NASA/CR—1999-209305

AIAA 98–3246
ICOMP-99–07

Computation of Steady and Unsteady Laminar Flames: Theory

Thomas Hagstrom
University of New Mexico, Albuquerque, New Mexico

Krishnan Radhakrishnan
Institute for Computational Mechanics in Propulsion, Cleveland, Ohio

Ruhai Zhou
University of New Mexico, Albuquerque, New Mexico

Prepared for the
34th Joint Propulsion Conference and Exhibit
cosponsored by AIAA, ASME, SAE, and ASEE
Cleveland, Ohio, July 13–15, 1998

Prepared under Cooperative Agreement NCC3–622

National Aeronautics and
Space Administration

Glenn Research Center

October 1999

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A03

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A03

COMPUTATION OF STEADY AND UNSTEADY LAMINAR FLAMES: THEORY

Thomas Hagstrom[*], Krishnan Radhakrishnan[†] and Ruhai Zhou[‡]

Abstract

In this paper we describe the numerical analysis underlying our efforts to develop an accurate and reliable code for simulating flame propagation using complex physical and chemical models. We discuss our spatial and temporal discretization schemes, which in our current implementations range in order from two to six. In space we use staggered meshes to define discrete divergence and gradient operators, allowing us to approximate complex diffusion operators while maintaining ellipticity. Our temporal discretization is based on the use of preconditioning to produce a highly efficient linearly implicit method with good stability properties. High order for time accurate simulations is obtained through the use of extrapolation or deferred correction procedures. We also discuss our techniques for computing stationary flames. The primary issue here is the automatic generation of initial approximations for the application of Newton's method. We use a novel time-stepping procedure, which allows the dynamic updating of the flame speed and forces the flame front towards a specified location. Numerical experiments are presented, primarily for the stationary flame problem. These illustrate the reliability of our techniques, and the dependence of the results on various code parameters.

1 Introduction

Comprehensive mathematical models of reacting gases have been known for some time. (See, e.g., [18].) Nonetheless, due to the complexity of the models, it is necessary to use numerical methods to extract detailed predictions and evaluate their ability to faithfully reproduce experimental results. In this note we review our ongoing efforts to design and implement reliable numerical methods for the solution of these complex models. Of course, we are not the first researchers to attempt to solve these equations. Of particular note is the work of groups at the Ballistic Research Laboratory [5], Sandia National Laboratory [11] and the Naval Research Laboratory [10]. Nonetheless, some of our approaches differ from those pursued by other groups and seem worthy of additional investigation. In the design process, we have been motivated by potential efficiency in multidimensional simulations. In this work, however, we will restrict ourselves to examples in one space dimension.

2 Models

Our starting point is the equations for conservation of mass, momentum, energy and species:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0, \qquad (2.0.1)$$

$$\frac{\partial u}{\partial t} + (u \cdot \nabla) u + \frac{1}{\rho} \nabla \phi = \frac{1}{\rho} \nabla \cdot J_{\nu}, \qquad (2.0.2)$$

$$\frac{\partial e}{\partial t} + (u \cdot \nabla) e = -\frac{1}{\rho} \nabla \cdot q - \frac{P}{\rho} \nabla \cdot u, \qquad (2.0.3)$$

$$\frac{\partial Y_i}{\partial t} + (u \cdot \nabla) Y_i = -\frac{1}{\rho} \nabla \cdot (\rho Y_i V_i) + r_i,. \qquad (2.0.4)$$

Here, ρ is the density of the gas mixture, u is the velocity vector, e is the internal energy and Y_i, $i = 1, \ldots, n_s$, is the mass fraction of the ith species.

We consider a zero Mach number model, akin to the incompressible equations for fluid flow. Precisely, our gas law equation of state takes the form:

$$\rho R T \sum_{i=1}^{n_s} \frac{Y_i}{M_i} = P(t), \quad e = \sum_{i=1}^{n_s} \frac{e_i(T) Y_i}{M_i}, \qquad (2.0.5)$$

[*] Department of Mathematics and Statistics, The University of New Mexico, Albuquerque, NM 87131. Work supported in part by NASA Grant NRA-96-LeRC-2, NSF Grant DMS-9600146 and by ICOMP, NASA Glenn Research Center. Partially carried out while the author was in residence at the Courant Institute of Mathematical Sciences, New York University.

[†] ICOMP, NASA Glenn Research Center, Cleveland, OH 44135.

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

$$e_i(T) = h_i(T) - RT. \qquad (2.0.6)$$

That is, we assume that the pressure, to leading order, is spatially constant. The quantity ϕ is the spatially varying pressure perturbation. It is only retained in the momentum equation. In a closed domain, an additional equation is needed to specify the temporal evolution of P. In an open domain, however, we may assume that P is constant. Detailed mathematical analyses of these equations and their derivation are given in [13, 8].

There are both advantages and disadvantages to using the zero Mach number system. The primary advantage is in the design of time stepping strategies, as large imaginary eigenvalues of the spatial operators are suppressed. Disadvantages are its restricted applicability to problems where the Mach number remains small and the necessity to solve additional equations to simulate sound production.

To complete the problem specification, we must describe the computation of the reaction rates, r_i, the diffusion velocities, V_i, the heat flux, q, the enthalpies, h_i, and the viscosity coefficients. Our programs are designed to accommodate any reasonable specifications of these quantities without requiring basic changes in the algorithm. In this work the species production rates were computed using the procedures (and rate coefficient expressions) built into the NASA Lewis kinetics code LSENS [16]. The NASA Lewis polynomial expressions [9] were used to calculate the thermodynamic properties, and the routines for the transport properties were adapted from the Sandia 1-D flame code [11]. The species diffusion velocities were obtained by solving the multicomponent diffusion equation, which is a matrix equation—see equation E18 in [18].

3 Plane Flames

A fundamental problem of computational combustion is to determine flame speeds and species profiles, or to establish the nonexistence of the flame, given the mixture composition, temperature and pressure. Mathematically, the plane flame is a traveling wave solution of the governing equations. That is, if x is the coordinate normal to the flame front and v is the flame speed we seek a solution of (2.0.1)-(2.0.4) which is a function of $z = x + vt$. Equivalently, we may make a Galilean transformation and seek a steady solution depending on x alone with a nonvanishing x-velocity, $u = v$, at $x = -\infty$.

Generally, boundary conditions for traveling wave problems involve specifying limiting rest states at $\pm\infty$. This description does not strictly apply to the plane flame problem, as there is no rest state associated with the unburnt gas. (This is sometimes called the cold boundary difficulty.) Here we simply impose an unburnt state, that is fix the unburnt fuel mixture, $Y_{i,0}$, temperature, T_0, and pressure, P_0. This raises doubts concerning the strict existence of the unbounded domain solution. However, so long as the unburnt state is an approximate equilibrium (that is the reaction rates are sufficiently small), we expect that our truncated problems do possess solutions which correctly model the physics.

True equilibria, which will define the burnt state, are defined by the n_s equations:

$$r_i(Y_{1,b}, \ldots, Y_{n_s,b}, T_b) = 0. \qquad (3.0.7)$$

As the number of unknowns exceeds the number of equations by one, it is reasonable to expect that a one-parameter family of equilibria exist. To uniquely determine the burnt state, we note that under the zero Mach number assumption, an additional conservation law for the enthalpy can be derived:

$$\frac{\partial}{\partial x}\left(q + \sum_{i=1}^{n_s} \rho(u + v + V_i)Y_i h_i\right) = 0. \qquad (3.0.8)$$

(Here h_i is the enthalpy of species i.) Integrating this equation from the unburnt to the burnt state, imposing the vanishing of the gradients and, hence, of q and V_i in these states, and taking into account mass conservation, we arrive at the additional relation:

$$\sum_{i=1}^{n_s} Y_{i,0} h_{i,0} = \sum_{i=1}^{ns} Y_{i,b} h_{i,b}. \qquad (3.0.9)$$

Supplementing (3.0.7) by (3.0.9), the burnt state is finally determined. In practice, the chemical equilibrium state is not computed using equations (3.0.7) and (3.0.9). It is instead obtained by minimizing the Gibbs function—for details see [9, 16].

An additional property of traveling wave solutions is their translation invariance. That is, given any solution a new solution can be obtained by adding any constant to the independent variable, effectively translating the profile. Mathematically, this implies that the space derivative of the solution is a nullfunction of the linearized equations. To fix our profile in space we impose an additional phase condition. Many such conditions are possible. (For an interesting discussion of the discretization theory of traveling wave problems see Beyn [4].) What is essential is that the additional condition be nonvanishing when applied to the derivative of the solution. We choose a temperature normalization:

$$T(x_h) = T_h, \quad T_0 < T_h < T_b. \qquad (3.0.10)$$

For this to be effective we must have:

$$\frac{dT}{dx}(x_h) > 0. \tag{3.0.11}$$

So long as $T = T_h$ occurs within the flame front, (3.0.11) is expected to be valid, and has been in all our experiments. Later we will discuss a new method for dealing with the phase condition in the context of a time stepping procedure for computing the flame.

4 Spatial Discretization and Adaptivity

We approximate spatial derivatives using second to sixth order central differences, switching to lopsided schemes where necessary near the boundaries. We expect that higher order methods will prove to be more efficient, particularly when the physical models are complex. In these cases, the overhead associated with the use of higher order differences is small compared with the cost of evaluating various physical quantities at each point. Hence, any reduction in the number of mesh points results in a reduction of computing time.

Two distinct spatial difference formulas are employed. The first, which is denoted D_g and used to approximate first order terms in the equation, takes a function defined on a uniform grid and produces approximate derivatives on that grid. The second operator, D_h, approximates derivatives midway between gridpoints. These operators are composed to approximate second order derivatives on the grid. For example, we approximate the species conservation equations (2.0.4) by:

$$uD_gY_i = -\frac{1}{\rho}D_h\left((I_h\rho)(I_hY_i)V_i\right) + r_i, \tag{4.0.12}$$

where I_h is an interpolation operator of appropriate order producing function values midway between gridpoints from values on the grid. The diffusion velocities are computed, again midway between gridpoints, by:

$$V_i = V_i(D_hY_j, D_hT, I_hY_j, I_hT). \tag{4.0.13}$$

The reason for staggering the grids in this way is to guarantee the ellipticity of the resulting discrete diffusion operators. The only grid functions annihilated by D_h are constant, whereas D_g also annihilates the function $(-1)^j$ where j is a grid index. Therefore, the Fourier transforms of second order operators formed by composing D_h with itself are always large for large wavenumbers while those obtained by composing D_g with itself decrease to zero as the wavenumber approaches $\pi/\Delta x$, that is as we approach the largest wavenumber representable on the grid.

For the one-dimensional problems considered here, the essential length scales are the computational domain length, L, which we cannot make too small without using far more sophisticated boundary treatments, and the width of the flame zone. In practice the ratio of these is sufficiently small to merit an adaptive mesh strategy. Currently, we use a parametrized family of coordinate maps. This strategy has been successfully employed and analyzed by Bayliss and coworkers [1, 2] in the context of spectral simulations of simpler combustion models. The maps we use are slightly different. Precisely, we write:

$$x = x_l + \frac{1}{2}\gamma y\left(\epsilon + \frac{(1-\epsilon)(y/\beta)^p}{1+(y/\beta)^p}\right), \tag{4.0.14}$$

and employ a uniform grid in y. For $\epsilon < 1$ there is a clustering of gridpoints near $x = x_l$. Using the temperature as our monitor function, we choose x_l to coincide with the location of the maximum derivative of T and choose ϵ according to the ratio:

$$\epsilon = \frac{T_b - T_0}{L(T_x)_{\max}}. \tag{4.0.15}$$

Throughout our simulations we have chosen $p = 8$. The parameters γ and β are chosen to scale the length of the y-interval and to guarantee that sufficiently many points are in the refinement layer. Our exact formulas are:

$$\beta = (1+3\epsilon)^{-1}, \quad \gamma = \frac{1+3\epsilon}{4\epsilon}\cdot L. \tag{4.0.16}$$

A defect of the very simple mapping described above is its inability to handle multiple fronts. An alternative method we have used which is capable of handling multiple fronts is given by the mapping:

$$y = A\int_0^x \left(1 + \xi|T_x| + (1-\xi)|T_{xx}|^{1/2}\right)^{1/2} dx. \tag{4.0.17}$$

where A is a normalization parameter and $0 \leq \xi \leq 1$.

In all cases we use static rezoning, that is update the mesh after some fixed number of time steps and interpolate the solution onto the new mesh. Each of these methods would require extensive modification for use in multiple dimensions. Therefore, a more flexible approach such as the AMR techniques of Berger and coworkers (e.g. [3]) will be considered.

5 Temporal Discretization

It is well known that the equations of reacting flows may be stiff when realistic reaction mechanisms are included. Therefore, if an explicit temporal discretization were employed, the time step could be severely restricted by stability requirements. Fully implicit discretizations, on the other hand, require the solution of a discrete approximation to a nonlinear elliptic system at each time step. Solving such a problem is potentially quite expensive in multiple dimensions. Here we take an intermediate approach, based on the concept of preconditioning.

Consider, for simplicity, a system of ordinary differential equations:

$$\frac{du}{dt} = F(u). \qquad (5.0.18)$$

Introduce a preconditioning matrix, $G(u)$. A first order approximation to (5.0.18) is given by:

$$(I - \Delta t G(u(t)))w = \Delta t F(u(t)), \qquad (5.0.19)$$

$$u(t + \Delta t) = u(t) + w. \qquad (5.0.20)$$

For reasons of efficiency, we want to choose G so that (5.0.19) may be easily solved. To study the stability of the approximation, we consider the special case $F = Ju$. Then we have:

$$u(t+\Delta t) = (I-\Delta tG)^{-1}(I+\Delta t(J-G))u(t) \quad (5.0.21)$$

so that the method is stable if and only if for some subordinate matrix norm:

$$\|(I - \Delta tG)^{-1}(I + \Delta t(J - G))\| \leq 1. \qquad (5.0.22)$$

As a simple example, suppose J is a discrete approximation to a variable coefficient multidimensional diffusion operator:

$$J = \sum_j D_{h,j}(\sigma_j(x)D_{h,j}), \qquad (5.0.23)$$

where D_h are as discussed above. Then it is possible to take:

$$(I - \Delta tG) = \prod_j(I - \Delta t\bar{\sigma}_j D_{+,j}D_{-,j}) \qquad (5.0.24)$$

for suitably chosen $\bar{\sigma}_j$. Then we only need to solve small tridiagonal systems, no matter what stencil width our actual difference approximations employ.

The time stepping scheme, as outlined above, is only first order. This is sufficient for the computation of stationary flames, where time-stepping is simply used to generate an initial approximation for Newton's method. For time accurate simulations, on the other hand, it is more efficient to use a higher order method. There are two distinct approaches we have considered for automatically producing higher order time advancement procedures based on the preconditioning strategy: extrapolation, as in the ode solver SEULIM (e.g. [6]), and iterated deferred correction, as discussed in [7]. As the second approach is less familiar, and is the one employed in the example given below, we describe it briefly.

Suppose $u(t_n)$ is our approximate solution at $t = t_n$ and that we wish to compute $u(t_{n+1})$. Recall the formula obtained by integrating (5.0.18) from t_n to an arbitrary time, t:

$$u(t) = u(t_n) + \int_{t_n}^{t} F(u(s))ds. \qquad (5.0.25)$$

Choose nodes (typically of Gauss, Radau or Lobatto type),

$$t_n \leq s_1 < s_2 < \ldots < s_m \leq t_{n+1}. \qquad (5.0.26)$$

Let the pth approximate solution on the nodes be given by:

$$v_j^p \approx u(s_j), \qquad (5.0.27)$$

where the inital approximation is computed by (5.0.19),(5.0.20) with Δt replaced by $s_{j+1} - s_j$. To compute corrections, define the interpolant, $Q^p(s)$, to $(s_j, F(v_j^p))$ and the residual:

$$\epsilon_j^p = u(t_n) + \int_{t_n}^{s_j} Q^p(s)ds - v_j^p. \qquad (5.0.28)$$

Then solve:

$$\begin{aligned} (I - k_jG(v_j^0))(\delta_{j+1}^p - \delta_j^p) &= \quad (5.0.29) \\ k_j(F(v_j^{p+1}) - F(v_j^p)) + \epsilon_{j+1}^p - \epsilon_j^p, & \end{aligned}$$

$$k_j = s_{j+1} - s_j, \qquad (5.0.30)$$

$$v_j^{p+1} = v_j^p + \delta_j^p. \qquad (5.0.31)$$

The overall method order increases by one at each iteration, up to a maximum determined by the integration scheme. Assuming this order is not less than l, an lth order approximation to $u(t_{n+1})$ is given by:

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} Q^{l-1}(s)ds. \qquad (5.0.32)$$

Note that our semidiscretized system is in fact a differential-algebraic rather than a differential equation. Formally, the index of this system is three. In particular, no equation for the the time derivative of the pressure variation, ϕ, is given. To obtain such an equation, one must differentiate the constraint, namely the equation of state (2.0.5), three times. (The first differentiation produces ρ_t; substituting for ρ_t using (2.0.1), the second differentiation produces u_t; finally, substituting for u_t using (2.0.2) yields an equation for ϕ which can be differentiated and solved for ϕ_t.) In one space dimension, however, (2.0.2) and, hence, ϕ may be eliminated. Then we have no equation for u_t, and the index, following the same reasoning as above, is reduced to two.

In the current code, we only apply our time-stepping procedure to the spatially discretized approximations to (2.0.3) and (2.0.4). The density is computed using (2.0.5) and the velocity is updated using (2.0.1). For stationary flames this reduces to:

$$u = \frac{\rho_0}{\rho} \cdot v, \tag{5.0.33}$$

where v is the velocity at $x = -\infty$. For time accurate approximations we found it difficult to use (2.0.1) directly. Instead, we replace ρ_t by first expressing it in terms of e_t and $Y_{i,t}$ and then using (2.0.3) and (2.0.4) to eliminate all time derivatives. This results in a somewhat complicated expression, which is nevertheless easily integrated for u.

In our code, G is chosen by replacing the heat conduction and species diffusion terms in equations (2.0.3) and (2.0.4) by simplified physical approximations and by using second order differences. It also includes the exact Jacobian of the reaction rates.

Finally, we note that the time steps are automatically adjusted, within preset upper and lower bounds, according to the relative change in the solution.

6 Stationary Flames

As discussed in Section 3, the plane flame problem is to compute stationary solutions to equations (2.0.1)-(2.0.4) along with the normalization condition (3.0.10) and the boundary conditions:

$$(e, Y_i) \to (e_0, Y_{i,0}), \quad x \to -\infty, \tag{6.0.34}$$

$$(e, Y_i) \to (e_b, Y_{i,b}), \quad x \to \infty, \tag{6.0.35}$$

$$u \to v, \quad x \to -\infty. \tag{6.0.36}$$

Recall that the state $(e_0, Y_{i,0})$ is specified, the state $(e_b, Y_{i,b})$ is computed given $(e_0, Y_{i,0})$ by minimizing the Gibbs function, and the flame speed, v, is to be detrmined as part of the calculation.

A direct approach to the computation is to truncate the infinite domain by a finite domain, replacing (6.0.34)-(6.0.36) by approximate boundary conditions at the artificial boundaries, then to approximate the spatial derivatives by differences on a (generally nonuniform) mesh, and finally to use some variant of Newton's method to solve the resulting algebraic system. The problem here is in the last step, as the convergence of Newton's method is only guaranteed for a sufficiently accurate initial guess, which seems difficult to automatically generate for this problem. A more reliable alternative is to use a time-stepping procedure in place of Newton iterations. In this approach one makes use of the apparently large dynamical basin of attraction of the plane flame solution. Disadvantages associated with standard time-stepping procedures include slow convergence to steady state, an inability to fix the flame front, and difficulties in determining a precise speed.

In this work we use a hybrid approach. We begin with a novel time-stepping procedure, which allows us to dynamically compute a flame speed and to modify the dynamics to push the front towards its normalized location. Similar ideas have been proposed by Sermange [17], but ours differ in the details. Second, either after the time derivatives become sufficiently small or after a maximum number of time steps are carried out, we switch to a damped Newton iterator to achieve final convergence.

The time-stepping procedure is most easily explained in a general setting. Consider the nonlinear, algebraic eigenvalue problem:

$$F(w) - vSw = 0. \tag{6.0.37}$$

Here v is the eigenvalue and S is a square matrix. Discrete approximations to the plane flame problem take this form when the velocity, u, is replaced using (5.0.33) and ρ is eliminated using (2.0.5). Equation (6.0.37) then represents the discrete approximation to (2.0.3)-(2.0.4) multiplied by ρ. The matrix S is the discrete derivative operator.

As mentioned above, solutions to the plane flame problem are only unique up to translation. This implies that the equations linearized about a plane flame solution have a nullfunction which is the derivative of the solution itself. Moreover, we determine a unique solution by imposing an additional normalization condition (3.0.10). We represent our discrete approximation to (3.0.10) by

$$N(w) = 0, \tag{6.0.38}$$

where N is scalar valued. Corresponding to the translation invariance, we assume that if w^* is the solution of (6.0.37)-(6.0.38), then the matrix:

$$A \equiv DF(w^*) - vS, \tag{6.0.39}$$

has 0 as a simple eigenvalue with normalized eigenvector $q_0 = Sw^*/||Sw^*||$. (This assumption is generally false after domain truncation. However, dropping it complicates the analysis beyond the scope of this note.) We further assume that w^* is dynamically stable, that is that all other eigenvalues, λ_j, of A, have negative real part. Moreover, we impose the nondegeneracy condition:

$$DN(w^*)Sw^* > 0. \qquad (6.0.40)$$

Note that if the left-hand side of (6.0.40) were zero, the normalization would fail to guarantee a unique solution. In our case, (6.0.40) is equivalent to (3.0.11).

Our time-stepping procedure is to replace (6.0.37) by the dynamical problem:

$$w_t = G(w), \qquad (6.0.41)$$

with G defined by:

$$G(w) = F(w) - (\Phi(w) + \mu N(w))Sw, \qquad (6.0.42)$$

where Φ is a scalar function and $\mu > 0$ is a damping parameter. In order for w^* be a rest point of the new system, it is necessary to guarantee that:

$$\Phi(w^*) = v. \qquad (6.0.43)$$

This we accomplish by choosing:

$$\Phi(w) = \frac{(Sw)^T RF(w)}{(Sw)^T R(Sw)}, \qquad (6.0.44)$$

for some symmetric semidefinite matrix R. In our current code, R is simply the projection onto the internal energy.

The effectiveness of our approach depends on the spectrum of $DG(w^*)$, which must lie in the left half-plane. By direct computation we find:

$$\begin{aligned} DG(w^*) = & \left(I - \tfrac{(Sw^*)(Sw^*)^T R}{(Sw^*)^T R(Sw^*)}\right) A \qquad (6.0.45) \\ & -\mu(Sw^*)DN(w^*). \end{aligned}$$

Let Q be an orthogonal matrix whose first column is q_0. Then we have:

$$Q^T AQ = \begin{pmatrix} 0 & \gamma^T \\ 0 & A^{(1)} \end{pmatrix}, \qquad (6.0.46)$$

and

$$Q^T DG(w^*)Q = \begin{pmatrix} \lambda_0 & \tilde{\gamma}^T \\ 0 & A^{(1)} \end{pmatrix}, \qquad (6.0.47)$$

$$\lambda_0 = -\mu DN(w^*)Sw^* < 0. \qquad (6.0.48)$$

Therefore, the spectrum of $DG(w^*)$ is given by the eigenvalues of $A^{(1)}$, which by (6.0.46) are also the nonzero eigenvalues of A, and λ_0. Hence, the local dynamic stabilty of w^* is assured.

Finally, we consider the boundary conditions imposed at the artificial boundaries introduced to truncate the infinite domain. At the cold boundary we simply impose the Dirichlet conditions:

$$(e, Y_i) = (e_0, Y_{i,0}). \qquad (6.0.49)$$

Our justification for this is the rapid exponential decay to this state which we observe in the computations. At the hot boundary, where the observed decay rates are much slower, we impose the conditions analyzed by Lohéac [12] for advection-diffusion equations:

$$e_{xx} = Y_{i,xx} = 0. \qquad (6.0.50)$$

Although our experiments indicate that these conditions do provide sufficient accuracy, more sophisticated conditions could be used and might lead to improvements. See [4] for a detailed discussion.

7 Results

In this section we present the results of some numerical experiments with our code. The vast majority of these involve computations of a stationary, stoichiometric hydrogen-air flame at a temperature of 300 degrees Kelvin and atmospheric pressure. Our mechanism includes 13 species.

This problem is rather standard, and was included in a series of benchmark problems in laminar flame computations sixteen years ago [15]. We note that our mechanism does differ from the one used there. Our complete set of results is summarized in Table 1. Our computed flame speeds vary between 235.77cms and 239.08cms. If the somewhat less reliable sixth order results are ignored, this variance is reduced from 236.86 to 237.92 - i.e. about one-half of one percent. It is interesting to note that these results show far better agreement with the experimental results of Milton and Keck [14] than do any presented in [15], though this no doubt says more about the accuracy of our models than it does about the merits of our code. Plots of the temperature, velocity, and various mole fractions for a typical case (fourth order, 149 points, domain width $L = 1$) are presented in Figures 1 and 2.

In all cases we tried except one, the code successfully ran to completion. (The one failure was with the sixth order differencing and our coarsest mesh, $n = 49$.) The maximum number of time steps was set to 7000 and the code automatically switched to damped Newton iterations whenever:

$$\frac{||dw/dt||}{||w||} \leq 5\text{sec}^{-1}. \qquad (7.0.51)$$

(Here w is the complete discrete solution.) The convergence criterion for Newton's method was a relative change less than 10^{-5}. This criterion was not always achieved, in which case the final estimated relative error is listed in the tolerance column. Such failures to fully converge are most likely due to a highly ill-conditioned Jacobian. Note that Newton's method fails to converge at all if the initial conditions for the time-stepping are used as its initial guess.

Mesh Refinement: Results are shown for varying numbers of mesh points and various orders. Fixing all other parameters, the results clearly converge under refinement.

Varying Order: Both the second and fourth order codes show consistent convergence behavior. However, fixing all other parameters, the flame speeds computed using the second order code are slightly smaller. One would, of course, hope for even better agreement here. Again this may be attributable to conditioning problems. The results for the sixth order code are a little worse, at least as judged by its convergence under mesh refinement. It is possible that the sixth order method is more susceptible to stability problems due to the large one-sided stencils it uses near the boundaries. It may be necessary to use some additional mesh clustering at the boundary to improve its performance.

Varying Domain Length, L: We have computed solutions on both a small domain, L = 1cm, and a large domain, L = 10cm. Again, for the second and fourth order codes there is excellent agreement (about 1/20 of one percent) between the results on different domains. This indicates that our artificial boundary conditions are adequate for the domain lengths considered.

Varying x_h: Fixing our domain to be $x \in [0, 1]$, we used the fourth order method with 149 points to compute solutions with x_h varying from 1/4 (our standard value) to 3/4. The flame speed varied between 236.86 and 237.90. Results for $x_h \in [1/4, 1/2]$ showed far less variance. We expected this, as the decay rate of the solution to the unburnt state is far greater than its decay to the burnt state.

Varying μ: We also varied the damping parameter, μ, with the fourth order method, $x_h = 1/4$, $L = 1$ and $n = 149$. There were significant changes in the number of time steps required, with the best results obtained with $\mu = 1000$. Surprisingly, the flame speeds also varied somewhat. Note that varying μ only changes the initial guess from the point of view of the Newton iterator. The discrepancies in the results are another indication of ill-conditioning.

Time Accurate Simulations: We have, at this time, far less experience running our code for time-dependent problems. Therefore, we include only one preliminary result. It involves using the stationary flame profile as initial data and setting the inlet velocity to zero. The flame then propagates to the right. From the graphs it is clear that the profile and speed are correctly maintained. The data shown were obtained using the fourth order method in space and time with 99 mesh points.

8 Conclusions

Our fundamental conclusion, based not only on the results shown here, but on numerous others to be published at a later date, is that our code is both accurate and robust for stationary plane flame calculations. We must, of course, carry out many more experiments to assess its behavior for time-dependent and multidimensional problems. The results here do point to some difficulties in achieving high accuracy, which are most likely due to poor conditioning. This issue should be investigated further.

References

[1] A. Bayliss, D. Gottlieb, B. Matkowsky and M. Minkoff, "An adaptive pseudo-spectral method for reaction diffusion problems", *J. Comp. Phys.*, 81, (1989), 421-443.

[2] A. Bayliss and E. Turkel, "Mappings and accuracy for Chebyshev pseudo-spectral computations", *J. Comp. Phys.*, 101, (1992), 349-359.

[3] M. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics", *J. Comp. Phys.*, 82, (1989), 64-84.

[4] W.-J. Beyn, "The numerical computation of connecting orbits in dynamical systems", *IMA J. Num. Anal.*, 9, (1990), 379-405.

[5] T.P. Coffee, "A computer code for the solution of the equations governing a laminar, premixed, one-dimensional flame", Memorandum Report ARBRL-MR-03165, Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland, (1982).

[6] P. Deuflhard, "Recent progress in extrapolation methods for ordinary differential equations", *SIAM Review*, 27, (1985), 505-535.

[7] A. Dutt, L. Greengard and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations", preprint.

[8] P. Embid, "Well-posedness of the nonlinear equations for zero Mach number combustion", *Comm. P.D.E.*, 12, (1987), 1227-1283.

[9] S. Gordon and 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).

[10] K. Kailasanath, E.S. Oran and J.P. Boris, "A one-dimensional time-dependent model for flame initiation, propagation and quenching", NRL Memorandum Report No. 4910, Naval Research Laboratory, Washington, D.C., (1982).

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

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

[13] A. Majda and J. Sethian, "The derivation and numerical solution of the equations for zero Mach number combustion", *Comb. Sci. Tech.*, 42, (1985), 185-205.

[14] B. Milton and J. Keck, "Laminar burning velocities in stoichiometric hydrogen and hydrogen-hydrocarbon gas mixtures", *Comb. and Flame*, 58, (1984), 13-22.

[15] N. Peters and J. Warnatz, *Numerical Methods in Laminar Flame Propagation*, Notes on Numerical Fluid Mechanics Vol. 6, Vieweg, Braunschweig, (1982).

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

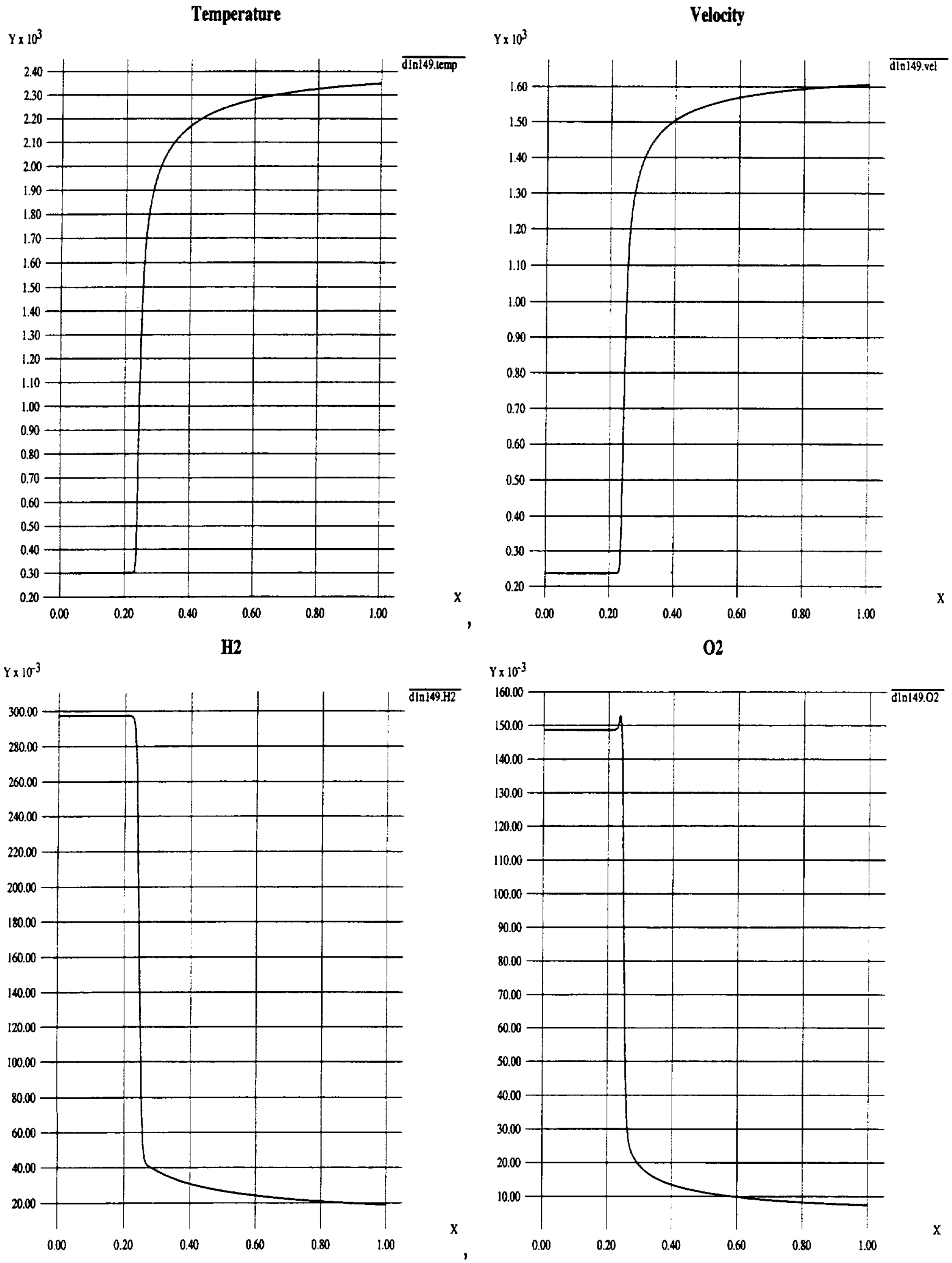
[17] M. Sermange, "Contribution to the numerical analysis of laminar stationary flames", *Reacting Flows: Combustion and Chemical Reactors*, Part 2, G.S.S. Ludford, ed., Lectures in Applied Mathematics, AMS, Providence, RI, (1986), 395-414.

[18] F.A. Williams, *Combustion Theory*, Second Edition, Benjamin/Cummings Publishing Company, Menlo Park, CA, (1985).

Appendix

Order	L	Mesh Pts.	x_h	μ	Time Steps	Tol.	v
2	1	49	$L/4$	50	1514	9×10^{-4}	237.35
2	1	49	$L/4$	50	1514	9×10^{-4}	237.35
2	1	74	$L/4$	50	2342	1×10^{-5}	237.21
2	1	99	$L/4$	50	1022	1×10^{-5}	237.32
2	1	149	$L/4$	50	1417	1×10^{-5}	237.46
2	1	199	$L/4$	50	1845	1×10^{-5}	237.47
2	1	249	$L/4$	50	1838	1×10^{-5}	237.46
4	1	49	$L/4$	50	7000	1×10^{-3}	237.43
4	1	74	$L/4$	50	3503	1×10^{-5}	237.81
4	1	99	$L/4$	50	2032	1×10^{-5}	237.41
4	1	149	$L/4$	50	1550	1×10^{-5}	237.92
4	1	199	$L/4$	50	1507	1×10^{-5}	237.82
4	1	249	$L/4$	50	1525	1×10^{-5}	237.78
6	1	74	$L/4$	1000	2716	1×10^{-5}	238.52
6	1	99	$L/4$	1000	1089	1×10^{-5}	236.16
6	1	149	$L/4$	1000	714	2×10^{-4}	235.64
6	1	199	$L/4$	1000	775	1×10^{-5}	238.61
4	1	149	$L/4$	10	2524	1×10^{-5}	236.91
4	1	149	$L/4$	250	1023	1×10^{-5}	237.96
4	1	149	$L/4$	500	799	1×10^{-5}	236.86
4	1	149	$L/4$	1000	768	1×10^{-5}	236.86
4	1	149	$L/4$	2000	903	1×10^{-5}	236.88
4	1	149	$L/3$	1000	780	1×10^{-5}	237.10
4	1	149	$L/2$	1000	759	1×10^{-5}	236.98
4	1	149	$2L/3$	1000	766	1×10^{-5}	237.90
4	1	149	$3L/4$	1000	735	1×10^{-5}	237.63
2	10	999	$L/4$	50	5434	1×10^{-5}	237.31
2	10	1999	$L/4$	50	5352	1×10^{-5}	237.48
2	10	2999	$L/4$	50	5670	1×10^{-5}	237.50
4	10	999	$L/4$	50	7000	1×10^{-5}	236.86
4	10	1499	$L/4$	50	7000	1×10^{-5}	237.92
4	10	1999	$L/4$	50	5381	1×10^{-5}	237.92
6	10	999	$L/4$	1000	1517	1×10^{-5}	235.77
6	10	999	$L/4$	50	5030	1×10^{-5}	236.09
6	10	1499	$L/4$	50	5713	1×10^{-5}	238.81
6	10	1999	$L/4$	1000	1722	1×10^{-5}	236.13
6	10	1999	$L/4$	50	5313	1×10^{-5}	239.08

Table 1: Summary of Results



Temperature

Velocity

H2

O2

Figure 1. Computed Solution, Order= 4, $n = 149$, $L = 1$.

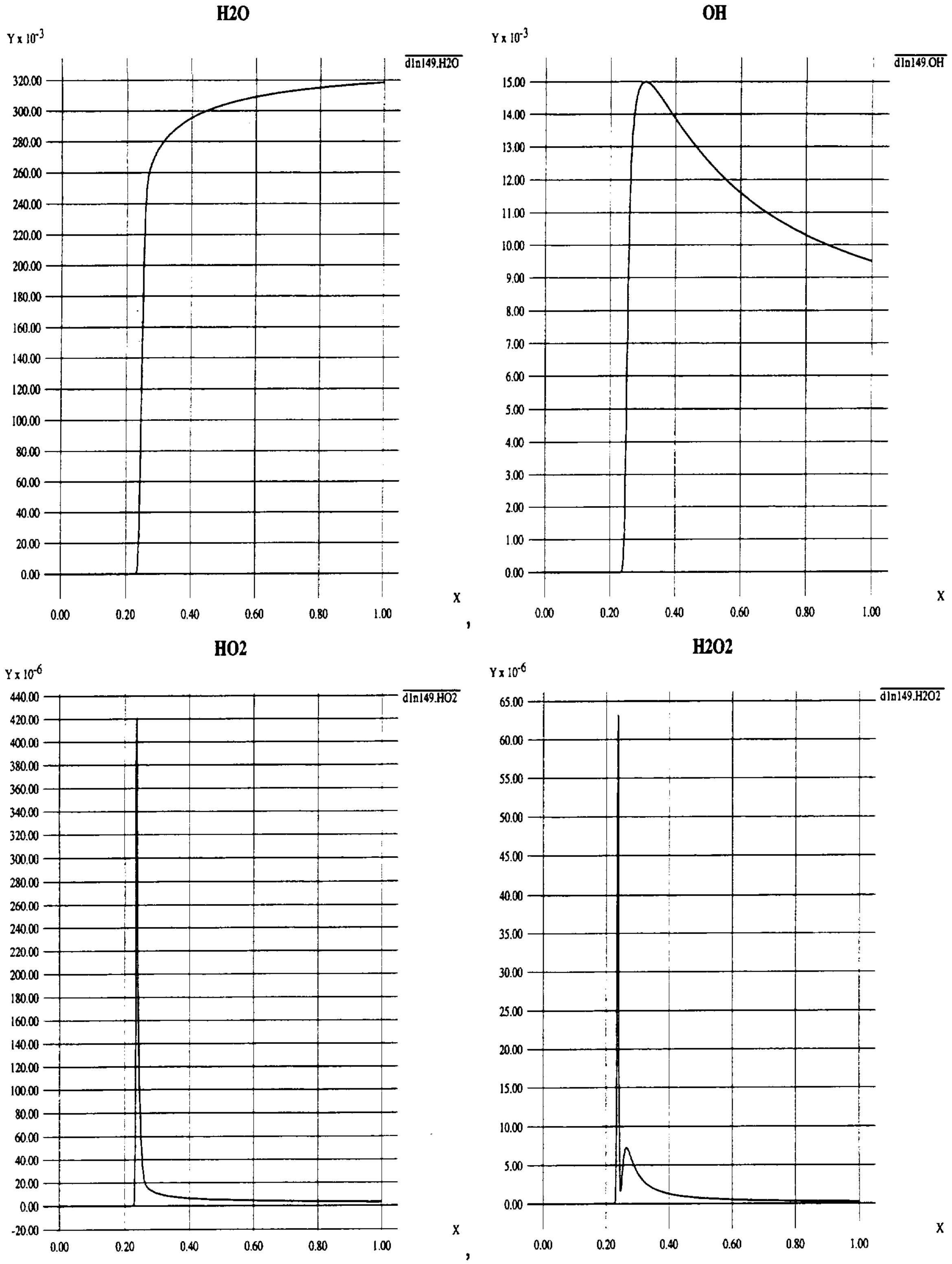


Figure 2. Computed Solution, Order= 4, $n = 149$, $L = 1$.

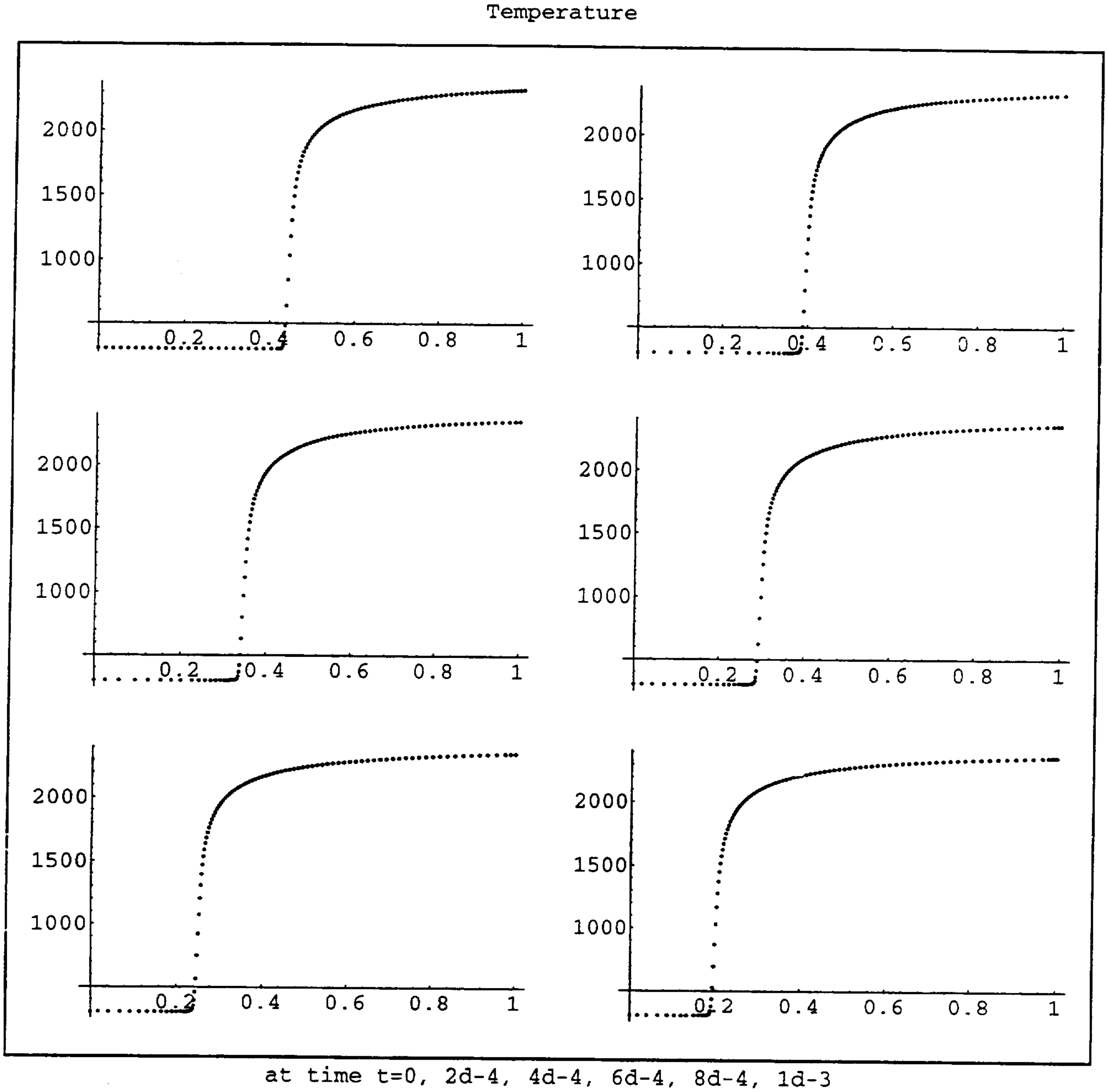


Figure 3. Unsteady Sol. $n = 99$, $L = 1$.

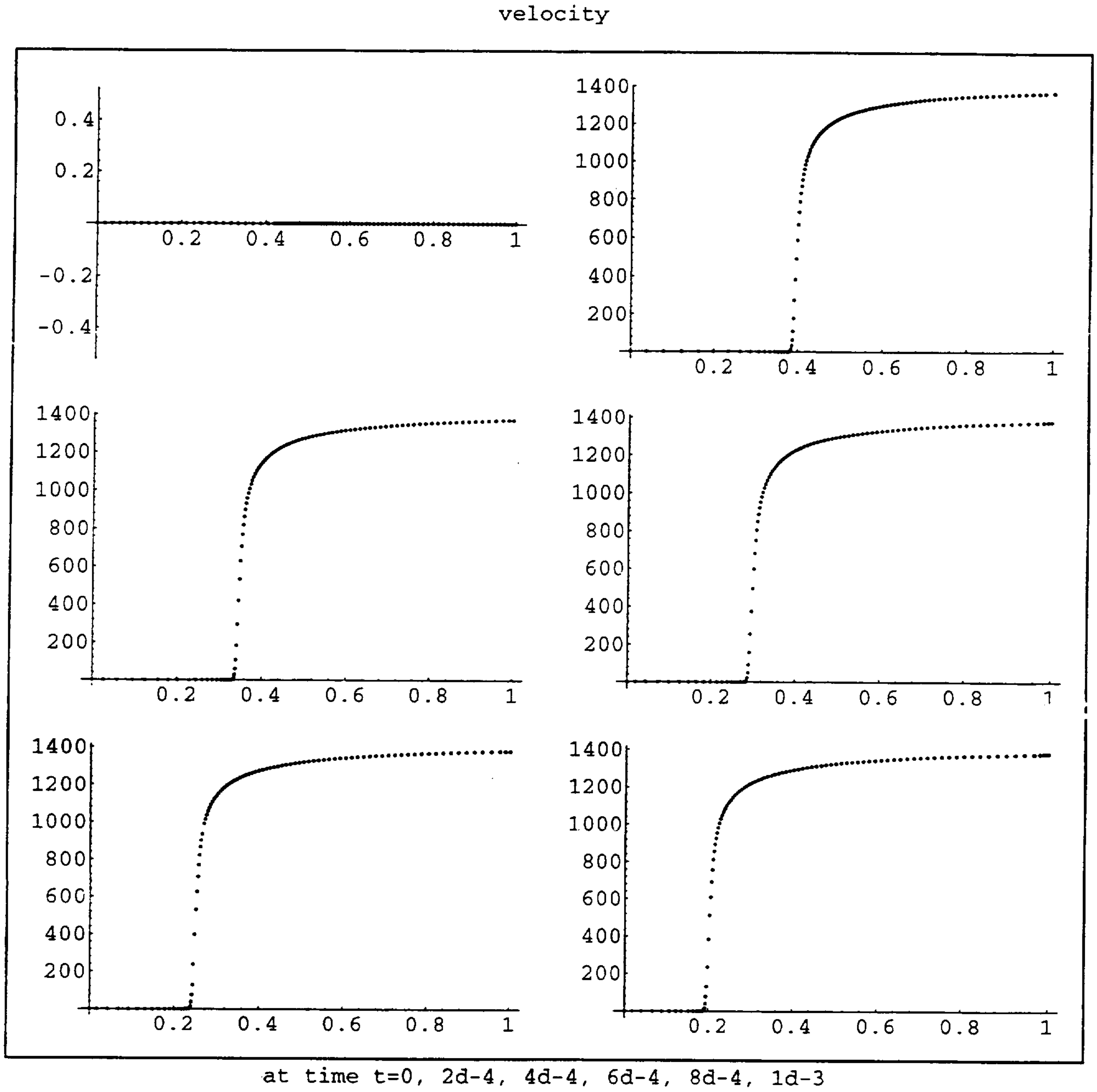


Figure 4. Unsteady Sol. $n = 99$, $L = 1$.

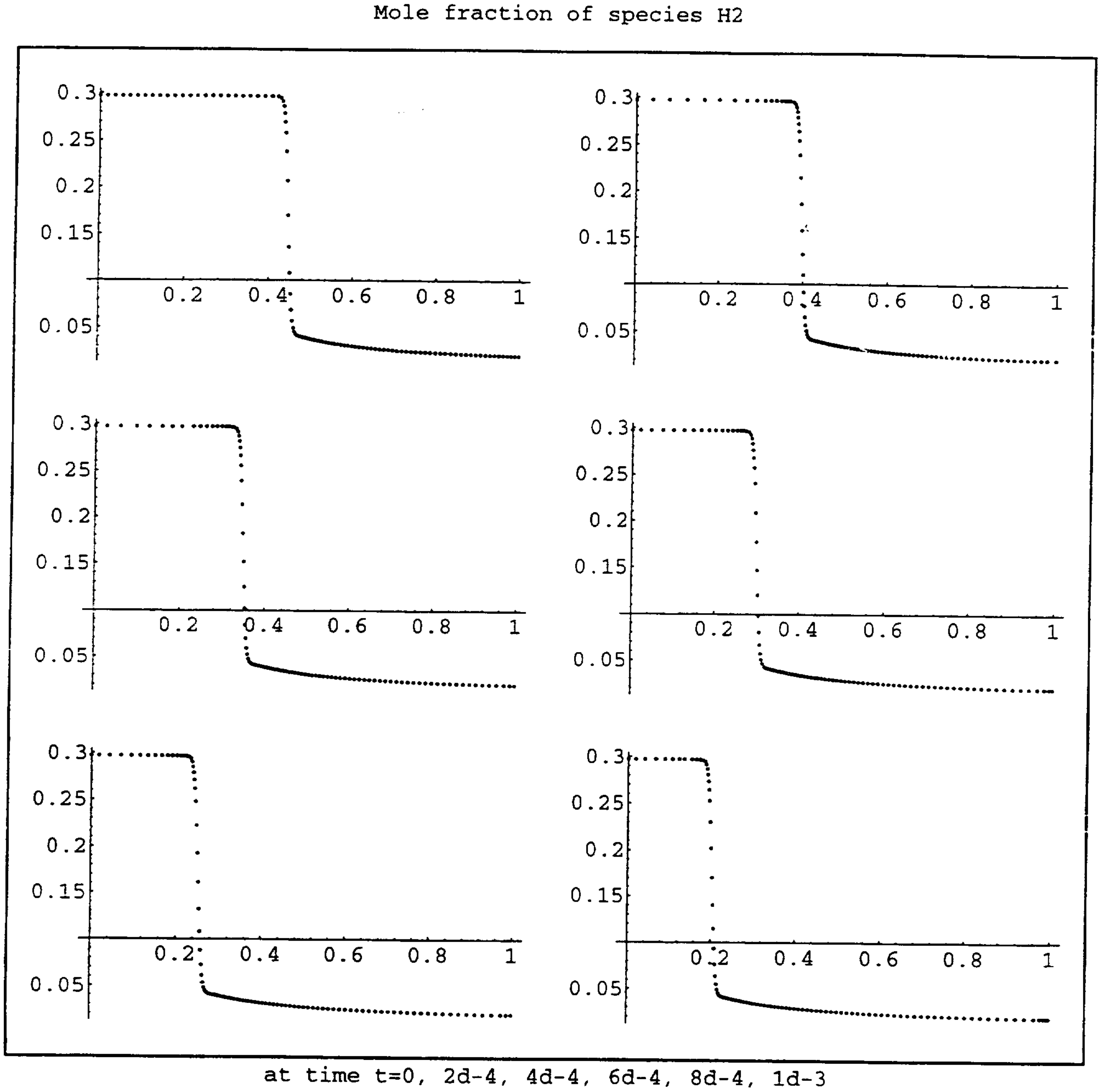


Figure 5. Unsteady Sol. $n = 99$, $L = 1$.

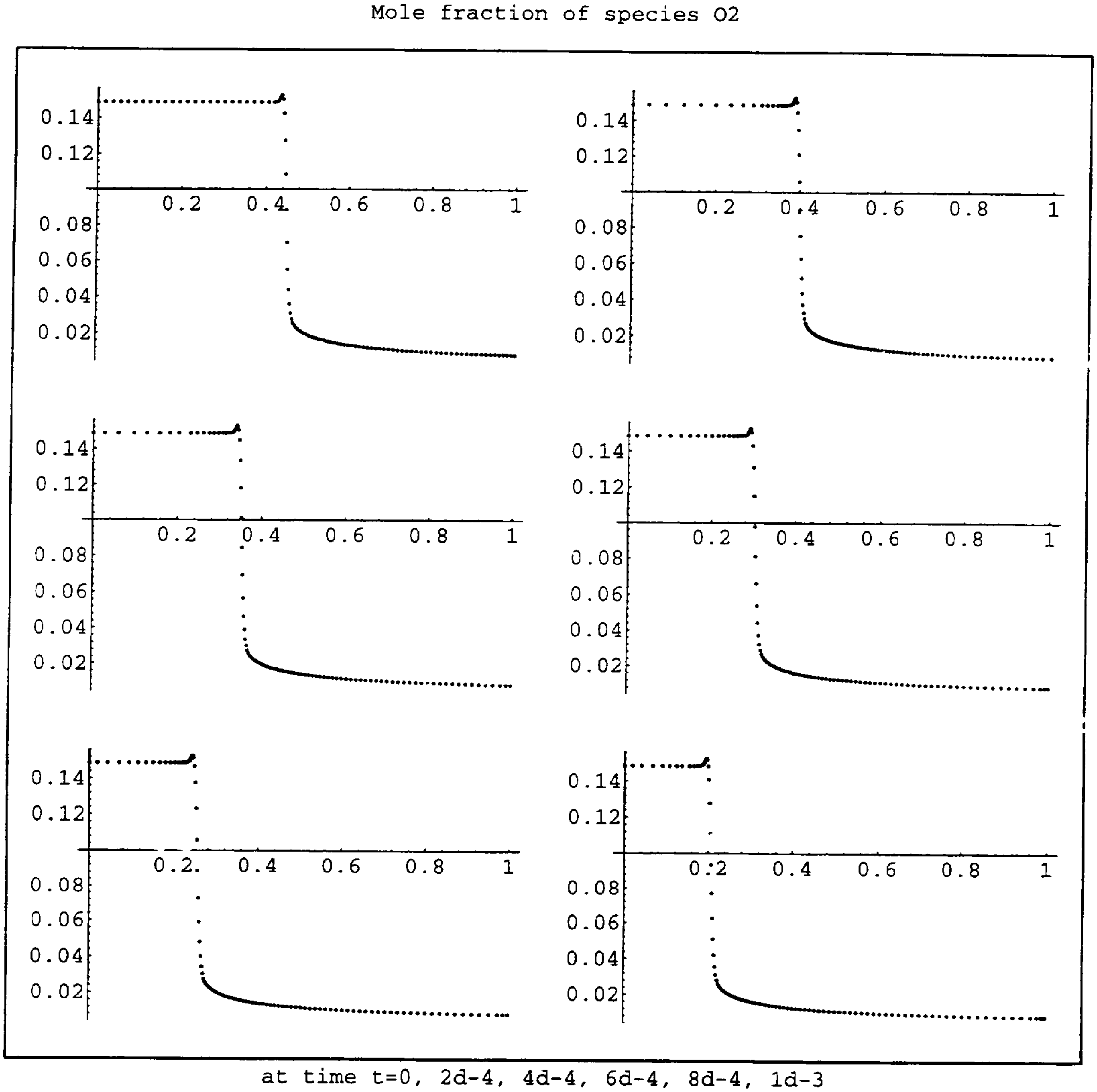


Figure 6. Unsteady Sol. $n = 99$, $L = 1$.

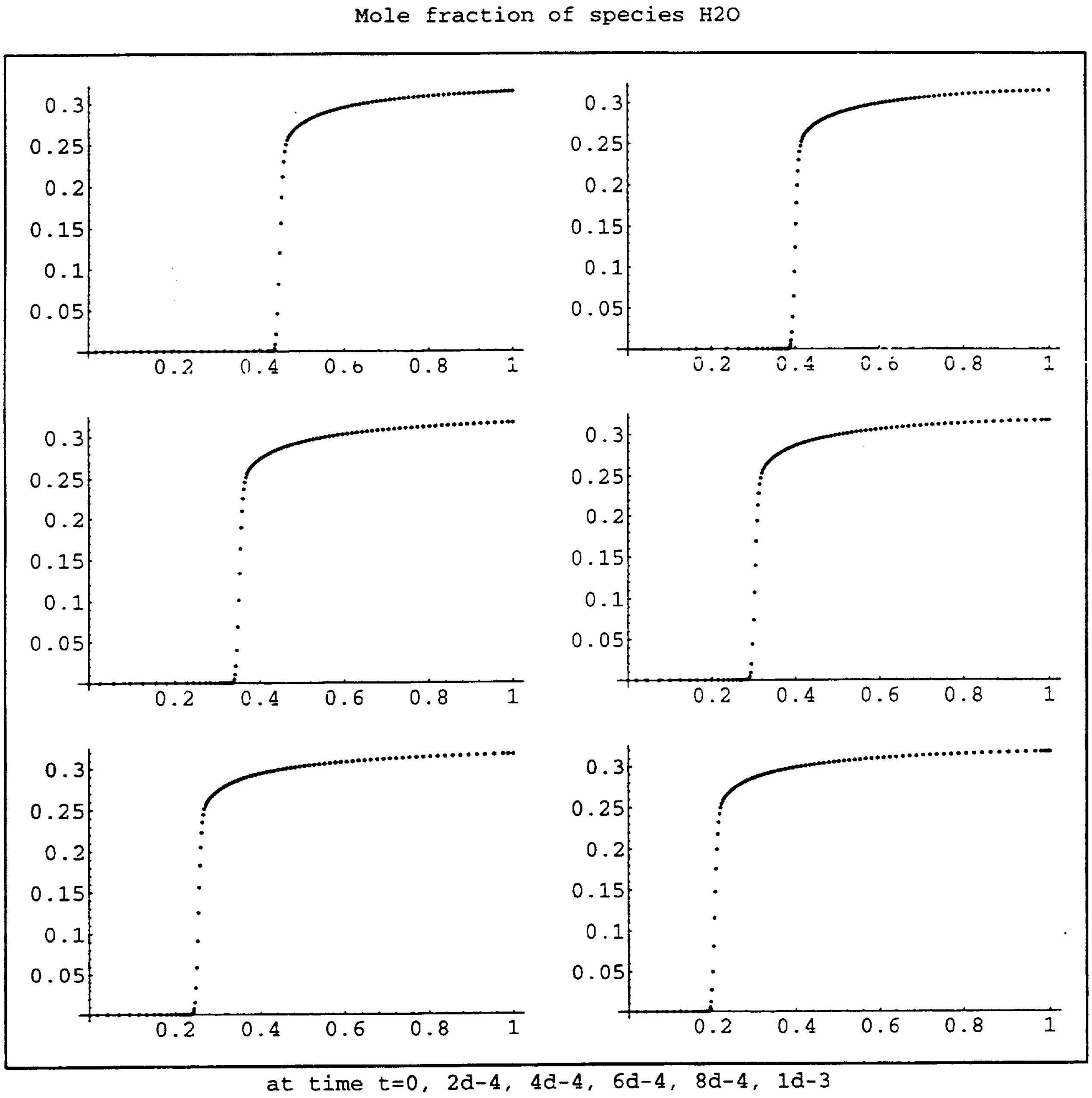


Figure 7. Unsteady Sol. $n = 99$, $L = 1$.

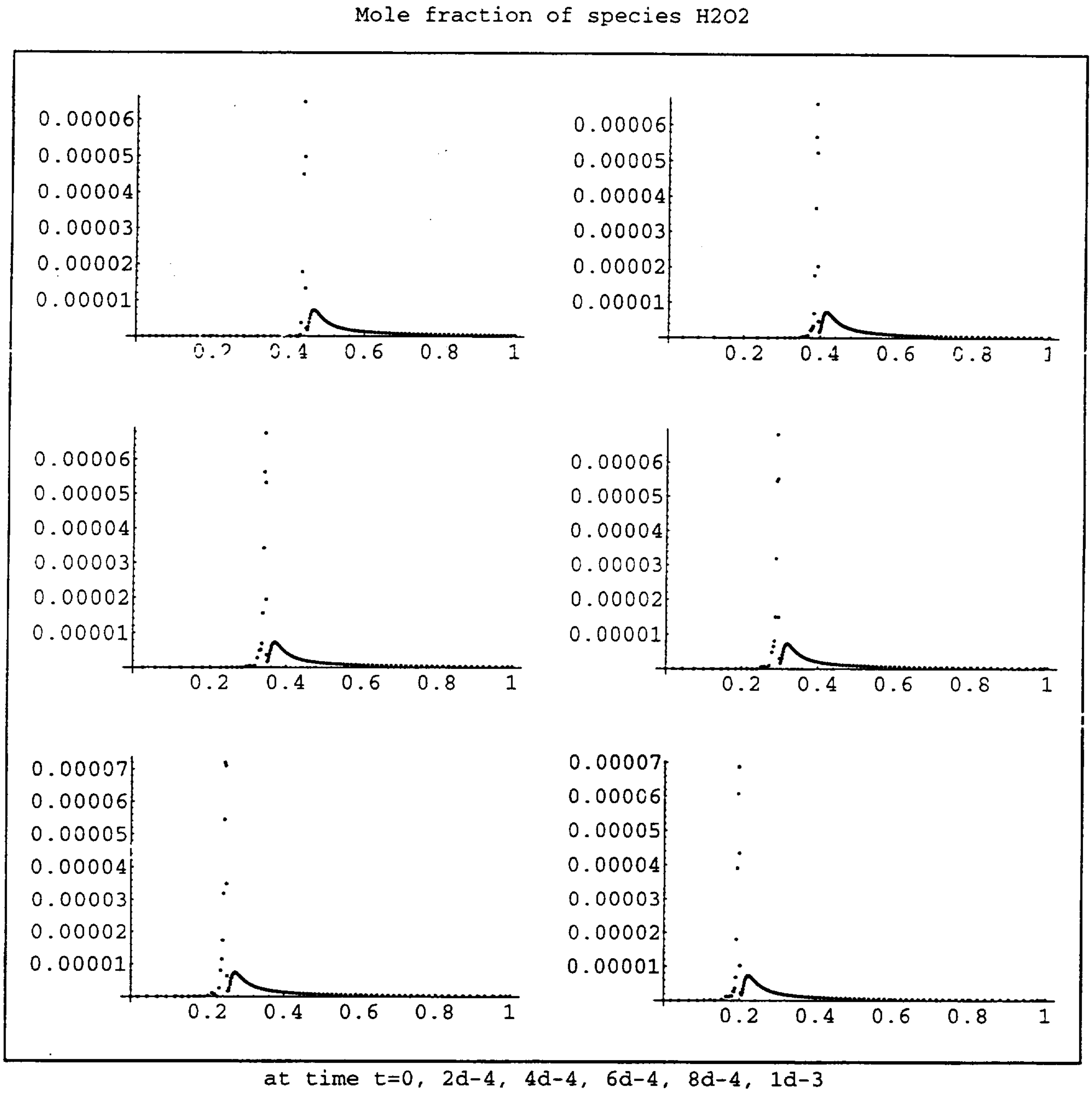


Figure 8. Unsteady Sol. $n = 99$, $L = 1$.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1999	Contractor Report

4. TITLE AND SUBTITLE

Computation of Steady and Unsteady Laminar Flames: Theory

5. FUNDING NUMBERS

WU–523–26–33–00
NCC3–622

6. AUTHOR(S)

Thomas Hagstrom, Krishnan Radhakrishnan, and Ruhai Zhou

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Computational Mechanics in Propulsion
22800 Cedar Point Road
Cleveland, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER

E–11826

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—1999-209305
AIAA 98–3246
ICOMP–99–07

11. SUPPLEMENTARY NOTES

Prepared for the 34th Joint Propulsion Conference and Exhibit cosponsored by AIAA, ASME, SAE, and ASEE, Cleveland, Ohio, July 13–15, 1998. Thomas Hagstrom and Ruhai Zhou, University of New Mexico, Department of Mathematics and Statistics, Albuquerque, New Mexico 87131; Krishnan Radhakrishnan, Institute for Computational Mechanics in Propulsion, NASA Glenn Research Center, Cleveland, Ohio 44135. ICOMP Program Director, Lou Povinelli, organization code 5880, (216) 433–5818.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category: 34 Distribution: Nonstandard

This publication is available from the NASA Center for AeroSpace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this paper we describe the numerical analysis underlying our efforts to develop an accurate and reliable code for simulating flame propagation using complex physical and chemical models. We discuss our spatial and temporal discretization schemes, which in our current implementations range in order from two to six. In space we use staggered meshes to define discrete divergence and gradient operators, allowing us to approximate complex diffusion operators while maintaining ellipticity. Our temporal discretization is based on the use of preconditioning to produce a highly efficient linearly implicit method with good stability properties. High order for time accurate simulations is obtained through the use of extrapolation or deferred correction procedures. We also discuss our techniques for computing stationary flames. The primary issue here is the automatic generation of initial approximations for the application of Newton's method. We use a novel time-stepping procedure, which allows the dynamic updating of the flame speed and forces the flame front towards a specified location. Numerical experiments are presented, primarily for the stationary flame problem. These illustrate the reliability of our techniques, and the dependence of the results on various code parameters.

14. SUBJECT TERMS

Flame propagation; Physical and chemical models; Numerical analysis; Implicit method

15. NUMBER OF PAGES

23

16. PRICE CODE

A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	