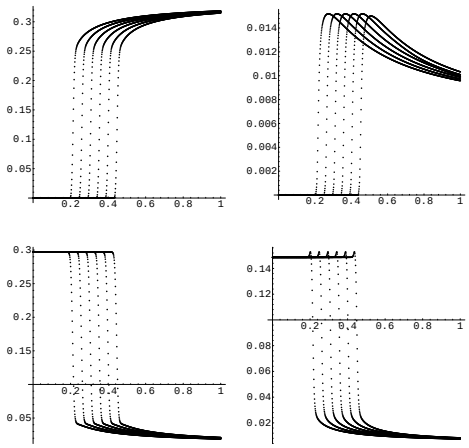
space	$\frac{1}{200}$	$\frac{1}{400}$	$\frac{1}{800}$
time	10^{-7}	0.5×10^{-7}	0.25×10^{-7}

Table 2: Spatial and Temporal Steps

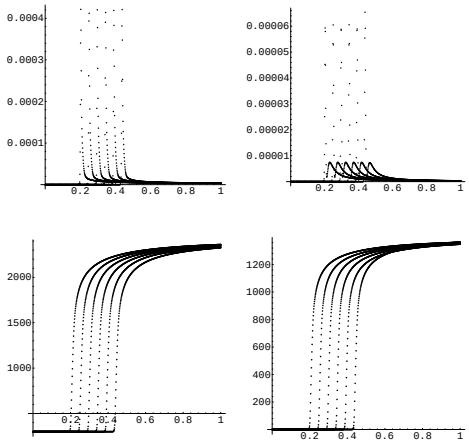
species	error1 h=1/150	error2 h=1/300	order
velocity	0.138	1.73×10^{-4}	9.63
H2O	2.04×10^{-4}	1.36×10^{-5}	3.90
H2	1.74×10^{-4}	1.03×10^{-5}	4.07
O2	2.89×10^{-4}	2.03×10^{-5}	3.83
N2	9.71×10^{-6}	7.98×10^{-7}	3.60
NO2	2.58×10^{-5}	1.06×10^{-6}	4.60

Table 3: Error and Order

First we checked the order of convergence of our code by running it for a small time interval, 2×10^{-5} with uniform (not adapted) grids in space and time (Table 2). We choose base spatial and temporal steps and successively divided this by two and estimated the L_2 norm relative error by computing the difference of the solution on the coarse grids and the finest grid. Table 3 contains the estimates of the errors and order of convergence for the velocity and some of the chemical species. From this table, we conclude that the method is indeed fourth order. (We don't have an explanation for the behavior of the velocity error.)



mole fraction of species
H_2O, OH, H_2, O_2



mole fraction of species
HO_2, H_2O_2, temperature, velocity

Figure 1: Front Traveling Left, 801 uniform points

time	front speed
1×10^{-4}	237.077
2×10^{-4}	237.032
3×10^{-4}	237.032
4×10^{-4}	237.028
5×10^{-4}	237.030
6×10^{-4}	237.031
7×10^{-4}	237.028
8×10^{-4}	237.028
9×10^{-4}	237.029
10^{-3}	237.031

Table 4: Front Speed at Different Times

We used the steady-state solution computed in [7] as initial data, with the front located around $x = 0.4$, and then computed the solutions shown in Figures 1. We only display the some of the more interesting species and the temperature and velocity for a uniform grid with 801 points and at the times 0, 2×10^{-4}, 4×10^{-4}, 6×10^{-4}, 8×10^{-4} and 10^{-3}. The time step size is 10^{-6}. As hoped, these profiles move smoothly to the left. We computed the front speed by estimating where the temperature was $1500°K$ and then computing the velocity of this point. These speeds are listed in Table 4. The computation of the front speed in the steady-state code is an integral part of the algorithm, so we would not expect to get exactly the same speed here. We find the agreement excellent.

# grid points	successful	failed
200	10^{-7}	1.2×10^{-7}
400	5×10^{-6}	6×10^{-6}
600	1×10^{-5}	1.1×10^{-5}
800	9×10^{-6}	10^{-5}
1000	9×10^{-6}	10^{-5}
1200	8×10^{-6}	9×10^{-6}
1400	8×10^{-6}	9×10^{-6}
1600	9×10^{-6}	10^{-5}

Table 5: Stable Time-Step Size

The most critical point for efficiency is not to have a severe restriction on the time step for fine spatial grids. Because we are preconditioning, rather than using a fully implicit method, we don't expect unconditional stability. What we do get is that for a sufficiently fine grid, the stable time step is approximately constant and about 10^{-5} as shown in Table 5. This table is constructed by setting the final time to 10^{-3}, and then adjusting the time step until we find two times, one where the code runs to the final time and one where the code halts on some error condition before the final time is reached.

time	H	O	N
1×10^{-4}	0.29610	0.14903	0.55487
2×10^{-4}	0.29607	0.14903	0.55489
3×10^{-4}	0.29605	0.14903	0.55491
4×10^{-4}	0.29602	0.14904	0.55494
5×10^{-4}	0.29600	0.14904	0.55495
6×10^{-4}	0.29599	0.14904	0.55496
7×10^{-4}	0.29598	0.14905	0.55497
8×10^{-4}	0.29597	0.14905	0.55498
9×10^{-4}	0.29597	0.14905	0.55498
10^{-3}	0.29596	0.14905	0.55499

Table 6: Conservation Test

Finally, we tested the conservation of atoms of H, O, and N. For an 801 point uniform grid with time step size 5×10^{-6}, the total mole fraction of these atoms are listed in the table 6. Again, this shows excellent conservation.

We conclude that the algorithm is indeed fourth-order accurate and, once the front is sufficiently resolved by the grid, that the stability constraint on the time step is constant. The constant time step constraint is ideal for the application of adaptive grids.

7 Adaptive Grid Generation

In the problems that interest us we always expect at least one flame front where the solution of the PDAEs will vary rapidly over short distances. Because, for our algorithm, the stability constraint on the time step is constant for small spatial steps, we can use highly resolved grids without adversely impacting the number of time steps. This is ideal for the use of solution adapted grids. Because our method is higher-order accurate, it is a good idea to use smooth grids. A method of choice is the moving grid adaptive scheme described in [10]. This book uses a variational approach to grid generation, but in one dimension this is equivalent to several other simple methods. Here we use the notion of the equidistribution of a

monitor function, but the variational ideas allow us to extend the grid generation to more dimensions.

In the equidistribution approach one creates a nonnegative monitor function $M(x)$ and then a mesh x_k such that the average of M achieves the same constant value over each subinterval:

$$\int_{x_k}^{x_{k+1}} M(x)\,dx = C, \qquad k = 0, 1, \cdots, N-1\,, \quad (15)$$

where C is a constant. In this and many other approaches it is common to take M a function of the first and second derivatives of the solution of the differential equations being solved, where this function should increase as these derivatives increase.

If u is some component of the solution then we take $M(x) = M(x, \alpha, r)$ where

$$M(x, \alpha, r) = \alpha + \beta M_1 + (1-\beta) M_2\,, \qquad (16)$$

with

$$\begin{aligned} M_1 &= \Big(\frac{|u_x|}{\max|u_x|}\Big)^{\frac{1}{2}r_1}, \\ M_2 &= \Big(\frac{|u_{xx}|}{\max|u_{xx}|}\Big)^{\frac{1}{4}r_2}, \end{aligned}$$

where α is a very small number and to keep M positive, β is chosen between 0 and 1. The nonnegative parameters r_1 and r_2 are determined by numerical experimentation. Next we define the map p by

$$p(x,\, \alpha) = \frac{\int_0^x M(x,\alpha)\,dx}{\int_0^1 M(x,\alpha)\,dx}\,.$$

One easily checks that the mesh $0 = x_0 < x_1 < \cdots < x_N = 1$ is equidistributed on $[0,1]$ with respect to M if and only if

$$p(x_k) - p(x_{k-1}) = 1/N\,, \quad k = 1, 2, \cdots, N\,. \quad (17)$$

In practice, M is only available at the old grid points, so linear interpolation is used to compute intermediate values. To produce the new grid points we use Newton iteration to solve (17) for x_{k+1} given x_k. That is, we use the iteration

$$M(w^m)\,(w^{m+1} - w^m) = \Delta_m\,,$$

with

$$\Delta_m = \int_{x_k}^{w^m} M(x)\,dx - \frac{1}{N}\int_0^1 M(x)\,dx\,,$$

and then let $x_{k+1} = w^\infty$. In practice only a few iterations are needed to meet our error tolerance.

A potential problem in the above grid generation is that the grid may not vary smoothly. Typically, the ratio of consecutive grid intervals may differ by several orders of magnitude. Such grids adversely affect the time-integration algorithm and the accuracy of the solution. To control this, let $h_j = x_j - x_{j-1}$ for $j = 1, 2, \cdots, N$, and then make the restriction

$$\frac{1}{C} \le \frac{h_j}{h_{j-1}} \le C, \qquad j = 2, 3, \cdots, N\,. \qquad (18)$$

The constant C is taken as 2.5 in our algorithm. If a grid is produced that violates (18), we throw it away and increase the parameter α in the expression for M (16). This is repeated until (18) is satisfied. Now suppose the whole time interval has been divide into subintervals $[t_j, t_{j+1}]$, $j = 0, 1, \cdots$, and a positive integer l is chosen, the *static adaptive grid algorithm* is:

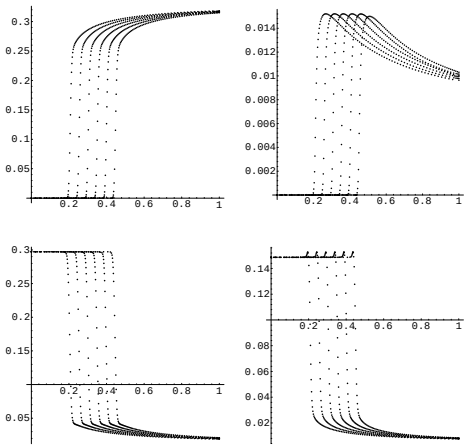
For $j = 0, 1, \cdots$,

- Do the spectral deferred correction algorithm in $[t_j, t_{j+1}]$ using the current grid;
- if j is a multiple of l, define M as in (16) and find proper parameters to get a new grid; then interpolate the solution to the new grid points.

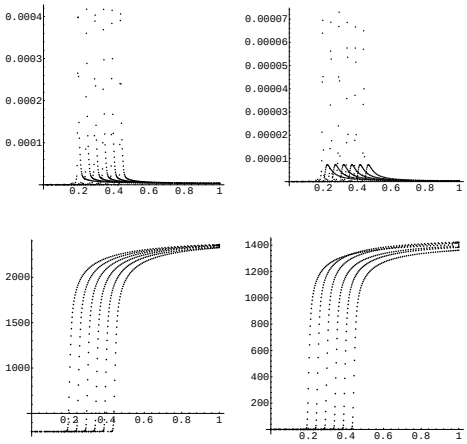
time	front speed
1×10^{-4}	237.55
2×10^{-4}	237.27
3×10^{-4}	237.70
4×10^{-4}	237.81
5×10^{-4}	237.68
6×10^{-4}	237.99
7×10^{-4}	237.52
8×10^{-4}	237.85
9×10^{-4}	237.64
1×10^{-3}	237.76

Table 7: Front Speeds on an Adapted Grid

From our numerical experiments we found that every parameter in the monitor function plays an important role in the grid adaption. In most of the literature, r_1 was taken to be 2, which corresponds to the arc length, and r_2 was also taken to be 2, which corresponds to curvature. But in our computations, those choices lead to failure in many cases. On the



mole fraction of species
H_2O, OH, H_2, O_2



mole fraction of species
HO_2, H_2O_2, temperature, velocity

Figure 2: Front Traveling Left, 101 adaptive points

other hand, for the choices $r_1 = r_2 = 1$, as well as other suitable choices of these parameters, the computations complete successfully. For good grids, we take the parameter β near 0. If α is too small, the grid produced does not satisfy the smoothness restriction (18). For α too large, the grid produced does not cluster points near the steep front. We haven't found a value of α that is successful for all computations, but our experience is adequate to give a good choice for any specific number of grid points.

To generate the solutions shown in Figure 2, we used 101 points in a grid adapted using $r_1 = r_2 = 1$, $\beta = 0$ and $\alpha = 5 \times 10^{-4}$, and the temperature as the monitor function. The initial data is the steady-state solution used in Section 6, and the time step size was taken to be 10^{-6}. The solution is plotted at the times 0, 2×10^{-4}, 4×10^{-4}, 6×10^{-4}, 8×10^{-4} and 10^{-3}. Table 7 records the estimated front speeds. This is in excellent agreement with the results in Table 4 for a uniform grid of 801 points.

Note that the fourth order method in time was used. That is, we did the spectral deferred correction three times. A second order method in time was also tried with several choices of the parameters in the monitor function. None of them could run to the designed time interval using 101 grid points and the same time step size (10^{-6}).

8 Quenching Distance

As a final experiment, we tried to compute the quenching distance of the flame, that is, the minimum distance of the flame front from the cold wall. It is well-known that this distance is not zero, meaning the flame cannot reach the cold wall. To simulate this phenomena, we set Neumann conditions on all species at $x = 0$, keeping the temperature at 300^oK. In each time step, the density of the gas and the energy are modified at the cold boundary by the equations (4) and (5). v_0 is still set to be 0. The boundary conditions (13) are still used at the hot boundary. The steady fronts described in Section 6 are used as initial conditions with the front nearby the boundary.

In our computations, there were 601 points in the space grid. The adaptive grid technique was used before the front approached the cold boundary. After a sufficient number of grid points clustered at the boundary, the grid was left unchanged. The time step size was chosen as 2×10^{-8} when the front was very close to the cold boundary. Otherwise it was set

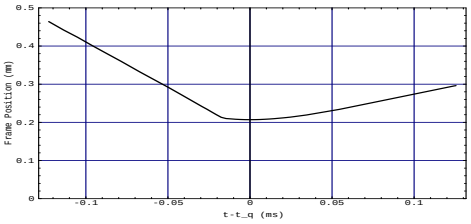


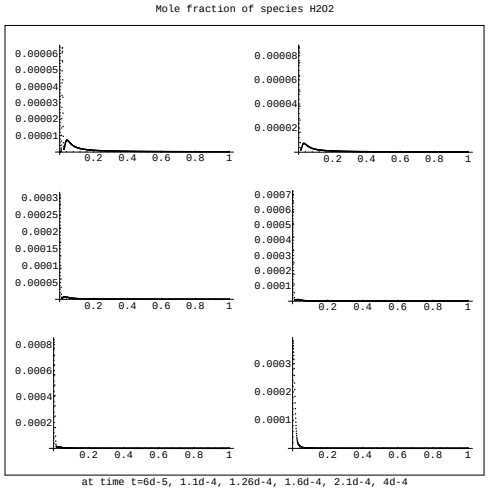
Figure 3: Distance of the flame front to the wall

to be 2×10^{-7}, and even 10^{-6} when the front is far away. Figure 3 shows the time behavior of the flame position (t_q is the time when the flame front is closest to the wall). The quenching distance we got is 0.207 mm. In Figure 4, 5 and 6, some species, as well as the temperature and velocity are plotted before and after the flame reaches the quenching distance. We could see the detail how the cold boundary affects the fronts. It is particularly interesting to note that essentially all the hydrogen is burned, while a layer of oxygen remains.

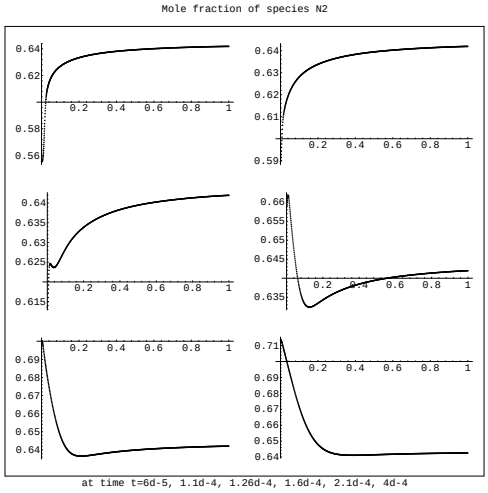
We emphasize that these are preliminary computations. We expect more efficient (and longer) runs by choosing appropriate parameters in the adaptive grid techniques and using far fewer grid points.

9 Summary

We have proposed a high order method solving a time dependent, one dimensional problem system of equations including complex models of reacting gases. Numerous experiments show that the method is accurate, robust and efficient. Typically, we simulate the propagation of the flame using uniform or adaptive grids. The correct front speed was obtained in either case. As a first application, we used our code to find the quenching distance of a flame impinging onto a cold wall. More interesting applications will be considered in the future.

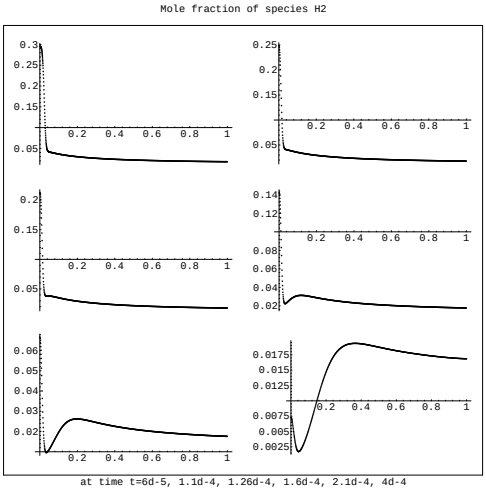


Mole Fraction of Species H_2O_2

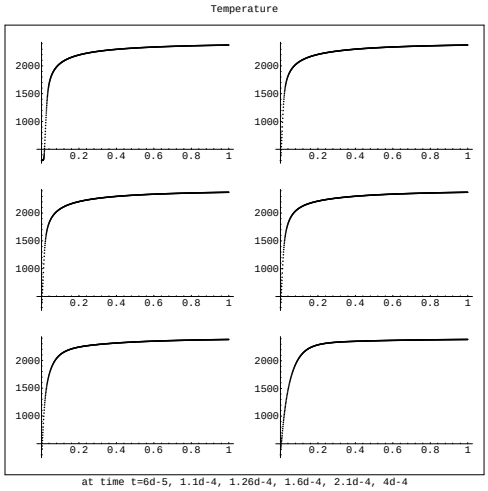


Mole Fraction of Species N_2

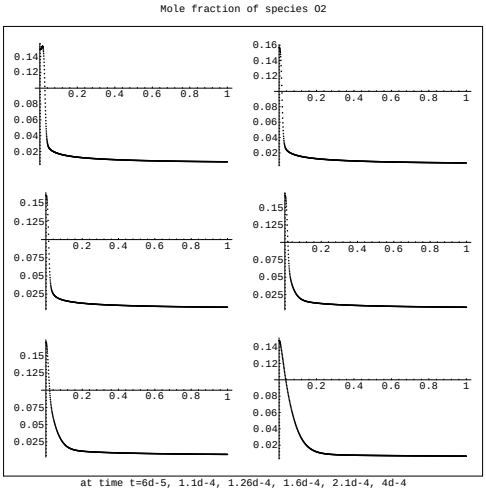
Figure 4: Front near the boundary



Mole Fraction of Species H_2

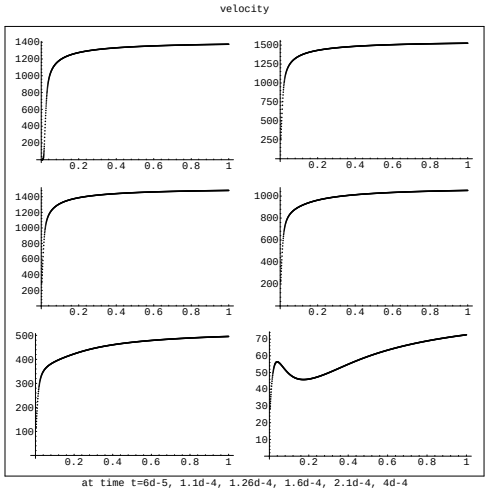


Temperature



Mole Fraction of Species O_2

Figure 5: Front near the boundary



Velocity

Figure 6: Front near the boundary

References

[1] U.M. Ascher & L.R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM Philadelphia, 1998

[2] K.Bohmer & H.J.Stetter, eds., *Defect Correction Methods, Theory and Applications*, Springer-Verlag, Wien-New York, 1987

[3] J.E. Castillo, J.M. Hyman, M.J. Shashkov & S. Steinberg, High-order mimetic finite difference methods on nonuniform grids, Proceedings of the Third International Conference on Spectral and High Order methods, Houston Journal of Mathematics, Houston, A.V. Ilin and L.R. Scottt (eds), 1996, 347-361. Also, Los Alamos National Laboratory report LA-UR-95-583.

[4] A. Dutt, L. Greengard & V. Rokhlin Spectral Deferred Correction Methods for Ordinary Differential Equations, Research Report YALEU/DCS/RR-1143, January 30, 1998

[5] S. Gordon & B.J. McBride, Computer program for calculation of complex chemical equilibrium composition and applications. Part I - Analysis, NASA RP-1311, NASA Lewis Research Center, Cleveland, OH, (1994)

[6] D. Gottlieb & S.A. Orszag, *Numerical Analysis of Spectral Methods*, SIAM, Philadelphia, 1977.

[7] T. Hagstrom, K. Radhakrishnan & R. Zhou Computation of steady and unsteady laminar flames: theory, 34th AIAA/ASME/SAE/ASEE Joint propulsion conference & Exhibit, July 13-15,1998, Cleveland, OH

[8] E. Hairer & G. Wanner, *Solving ordinary differential equations II, stiff and differential-algebraic problems*, Springer-Verlag, New York, 1996

[9] R.J. Kee, J.F. Grcar, M.D. Smooke & J.A. Miller, A Fortran program for modelling steady laminar one-dimensional premixed flames, Sandia Report SAND85-8240, Sandia National Laboratory, Livermore, CA, (1985)

[10] P.Knupp & S.Steinberg, *Fundamentals of Grid Generation*, CRC Press, Boca Raton, 1993

[11] J.P. Loheac, An artificial boundary condition for an advection-diffusion equation, Math. Meth. Appl. Sci., 14, (1991), 155-175

[12] K. Radhakrishnan LSENS - A general chemical kinetics and sensitivity analysis code for homogeneous gas-phrase kinetics. I. Theory and numerical solution procedures, NASA RP-1328, NASA Lewis Research Center, Cleveland, OH, (1994)

[13] J. Stoer & R. Bulirsch, *Introduction to Numerical Analysis*, Springer, 1992.

[14] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks Cole, Pacific Grove, CA, 1989

[15] F.A. Williams *Combustion Theory*, second edition, The Benjamin/Cummings Publishing Company, Inc. CA, 1985