# Unsteady Strained Flames: Fundamentals and Numerical Modeling*

Youssef Marzouk
Reacting Gas Dynamics Laboratory
MIT Department of Mechanical Engineering

## 1 Introduction

Understanding the response of laminar flames to steady and unsteady strain rates is crucial to modeling the coupled interactions of turbulent reacting flow [18, 7, 12, 14, 1, 4, 15, 13, 6, 8, 2]. Strain affects a flame by changing scalar gradients and thus the rates of diffusive transport feeding the flame. Accurately predicting the effects of strain therefore requires detailed models for chemistry and species transport. However, certain qualitative results can be summarized:

Consider a premixed flame in which, for simplicity, a single Lewis number ($Le \equiv \alpha/D$) describes the transport of all species. The direct impact of strain is on the convection-diffusion zone of the flame, but the actual effect of strain on burning, i.e., on the reaction zone, depends closely on the Lewis number. For a unity Lewis number, mass and heat diffusion rates are equal, and the reaction-diffusion zone is essentially unaffected over a range of weak to intermediate strains. Once the flow time scale imposed by the strain becomes comparable to the chemical time scale, however, the temperature in the reaction zone begins to fall and the burning rate decreases. Here the Karlovitz number is of order unity; at even higher strains (and values of $Ka$), flame extinction may be observed.

For nonunity Lewis number, the imbalance between mass and heat flux leads to changes in flame structure at weak and intermediate strains. Strain

*Lecture in *Fundamentals and Modeling in Combustion*, 2.280, April 2001

shifts values of temperature and fuel mass fraction relative to one another, thus altering the conditions in the reaction zone. For $Le > 1$, heat diffuses more quickly than mass; a smoother temperature profile reduces the temperature in the reaction zone, and the burning rate decreases monotonically with strain. For $Le < 1$, fuel diffuses more quickly than heat, and temperature actually increases at the location of the maximum reaction rate; the burning rate thus increases with weak to intermediate strain.

At high strain, Lewis number effects become less important. As described above for unity Lewis number, the burning rate at $Ka > 1$ must decrease for nonunity and unity Lewis number flames alike. Flame extinction occurs as the influx of cold reactants exceeds the rate of heat release from chemical reaction and the reaction zone falls below the ignition temperature. Numerical results supporting these trends can be found in [12].

Of course, a single Lewis number applied to all species cannot describe a real reacting flow. This simplified discussion of strain at varying Lewis numbers emphasizes the importance of detailed chemistry and transport in accurately modeling a flame, even to predict the steady-state effect of a given strain rate. In a realistic flame, the diffusive flux of each mixture component reacts differently to a change in strain rate, and the superposition of these diffusive fluxes changes the composition and temperature of the reaction zone, where, in turn, a complex set of chemical pathways determines the burning rate.

Now consider unsteady strain rates: The effect of oscillatory strain on a flame depends on the frequency of the applied oscillations. At low frequency the flame may respond in a quasi-steady manner; at moderate frequency, the flame responds with a time lag, and thus a phase shift; while at even higher frequency, a flame may be unresponsive to oscillations in strain. The bounds between these regimes depend on the relative sizes of the flame time scale and the period of strain oscillations. It is also important to note that a flame may survive at higher unsteady strains than it would in a steady-state environment, a result reported by Najm [9] and others. Moreover, flames in a turbulent environment can undergo quenching and subsequent re-ignition. All these effects must be captured to model the effects of flow inhomogeneity on combustion.

# 2 Strain in Multi-Dimensional Reacting Flow

## 2.1 Kinematics of Flame Front Stretching

Combustion in a turbulent flow may take the form of a flame surface convoluted over a range of length scales by vortical structures. Of course, this model is not applicable to all regimes of turbulent combustion. Turbulent combustion processes may be classified by Damköhler and Karlovitz numbers. The Damköhler number is the ratio of the flow (integral) time scale to chemical time scale

$$Da = \frac{\tau_F}{\tau_R} \tag{1}$$

while the Karlovitz number is taken as the ratio of the flame stretch rate to a critical stretch rate, using $l_F$ as a flame thickness and $s_L$ as the laminar flame speed:

$$Ka = \frac{\frac{1}{A}\frac{dA}{dt}}{s_L/l_F} \tag{2}$$

The limit of fast chemistry corresponds to $Da \gg 1$, while one expects local quenching and distributed reaction zones for $Ka > 1$. Thus the model of a continuous flame surface wrinkled or corrugated by turbulence, as described here, is valid in the regime $Da \gg 1$, $Ka \leq 1$.

Now consider the kinematics of a material element on a two-dimensional flame front, as shown in Figure 1. The motion of any such fluid line segment may be decomposed into translation, rotation, and strain. In flame coordinates—that is, in the reference frame of the fluid line segment—the only effect of motion that will be apparent is that of strain. Translation and rotation may change the mixture that the flame front encounters, but only strain will alter the flame's internal structure. The strain is given by the relative change in length of the line segment:

$$\epsilon = \frac{1}{|\delta\mathbf{l}|}\frac{d\delta\mathbf{l}}{dt} = \frac{\delta\mathbf{l}\cdot\nabla u}{|\delta\mathbf{l}|} \tag{3}$$

Further insight may be obtained by decomposing the flow local to the flame front. Figure 2 shows two initially perpendicular line segments, $l$ and $m$, in a reference frame $\mathcal{F}$ chosen to follow a flame element. The x-axis of this reference frame is always parallel to the flame surface, while the y-axis

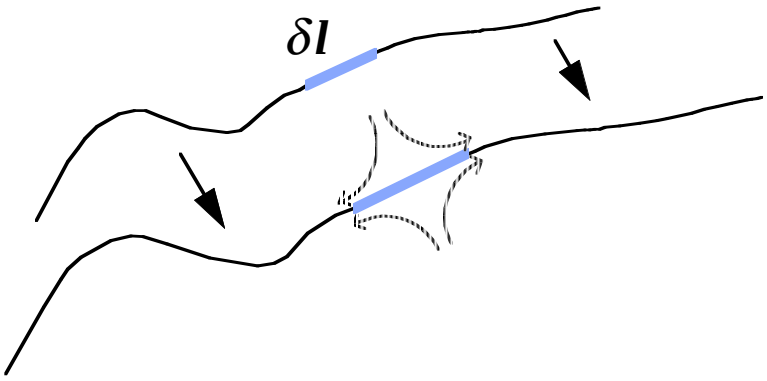


Figure 1: Deformation of material element on a flame front.

is always perpendicular to it. The line segment $l$ is a material element of the flame front and thus always lies on the x-axis, while the line segment $m$ is initially coincident with the y-axis. At an initial time $t$, the projections of the the line segments on the axes are

$$l_x(t) = l(t),\ l_y(t) = 0 \tag{4}$$

$$m_x(t) = 0,\ m_y(t) = m(t) \tag{5}$$

After an elapsed time $dt$, the projections become

$$l_x(t+dt) = \left(1 + \frac{\partial u}{\partial x}dt\right) l(t) \tag{6}$$

$$l_y(t+dt) = \left(\frac{\partial v}{\partial x}dt\right) l(t) = 0 \tag{7}$$

$$m_x(t+dt) = \left(\frac{\partial u}{\partial y}dt\right) m(t) \tag{8}$$

$$m_y(t+dt) = \left(1 + \frac{\partial v}{\partial y}dt\right) m(t) \tag{9}$$

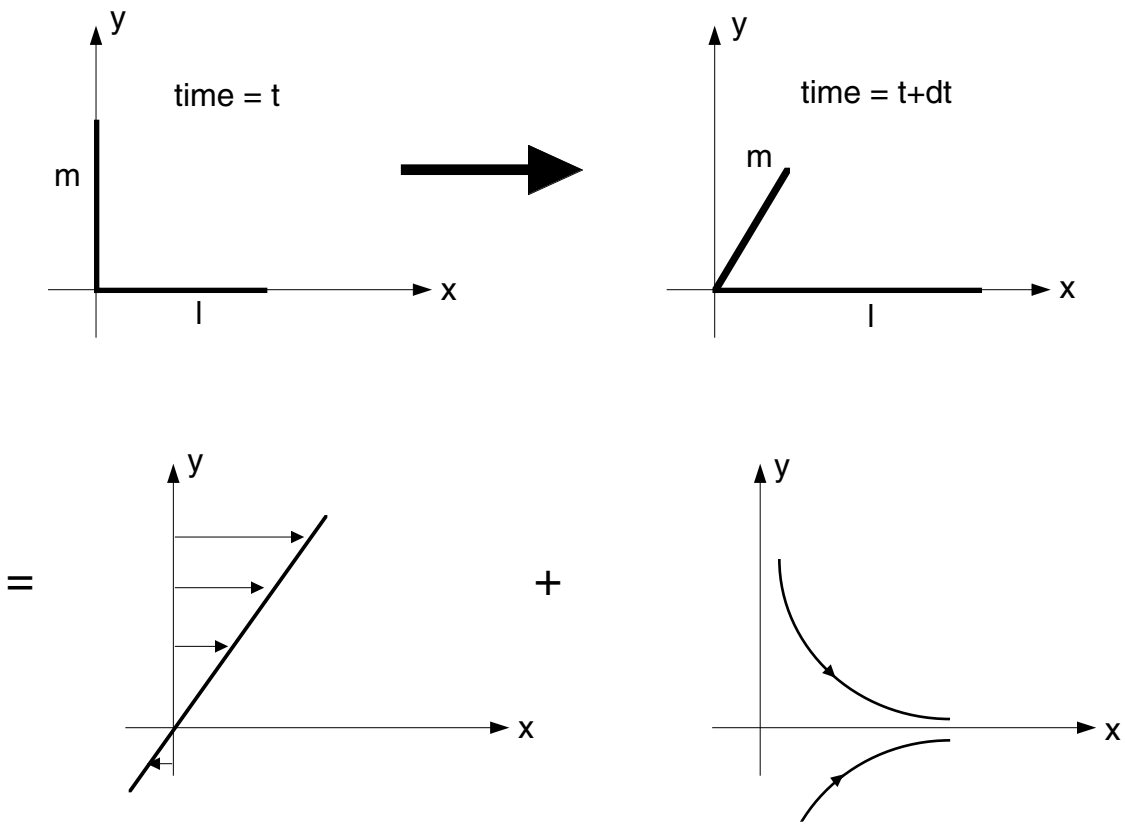


Figure 2: Flow-induced deformation in the reference frame of a flame element.

where $u$ and $v$ are the flow velocities in the $x$ and $y$ directions, respectively. Note that our choice of reference frame requires the derivative in Equation 7 to equal zero.

As expressed in Equation 8, the projection $m_x$ grows at the rate $\partial u/\partial y$; the segment initially in the y-direction thus rotates without changing its length. This deformation is consistent with a *pure shear flow*, corresponding to off-diagonal components of the strain rate tensor:

$$\tau_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{10}$$

The second component of flow local to the flame element is a *stagnation point flow*, observed in Equations 6 and 9. Both line segments are strained along the principal axes, and because of continuity in an incompressible flow, the rates of strain are equal and opposite in sign:

$$\epsilon = \frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y} \tag{11}$$

Decomposition of the flow in the vicinity of a flame element is shown schematically in Figure 2.

What does this flow decomposition imply for modeling the effect of inhomogeneities on a flame surface? For one, scalar gradients (e.g., gradients of temperature, density, species concentrations) normal to the flame are much larger than those tangential to the flame. In the reference frame $\mathcal{F}$, this can be written

$$\frac{\partial}{\partial x} \ll \frac{\partial}{\partial y} \tag{12}$$

Now consider the transport equation for a scalar in $\mathcal{F}$; this equation must contain the following differential operator:

$$\mathcal{L} \equiv \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \tag{13}$$

The velocity of the shear flow, $u_{shear} = cy$, contributes only to the $u$ in the second term of Equation 13. Following Equation 12, this second term can be neglected entirely, rendering the shear component of the velocity field unimportant in local transport of scalars. Physically, this can be understood from Figure 2; the shear field simply advects constant values of scalars parallel to

the flame surface and cannot alter the distribution of scalars normal to the flame. As a result, the instantaneous burning of a turbulent flame surface is locally similar to that of a stagnation point flow with an appropriately chosen strain rate.

## 2.2 Expressions for the Strain Rate on a Flame Surface

More general expressions for the rate of strain of a flame surface (or, for that matter, any material surface in a fluid) can be obtained as follows. Motion in the vicinity of any point in the fluid can be resolved into a uniform translation, a rigid-body rotation, and a pure straining motion given by the strain-rate tensor $\mathbf{e}$:

$$\mathbf{e} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) \tag{14}$$

The pure straining motion may itself be decomposed into simpler flows; as argued above, the element of this decomposition most relevant to the internal structure of a locally planar sheet is stagnation-point flow [20]. Stagnation point flow may be characterized by the normal component of the strain rate tensor

$$\epsilon \equiv \mathbf{n}\cdot\mathbf{e}\cdot\mathbf{n} \tag{15}$$

which in a region of constant density is equal to $-\mathbf{t}\cdot\mathbf{e}\cdot\mathbf{t}$, where $\mathbf{t}$ is the local tangent vector to the flame surface.

It is important to note that the preceding analysis does not account for curvature of the flame surface. (The line segments displaced in Figures 1 and 2 are perfectly straight.) Propagation of a curved flame contributes to the total *stretch* rate as follows:

$$\frac{1}{|\delta\mathbf{l}|}\frac{d\delta\mathbf{l}}{dt} = \epsilon + 2s_d h_m \tag{16}$$

where $\epsilon$ is the ordinary hydrodynamic strain, $s_d$ is the speed of the flame surface normal to itself, and $h_m$ is the mean curvature. This effect is relevant only to premixed flames ($s_d \neq 0$), and typically becomes important only in regions of high curvature (where the radius of curvature is small relative to the flame thickness)—in the cusp of a vortical structure, for example. For simplicity, the flame models described in subsequent sections will ignore curvature effects.

# 3 Governing Equations

## 3.1 Unsteady Strained Laminar Flame

Consider a flame strained in a planar stagnation point flow, with coordinates and flow velocities shown in Figure 3. The flame introduces directionality to this figure and enables use of the boundary layer approximation; for any scalar $\varphi$, gradients parallel to the flame $\partial^2\varphi/\partial x^2$ are negligible compared to those perpendicular to the flame, $\partial^2\varphi/\partial y^2$. We thus write unsteady boundary layer equations for transport of species, energy, momentum, and mass:

$$\rho\frac{\partial Y_k}{\partial t} + \rho u\frac{\partial Y_k}{\partial x} + \rho v\frac{\partial Y_k}{\partial y} = -\frac{\partial}{\partial y}\left(\rho Y_k \mathcal{V}_k\right) + \dot{w}_k W_k \tag{17}$$

$$\rho\frac{\partial T}{\partial t} + \rho u\frac{\partial T}{\partial x} + \rho v\frac{\partial T}{\partial y} = \frac{1}{c_p}\frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) - \frac{1}{c_p}\left(\sum_k^K \rho c_{p,k} Y_k \mathcal{V}_k\right)\frac{\partial T}{\partial y} - \frac{1}{c_p}\sum_k^K \dot{w}_k H_k \tag{18}$$

$$\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) \tag{19}$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial\left(\rho u\right)}{\partial x} + \frac{\partial\left(\rho v\right)}{\partial y} = 0 \tag{20}$$

where the diffusion velocity is:

$$\mathcal{V}_k = -\frac{1}{X_k} D_{km}\frac{\partial X_k}{\partial y} \tag{21}$$

and the mixture-averaged diffusion coefficient, $D_{km}$ is defined as:

$$D_{km} = \frac{1 - Y_k}{\sum_{j\neq k}^K X_j/\mathcal{D}_{jk}} \tag{22}$$

Here $Y_k$ is the mass fraction of species $k$ while $W_k$ and $\dot{w}_k$ are the molecular weight and molar production rate, respectively. In the remaining equations, $c_p$ is the specific heat of the mixture, $\lambda$ is the thermal conductivity, $H_k$ is

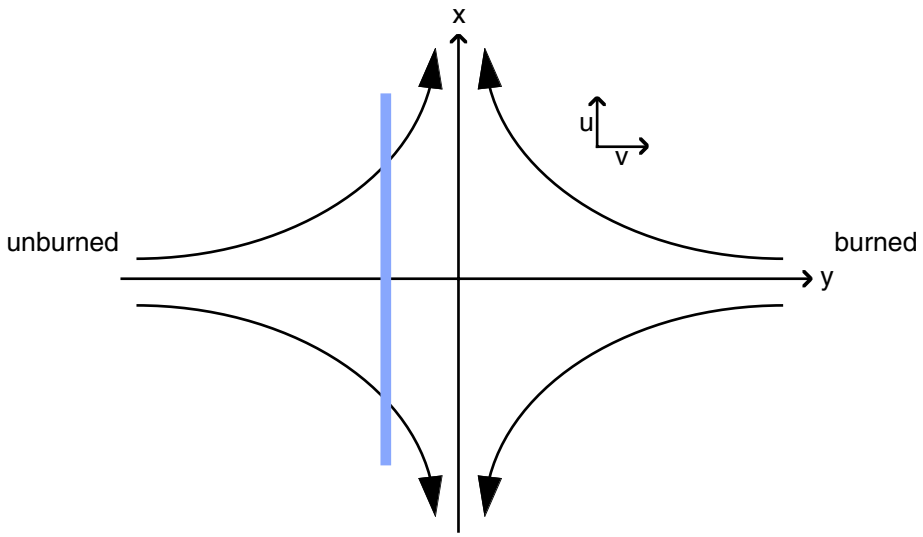


Figure 3: Flame strained in a stagnation point flow; the elemental flame model.

the molar enthalpy of species $k$, $p$ is the hydrodynamic pressure, and $\mu$ is the absolute viscosity of the mixture. Note that thermal diffusion velocity is neglected. Smooke [17] and others have shown that this effect is unimportant in strained hydrocarbon flames.

The pressure gradient $dp/dx$ inside the boundary layer is identical to that in the outer stagnation point flow, since by the usual boundary layer assumption, $\Delta p$ across the boundary layer is negligible. Substituting the stagnation point flow velocity profile $u_\infty = \epsilon(t)\, x$, $v_\infty = -\epsilon(t)\, y$ into the general x-direction momentum conservation equation, we obtain the pressure gradient as a function of the imposed strain:

$$\frac{\partial p}{\partial x} = -\rho_u u_\infty \epsilon - \rho_u \frac{u_\infty}{\epsilon}\frac{\partial \epsilon}{\partial t} \tag{23}$$

The notation $\rho_u$ above emphasizes that the density of the *unburned* mixture is used to define the pressure gradient. In the diffusion flame, densities outside of the boundary layer are equal, $\rho_u = \rho_{-\infty} = \rho_\infty$, so this distinction is moot. In the premixed flame, however, heat release within the flame leads to a products-side stream entering with lower density—and to maintain a constant pressure gradient, a higher effective strain. Defining the global strain $\epsilon$

on the unburned side, in accordance with the expression for pressure gradient above, thus insures consistency and clarity across all flame configurations.

Introducing the notation $U \equiv u/u_\infty$, $V \equiv \rho v$, and substituting the pressure gradient expression into the equation for momentum conservation inside the boundary layer (Equation 19), Equations 17–20 can be rewritten as follows:

$$\rho\frac{\partial Y_k}{\partial t} + \rho u_\infty U\frac{\partial Y_k}{\partial x} + V\frac{\partial Y_k}{\partial y} = -\frac{\partial}{\partial y}\left(\rho Y_k \mathcal{V}_k\right) + \dot{w}_k W_k \tag{24}$$

$$\rho\frac{\partial T}{\partial t} + \rho u_\infty U\frac{\partial T}{\partial x} + V\frac{\partial T}{\partial y} = \frac{1}{c_p}\frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) - \frac{1}{c_p}\left(\sum_k^K \rho c_{p,k} Y_k \mathcal{V}_k\right)\frac{\partial T}{\partial y} - \frac{1}{c_p}\sum_k^K \dot{w}_k H_k \tag{25}$$

$$\rho u_\infty \frac{\partial U}{\partial t} + \rho u_\infty U\frac{1}{\epsilon}\frac{\partial \epsilon}{\partial t} + \rho u_\infty^2 U\frac{\partial U}{\partial x} + \rho u_\infty U^2\epsilon + u_\infty V\frac{\partial U}{\partial y} = \rho_{ub} u_\infty\left(\frac{1}{\epsilon}\frac{\partial \epsilon}{\partial t} + \epsilon\right) + u_\infty\frac{\partial}{\partial y}\left(\mu\frac{\partial U}{\partial y}\right) \tag{26}$$

$$\frac{\partial \rho}{\partial t} + u_\infty U\frac{\partial \rho}{\partial x} + \rho U\epsilon + \rho u_\infty\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \tag{27}$$

Now consider the solution of the boundary layer equations along the stagnation streamline, $x = 0$. Along this streamline, $u = u_\infty U = 0$ and by symmetry, $\partial\varphi/\partial x = 0$, where $\varphi$ is a passive scalar—namely, $T$, $Y_k$, $U$, or $\rho$. Thus, we can eliminate the x-direction convective terms in Equations 24–27. Note, however, that $\partial u/\partial x$ and $\partial u_\infty/\partial x$ are nonzero on the stagnation streamline, and that this flow divergence should be reflected in the momentum and continuity equations. Every term in the momentum conservation equation (Equation 26) is proportional to $u_\infty$, so to retain this equation in the system, we divide by $u_\infty$. The resulting equation governs the strain rate through the reaction zone, where $\epsilon\, U$ can be thought of as the effective strain. The final set of governing equations takes the following form:

$$\rho\frac{\partial Y_k}{\partial t} + V\frac{\partial Y_k}{\partial y} + \frac{\partial}{\partial y}\left(\rho Y_k \mathcal{V}_k\right) - \dot{w}_k W_k = 0 \tag{28}$$

$$\rho\frac{\partial T}{\partial t} + V\frac{\partial T}{\partial y} - \frac{1}{c_p}\frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) + \frac{1}{c_p}\left(\sum_k^K \rho c_{p,k} Y_k \mathcal{V}_k\right)\frac{\partial T}{\partial y} + \frac{1}{c_p}\sum_k^K \dot{w}_k H_k = 0 \tag{29}$$

$$\rho\frac{\partial U}{\partial t} + \rho U\frac{1}{\epsilon}\frac{\partial \epsilon}{\partial t} + \rho U^2\epsilon + V\frac{\partial U}{\partial y} - \frac{\partial}{\partial y}\left(\mu\frac{\partial U}{\partial y}\right) - \rho_u\left(\frac{1}{\epsilon}\frac{\partial \epsilon}{\partial t} + \epsilon\right) = 0 \tag{30}$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial V}{\partial y} + \rho U\epsilon = 0 \tag{31}$$

Note that the low Mach number assumption has been used in the governing equations above. Density is calculated as a function of the temperature, species mass fractions, and thermodynamic pressure (assumed constant) via the ideal gas equation of state.

At various times, it will be useful to consider a flame strained in an axisymmetric stagnation point flow rather than in the planar configuration used above. Denoting the radial coordinate with $r$ and the axial coordinate with $y$, the velocity profile of an axisymmetric stagnation point flow is $u_\infty = \epsilon(t)\, r,\ v_\infty = -2\epsilon(t)\, y$. The governing equations for this configuration are identical to those derived above (Equations 28–30) with the exception of the continuity equation. The revised continuity equation is:

$$\frac{\partial \rho}{\partial t} + \frac{\partial V}{\partial y} + 2\rho U\epsilon = 0 \tag{32}$$

Unsteadiness may enter in the form of the species and temperature boundary conditions at $\pm\infty$; these conditions define the composition and temperature of the two incoming streams of the stagnation point flow.

$$y = \pm\infty : Y_k = Y_{k,\pm\infty}(t), T = T_{\pm\infty}(t) \tag{33}$$

The continuity equation requires only one boundary condition; typically, this boundary condition would specify zero velocity at the stagnation point, $V(y = 0) = 0$. Numerical considerations discussed in the next section, however, require that we impose a boundary condition at $y = -\infty$, and leave the stagnation point definition to fix the origin of the $y$-axis.

The momentum conservation equation requires two boundary conditions. At an unburned stream, $u = u_\infty$, so the boundary condition is by definition $U = 1$. Setting the spatial gradients in Equation 30 to zero gives the boundary condition burned stream. This far-field boundary condition places an important requirement on the size of the computational domain; the flame must be far enough from the $+\infty$ and $-\infty$ boundaries for spatial gradients in $U$ to vanish. In the case of steady strain, the burned-stream boundary condition on $U$ is

$$U_b = \sqrt{\frac{\rho_u}{\rho_b}} \tag{34}$$

where $\rho_b$ is the density of the burned mixture. For unsteady strain, we must integrate the resulting ODE for $U_b$:

$$\frac{\partial U_b}{\partial t} = -U_b^2 \epsilon - U_b \left( \frac{1}{\epsilon} \frac{\partial \epsilon}{\partial t} \right) + \frac{\rho_u}{\rho_b} \left( \frac{1}{\epsilon} \frac{\partial \epsilon}{\partial t} + \epsilon \right) \tag{35}$$

For a premixed flame, the effective strain rate in the incoming products-side mixture thus responds dynamically to the imposed strain rate.

Molar production rates $\dot{w}_k$ are composites of elementary rate expressions, which are given by the kinetic model. As an example, a moderately detailed C1 kinetic model for methane-air combustion (with 46 reactions among 16 species) is shown in Table 1. This model is due to Smooke [17].

## 3.2 Unstrained Steady Laminar Flame

To better illustrate the effect of strain on flame structure, we reproduce here the governing equations for an *unstrained* freely propagating laminar premixed flame.

$$\dot{m} \frac{\partial Y_k}{\partial y} + \frac{\partial}{\partial y} \left( \rho Y_k \mathcal{V}_k \right) - \dot{w}_k W_k = 0 \tag{36}$$

$$\dot{m} \frac{\partial T}{\partial y} - \frac{1}{c_p} \frac{\partial}{\partial y} \left( \lambda \frac{\partial T}{\partial y} \right) + \frac{1}{c_p} \left( \sum_k^K \rho c_{p,k} Y_k \mathcal{V}_k \right) \frac{\partial T}{\partial y} + \frac{1}{c_p} \sum_k^K \dot{w}_k H_k = 0 \tag{37}$$

$$\dot{m} = \rho v \tag{38}$$

| | REACTIONS CONSIDERED | | | (k = A T**b exp(-E/RT)) A | b | E |
|---|---|---|---|---|---|---|
| 1. | CH3+H=CH4 | | | 1.90E+36 | -7.0 | 9050.0 |
| 2. | CH4+O2=CH3+HO2 | | | 7.90E+13 | 0.0 | 56000.0 |
| 3. | CH4+H=CH3+H2 | | | 2.20E+04 | 3.0 | 8750.0 |
| 4. | CH4+O=CH3+OH | | | 1.60E+06 | 2.4 | 7400.0 |
| 5. | CH4+OH=CH3+H2O | | | 1.60E+06 | 2.1 | 2460.0 |
| 6. | CH2O+OH=HCO+H2O | | | 7.53E+12 | 0.0 | 167.0 |
| 7. | CH2O+H=HCO+H2 | | | 3.31E+14 | 0.0 | 10500.0 |
| 8. | CH2O+M=HCO+H+M | | | 3.31E+16 | 0.0 | 81000.0 |
| 9. | CH2O+O=HCO+OH | | | 1.81E+13 | 0.0 | 3082.0 |
| 10. | HCO+OH=CO+H2O | | | 5.00E+12 | 0.0 | 0.0 |
| 11. | HCO+M=H+CO+M | | | 1.60E+14 | 0.0 | 14700.0 |
| 12. | HCO+H=CO+H2 | | | 4.00E+13 | 0.0 | 0.0 |
| 13. | HCO+O=OH+CO | | | 1.00E+13 | 0.0 | 0.0 |
| 14. | HCO+O2=HO2+CO | | | 3.00E+12 | 0.0 | 0.0 |
| 15. | CO+O+M=CO2+M | | | 3.20E+13 | 0.0 | -4200.0 |
| 16. | CO+OH=CO2+H | | | 1.51E+07 | 1.3 | -758.0 |
| 17. | CO+O2=CO2+O | | | 1.60E+13 | 0.0 | 41000.0 |
| 18. | CH3+O2=CH3O+O | | | 7.00E+12 | 0.0 | 25652.0 |
| 19. | CH3O+M=CH2O+H+M | | | 2.40E+13 | 0.0 | 28812.0 |
| 20. | CH3O+H=CH2O+H2 | | | 2.00E+13 | 0.0 | 0.0 |
| 21. | CH3O+OH=CH2O+H2O | | | 1.00E+13 | 0.0 | 0.0 |
| 22. | CH3O+O=CH2O+OH | | | 1.00E+13 | 0.0 | 0.0 |
| 23. | CH3O+O2=CH2O+HO2 | | | 6.30E+10 | 0.0 | 2600.0 |
| 24. | CH3+O2=CH2O+OH | | | 5.20E+13 | 0.0 | 34574.0 |
| 25. | CH3+O=CH2O+H | | | 6.80E+13 | 0.0 | 0.0 |
| 26. | CH3+OH=CH2O+H2 | | | 7.50E+12 | 0.0 | 0.0 |
| 27. | HO2+CO=CO2+OH | | | 5.80E+13 | 0.0 | 22934.0 |
| 28. | H2+O2=2OH | | | 1.70E+13 | 0.0 | 47780.0 |
| 29. | OH+H2=H2O+H | | | 1.17E+09 | 1.3 | 3626.0 |
| 30. | H+O2=OH+O | | | 2.00E+14 | 0.0 | 16800.0 |
| 31. | O+H2=OH+H | | | 1.80E+10 | 1.0 | 8826.0 |
| 32. | H+O2+M=HO2+M | | | 2.10E+18 | -1.0 | 0.0 |
| | H2O | Enhanced by | 2.100E+01 | | | |
| | CO2 | Enhanced by | 5.000E+00 | | | |
| | H2 | Enhanced by | 3.300E+00 | | | |
| | CO | Enhanced by | 2.000E+00 | | | |
| | O2 | Enhanced by | 0.000E+00 | | | |
| | N2 | Enhanced by | 0.000E+00 | | | |
| 33. | H+O2+O2=HO2+O2 | | | 6.70E+19 | -1.4 | 0.0 |
| 34. | H+O2+N2=HO2+N2 | | | 6.70E+19 | -1.4 | 0.0 |
| 35. | OH+HO2=H2O+O2 | | | 5.00E+13 | 0.0 | 1000.0 |
| 36. | H+HO2=2OH | | | 2.50E+14 | 0.0 | 1900.0 |
| 37. | O+HO2=O2+OH | | | 4.80E+13 | 0.0 | 1000.0 |
| 38. | 2OH=O+H2O | | | 6.00E+08 | 1.3 | 0.0 |
| 39. | H2+M=H+H+M | | | 2.23E+12 | 0.5 | 92600.0 |
| | H2O | Enhanced by | 6.000E+00 | | | |
| | H | Enhanced by | 2.000E+00 | | | |
| | H2 | Enhanced by | 3.000E+00 | | | |
| 40. | O2+M=O+O+M | | | 1.85E+11 | 0.5 | 95560.0 |
| 41. | H+OH+M=H2O+M | | | 7.50E+23 | -2.6 | 0.0 |
| | H2O | Enhanced by | 2.000E+01 | | | |
| 42. | H+HO2=H2+O2 | | | 2.50E+13 | 0.0 | 700.0 |
| 43. | HO2+HO2=H2O2+O2 | | | 2.00E+12 | 0.0 | 0.0 |
| 44. | H2O2+M=OH+OH+M | | | 1.30E+17 | 0.0 | 45500.0 |
| 45. | H2O2+H=HO2+H2 | | | 1.60E+12 | 0.0 | 3800.0 |
| 46. | H2O2+OH=H2O+HO2 | | | 1.00E+13 | 0.0 | 1800.0 |

NOTE: A units mole-cm-sec-K, E units cal/mole

Table 1: Smooke-46 chemical mechanism for methane-air combustion.

This is the set of governing equations solved by Sandia's PREMIX code, taking the stream-tube area $A$ to be constant in space and equal to unity. Note that, as written above, $\dot{m}$ has a physical meaning similar to $V$ in the strained flame governing equations. The transverse momentum equation (30, however, has dropped out of the system. Equations 36–38 describe an ODE boundary-value problem, in which the burning rate $\dot{m}$ for the freely propagating flame is an eigenvalue and must be found as part of the solution. The strain field in Equations 28–31, by contrast, ensures that $V$ varies normal to the flame.

# 4 Numerical Solution

The evolution of the unsteady strained flame is governed by a set of one-dimensional partial differential equations, as derived in the previous section. Difficulty in the numerical solution of these PDEs stems from the inclusion of detailed chemistry. The chemical source terms $\dot{w}_k$ are composites of elementary rate expressions, where each rate expression is a strong, usually exponential, function of temperature. Together, these source terms introduce a wide range of time scales to the governing equations, extending to the order of nanoseconds. Overcoming the stiffness of detailed chemistry has been a long-standing challenge in numerical combustion simulation, one which here necessitated the use of several novel numerical tools.

## 4.1 Finite-difference Discretization

The governing equations of the strained flame (Equations 28–31) must be integrated implicitly, as required by stiffness. A first-order, backward Euler formulation is used, discretizing the time derivatives as follows:

$$\rho\frac{\partial T}{\partial t} = \mathcal{F}\ (T, V, Y_k, \dots) \rightsquigarrow \rho^{n+1}\frac{T^{n+1} - T^n}{\Delta t} = \mathcal{F}\left(T^{n+1}, V^{n+1}, Y_k^{n+1}, \dots\right) \tag{39}$$

Here and in all subsequent discretizations, the superscript indicates the time layer; the time derivative in the energy equation is used as an example. In contrast to previous formulations [11], all the governing equations—energy conservation, momentum conservation, species conservation, and continuity—are solved simultaneously.

To prevent the formation of numerical instabilities, convective terms in the species, energy, and momentum conservation equations are discretized with a first-order upwind stencil. This stencil thus depends on the sign of the convective velocity:

$$V_j < 0 : \; V\frac{\partial T}{\partial y} \approx V_j \frac{T_{j+1} - T_j}{h_j} \tag{40}$$

$$V_j > 0 : \; V\frac{\partial T}{\partial y} \approx V_j \frac{T_j - T_{j-1}}{h_{j-1}} \tag{41}$$

Here, the subscript represents the point on the spatial grid, while $h$ is the grid spacing: $h_j \equiv y_{j+1} - y_j$. Diffusion terms are approximated to second-order accuracy. Again, the energy equation (29) is used as an example:

$$\frac{1}{C_p}\frac{\partial}{\partial y}\left(\lambda \frac{\partial T}{\partial y}\right) \approx \frac{1}{C_{p,j}} \frac{2}{y_{j+1} - y_{j-1}} \left( \lambda_{j+1/2} \frac{T_{j+1} - T_j}{h_j} - \lambda_{j-1/2} \frac{T_j - T_{j-1}}{h_{j-1}} \right) \tag{42}$$

The continuity equation (31) must be carefully discretized to avoid numerical oscillations in the mass flux profile. Without an appropriate stencil, oscillations are observed during translation of the flame with respect to the spatial grid; susceptibility to oscillations is high since the continuity equation is the only governing equation without physical dissipation. Thus, an upwind discretization is used. The term 'upwind' is written with some qualification, since the continuity equation is not properly a transport equation and $\partial V/\partial y$ is not a convective term per se. Nonetheless, taking the positive sign of $\partial V/\partial y$ to suggest a positive upwind velocity, the following discretization of the continuity equation adds dissipation of the appropriate sign:

$$\frac{\partial \rho}{\partial t} + \frac{\partial V}{\partial y} + \rho U \epsilon = 0 \; \rightsquigarrow$$

$$\frac{\rho_j^{n+1} - \rho_j^n}{\Delta t} \; + \; \frac{V_j^{n+1} - V_{j-1}^{n+1}}{h_{j-1}} + \rho_j^{n+1} U_j^{n+1} \epsilon^{n+1} = 0 \tag{43}$$

This discretization of $\partial V/\partial y$ prevents direct implementation of the stagnation point boundary condition $V(y = 0) = 0$. Instead, a boundary value on the mass flux $V$ must be chosen at $y = -\infty$. The boundary value is arbitrary provided that it is large enough for a flame to stabilize at a lower mass flux, since $V$ decreases in the direction of the stagnation point. If the

boundary value is too large, on the other hand, the flame (or for that matter, the stagnation point) may not fall within the computational domain. The boundary value can thus be set to any reasonable number based on the size of the computational domain and the strain rate. The solution to the problem matches the mass flux profile to the flame location, as reflected in the profiles of $T$, $Y_k$, $\rho$, and $U$. In computations with changing strain rate, the boundary condition on $V$ must be updated periodically, a process detailed in §4.4.2.

Spatial discretization of the governing equations is performed on a non-uniform adaptive grid. Non-uniform grid spacing permits a greater clustering of grid points in regions where spatial gradients are strong, thus ensuring adequate resolution through the reaction-diffusion zone; a fine grid resolution tends to offset the fact that convective discretizations are only first-order accurate. Grid points are more sparsely spaced away from the flame, in regions of constant gradient. This non-uniform allocation of grid points provides accuracy without sacrificing computational efficiency.

As the flame translates in space or as its internal structure changes, adaptivity of the grid becomes important. Several criteria for grid resolution are enforced at the end of each time step, and where these criteria are not met, grid points are added or removed appropriately. First, the gradient of each scalar must be adequately resolved, relative to its range:

$$|\varphi_{j+1} - \varphi_j| < \alpha \left| \max_j \varphi - \min_j \varphi \right| \tag{44}$$

Here, $\varphi$ is chosen to include all the dependent variables ($T$, $Y_k$, $U$, and $V$) as well as each chemical source term $\dot{w}_k$; $\alpha$ is a tolerance parameter on the order of 0.1. Second gradients are subject to an analogous criterion:

$$\left| \left( \frac{\partial \varphi}{\partial y} \right)_{j+1} - \left( \frac{\partial \varphi}{\partial y} \right)_j \right| < \beta \left| \max_j \left( \frac{\partial \varphi}{\partial y} \right) - \min_j \left( \frac{\partial \varphi}{\partial y} \right) \right| \tag{45}$$

The final criterion forces the grid to be relatively uniform:

$$\gamma < \frac{h_j}{h_{j-1}} < 1/\gamma \tag{46}$$

Linear interpolation is used to calculate the value of the dependent variables at any new grid point.

At the end of each time step, the thickness of the flame's temperature profile is compared to the distance from each boundary to the nearest end

of the flame. If the flame comes to within two thermal thicknesses of the boundary, the computational domain is extended and values of each dependent variable are extrapolated to the new grid points. This criterion for grid extension is most important on the products side of a premixed flame, where the boundary conditions expect the mixture to have reached equilibrium, with all spatial gradients in $T$, $Y_k$, and $U$ falling to zero.

The time step for integration is constant, typically chosen in the range of 1–10 $\mu$s. At each time step, discretization reduces the governing PDEs to a set of nonlinear algebraic equations. The nonlinear system can be written as:

$$F(x) = 0, \quad F : \mathbb{R}^n \to \mathbb{R}^n \tag{47}$$

The output of the function $F$ is a column vector containing residuals of the discretized governing equations, while $x$ is the solution vector, containing profiles of each fundamental variable, $Y_k$, $T$, $U$, and $V$. The organization of the values within these column vectors is important, as it affects the bandwidth and structure of the Jacobian matrix, $F'_{ij} \equiv \partial F_i / \partial x_j$. Variables and residuals are thus grouped by grid point:

$$x = \begin{pmatrix} \vdots \\ Y_{(k=1),\,j} \cdots Y_{(k=K),\,j} \\ T_j \\ U_j \\ V_j \\ Y_{(k=1),\,j+1} \cdots Y_{(k=K),\,j+1} \\ T_{j+1} \\ U_{j+1} \\ V_{j+1} \\ \vdots \end{pmatrix} \tag{48}$$

$$F = \begin{pmatrix} \vdots \\ res_{spec(k=1),\,j} \ldots res_{spec(k=K),\,j} \\ res_{energy,\,j} \\ res_{mntm,\,j} \\ res_{cont,\,j} \\ res_{spec(k=1),\,j+1} \ldots res_{spec(k=K),\,j+1} \\ res_{energy,\,j+1} \\ res_{mntm,\,j+1} \\ res_{cont,\,j+1} \\ \vdots \end{pmatrix} \tag{49}$$

Here, $res_{energy,\,j}$ denotes the residual of the energy conservation equation discretized at grid point $j$; similar notation applies to the residuals of the conservation equation for species $k$ ($res_{spec(k),\,j}$), the momentum conservation equation ($res_{mntm,\,j}$), and the continuity equation ($res_{cont,\,j}$).

## 4.2 Inexact Newton Method

The most basic means of solving nonlinear equations is Newton's method. The solution is approached by iteration; a linear model is constructed at each trial solution to yield a step, or correction, towards a more accurate solution. For a nonlinear system of equations as in (47), the Newton step is given by

$$F'(x_k)s_k = -F(x_k) \tag{50}$$

where $x_k$ is the current approximate solution and $s_k$ is the step towards the next solution, $x_{k+1} = x_k + s_k$. As defined in the previous section, $F'(x_k)$ is the Jacobian computed at $x_k$, $F'_{ij} \equiv \partial F_i / \partial x_j$.

Newton's method converges quadratically in the neighborhood of a solution, a desirable feature when the trial solution is accurate. Far away from a solution, however, Newton's method is not robust, converging poorly if at all. Directly solving Equation 50 for a Newton step—a computationally expensive undertaking in any large scale problem—is not effective in this situation. Replacing the Newton condition of Equation 50, we introduce the more flexible *inexact Newton condition*:

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\| \tag{51}$$

This modification of the Newton equation is at the core of the Inexact Newton Backtrackng (INB) method, developed by Eisenstat and Walker [3].

The key parameter in Equation 51 is $\eta_k$, known as the forcing term. The inexact Newton condition essentially restates the exact Newton conditon from the perspective of an iterative linear solver. An iterative method is used to find an approximate solution to (50), and $\eta_k$ specifies the tolerance to which this solution ($s_k$) is found. For $\eta_k$ equal to zero, the Newton condition is solved exactly, reducing Equation 51 to Equation 50, while as $\eta_k$ approaches 1, computational effort falls to zero.

Proper specification of $\eta_k$ can vastly enhance efficiency and convergence of the nonlinear solver. Far away from a solution, where the linear model of Equation 50 may poorly describe the nonlinear surface of the function $F$, it makes little sense to calculate each Newton step exactly. Doing so is computationally expensive and may result in little or no progress toward a solution; indeed, a less accurate solution of (50) may be more effective in reducing $\|F\|$. The forcing term is thus chosen close to 1. If the linear model becomes accurate, however, a precise Newton step is likely to provide fast reduction in $\|F\|$; the forcing term is thus chosen close to zero. Explicit formulas governing the choice of $\eta_k$ at each iteration will be given, but first let us outline the entire Inexact Newton Backtracking algorithm:

**Algorithm 4.1 (Inexact Newton Backtracking Method)**
*Let* $x_o,\ \eta_{max} \in [0,1),\ t \in (0,1)$, *and* $0 < \theta_{min} < \theta_{max} < 1$ *be given.*
*For* $k = 0, 1, \ldots,$ until convergence*:*
*Choose initial* $\eta_k \in [0, \eta_{max}]$ *and* $s_k$ *such that*
$\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\|$.
*While* $\|F(x_k + s_k)\| > [1 - t(1 - \eta_k)] \|F(x_k)\|$ *do:*
*Choose* $\theta \in [\theta_{min}, \theta_{max}]$.
*Update* $s_k \longleftarrow \theta s_k$ *and* $\eta_k \longleftarrow 1 - \theta(1 - \eta_k)$.
*Set* $x_{k+1} = x_k + s_k$.

This algorithm augments the inexact Newton condition with a globalization known as *safeguarded backtracking*. Sufficient reduction in the norm of $F$ is enforced at the 'while' loop; if the step $s_k$ of the inexact Newton condition does not sufficiently reduce $\|F\|$, the step is reduced by a scalar factor $\theta$, essentially 'backtracking' along the search direction. Backtracking continues until the condition on $\|F\|$ is met, for in a sufficiently small neighborhood of the trial solution $x_k$, the linear model must indicate the correct downward path; the Newton equation is consistent. The backtracking process is considered 'safeguarded' because limits are placed on the choice of $\theta$, and it is

known as a 'globalization' because it vastly improves the domain of convergence of Newton's method. Indeed, in exact arithmetic, the INB algorithm is guaranteed to converge from an arbitary initial guess unless $x_k$ has no limit points or $x_k$ only has limit points at which $F'$ is singular [3].

As described above, the forcing term $\eta_k$ is updated at each iteration to reflect the accuracy of the local linear model. Two schemes that provide fast local convergence while minimizing oversolving are given by Pernice and Walker [10]. For the first, we select $\eta_o \in [0, 1)$ and choose subsequent values of $\eta_k$ as follows:

$$\eta_k = \frac{|\|F(x_k)\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\||}{\|F(x_{k-1})\|}, \quad k = 1, 2, \ldots \tag{52}$$

The second scheme updates $\eta_k$ based on the norm reduction of the previous Newton iteration, a more indirect measure of the accuracy of the local linear model:

$$\eta_k = \gamma \left( \frac{\|F(x_k)\|}{\|F(x_{k-1})\|} \right)^{\alpha}, \quad k = 1, 2, \ldots \tag{53}$$

The scaling $\gamma$ is chosen in $[0, 1]$ and the exponent $\alpha$ is chosen in $(1, 2]$.

In our implementation of the INB method, convergence is declared when $\|F(x_k)\|_2$ or $\|F(x_k)\|_\infty$ falls below a specified tolerance, usually $10^{-9}$. Equation 52 is typically used to specify the forcing terms, with $\eta_o = 0.5$. The remaining parameters are adjusted arbitrarily to enhance convergence.

## 4.3 Krylov Subspace Iterative Solver

Iterative solution of the Newton equation (50) is a rather large scale problem. For a typical grid of 150 points and a detailed mechanism of 16 species, the Jacobian matrix $F'$ is of dimension $m \approx 3000$. Given that the Newton equation is solved repeatedly in the course of a single time step, an efficient linear iterative method is essential to the performance of the code. Here we turn to a modern class of iterative methods known as Krylov subspace methods, and we develop preconditioners to accelerate their convergence.

### 4.3.1 BiCGSTAB

Krylov subspace methods are based on the idea of projecting an $m$-dimensional problem onto a lower-dimensional Krylov subspace [19]. For a linear system

$Ax = b$, a Krylov subspace of dimension $n$ is defined as follows:

$$\mathcal{K}_n = \left\langle b, Ab, \ldots, A^{n-1}b \right\rangle \tag{54}$$

At each iteration, the trial solution $x_n$ is chosen from the subspace $\mathcal{K}_n$; iteration increases the size of the Krylov subspace and thus the accuracy of the solution. GMRES (*G*eneralized *M*inimum *Res*idual) is probably the simplest and best-known Krylov method, and its workings are illustrative: At each step, Arnoldi iterations construct an orthonormal basis for $\mathcal{K}_n$. A least-squares approximation then finds the $x_n \in \mathcal{K}_n$ that minimizes the norm of the residual $b - Ax$.

In our implementation of the INB algorithm, we use a variant Krylov subspace method known as BiCGSTAB—*Bi*-*C*onjugate *G*radients *Stab*ilized. The details of this method are more complex, and are left better-described elsewhere [19, 5, 16]. BiCGSTAB is based on Lanczos iteration, which, in contrast to the Arnoldi iterations of GMRES, uses a three-term recurrence relation to construct an orthonormal basis for $\mathcal{K}_n$. BiCGSTAB does not strictly minimize the residual norm at each iteration, and its convergence is not monotonic. However, it requires less storage space, and in our experience, provides greater numerical stability.

An important attribute of Krylov subspace methods is that they do not operate directly on the matrix $A$. All that is required is matrix-vector products $Ax$, making it simple to design procedures that exploit the sparsity of $A$. For a matrix of dimension $m$ with only $\nu$ nonzero entries per row, one can easily compute $Ax$ in $O(\nu m)$ rather than $O(m^2)$ operations. In the context of the INB algorithm, our matrix $A$ is the Jacobian, $F'$, and a finite-difference formula is used to calculate the product of $F'$ with an arbitrary vector $v$:

$$F'(x_k)v \approx \frac{1}{2\delta}\left[F(x_k + \delta v) - F(x_k - \delta v)\right] \tag{55}$$

The scalar perturbation $\delta$ is chosen as follows:

$$\delta = \frac{\left[(1 + \|x_k\|)\,\epsilon_{mach}\right]^{1/3}}{\|v\|} \tag{56}$$

where $\epsilon_{mach}$ denotes machine epsilon.

### 4.3.2 ILUTP Preconditioning

The inclusion of detailed chemistry in the governing equations (28–31) creates scalar entries of the Jacobian that span several orders of magnitude relative

to one another. Stiffness on the continuous scale thus manifests itself as an extremely ill-conditioned linear algebra problem. Indeed, the condition number $\kappa(F')$ is typically on the order of $10^9$. The convergence properties of BiCGSTAB and other Krylov methods are not completely understood, but it is known that they hinge on the eigenvalue or singular value distribution of the matrix. As a rule of thumb, closely and uniformly distributed eigenvalues produce fast convergence, and an ill-conditioned matrix has a spectrum that is quite the opposite. The question then arises of how to improve the properties of our Jacobian matrix so that BiCGSTAB converges rapidly? In principle, a Krylov subspace method must converge in $n \ll m$ steps if it is to be useful. The answer lies in preconditioning—subjecting the matrix to operations that improve its eigenvalue spectrum from the perspective of the Krylov subspace solver.

The basic procedure of preconditioning is outlined as follows.[1] The linear system $Ax = b$ is replaced by two equations:

$$AM^{-1}y = b \tag{57}$$

$$Mx = y \tag{58}$$

The Krylov subspace solver is applied to the first equation, while $x$ is calculated from the second. To accelerate convergence of the Krylov solver, $M$ is chosen so that $AM^{-1}$ is not too far from normal and its eigenvalues are clustered; $AM^{-1}$ is the well-behaved matrix that the Krylov solver "sees." On the other hand, $M$ must also be chosen so that Equation 58 is simple to solve. The preconditioner $M$ must lie somewhere between $A$ and $I$ on a spectrum of approximation; choosing $M = A$ or $M = I$ would render Equation 58 or Equation 57, respectively, equivalent to the original ill-conditioned problem.

The right preconditioner implemented here is based on the idea of *incomplete LU factorization*, ILU for short. An exact LU factorization of a sparse matrix like $F'$ produces two dense matrices $L$ and $U$; the replacement of zero entries in $F'$ with nonzero entries in $L$ and $U$ is known as fill-in. Suppose, however, that fill-in were prevented, that $\tilde{L}$ and $\tilde{U}$ could have nonzero entries only where $F'$ was nonzero. Then, $\tilde{L}$ and $\tilde{U}$ are the incomplete LU factors

[1] Actually, this is just *right* preconditioning. A preconditioner may also be applied on the left ($M^{-1}Ax = M^{-1}b$), or on both sides, to preserve hermitian positive definiteness ($\left[C^{-1}AC^{-*}\right] C^* = C^{-1}b$).

of $F'$, and

$$M = \tilde{L}\tilde{U} \approx F' \tag{59}$$

becomes a useful preconditioner.

Several refinements of ILU factorization have been developed by Saad [16] and published in his SPARSKIT sparse matrix toolkit. Using this package, we implement an ILUTP preconditioner, a more accurate ILU factorization that allows some fill-in. ILUTP preconditioning regulates fill-in in $L$ and $U$ on the basis of two parameters: one, called *lfil*, is the absolute number of fill-in elements that are allowed in each row of $L$ or $U$; the second, called *droptol* is threshold value below which a potential fill-in element is set to zero. The ILUTP approach also provides for pivoting, in which the original matrix is permuted to insure diagonal dominance. Details of the ILUTP algorithm are more fully described elsewhere [16].

To improve the stability of the factorization process, we scale each column of the matrix by its 1-norm prior to computing the preconditioner. The entire preconditioning process can now be summarized as follows, where $A$ is understood to be the Jacobian $F'$:

$$ASP \approx LU \equiv M \tag{60}$$

$$Ax = b \longrightarrow$$

$$ASP(LU)^{-1}y = b \tag{61}$$

$$S^{-1}P^{-1}LUx = y \tag{62}$$

$S$ is a diagonal matrix containing the column-scaling parameters and $P$ is the permutation matrix. The Jacobian is scaled and permuted, then factored into $L$ and $U$. The original linear problem then consists of solving (61) and (62). BiCGSTAB is applied to Equation 61; in Equation 62, $S$ and $P$ are inverted by inspection, and back-substitution is used to invert $L$ and $U$.

Implementing this preconditioning scheme in the context of the INB algorithm is relatively simple. The preconditioning factors $L$, $U$, $S$, and $P$ are computed and stored at the start of each Newton iteration. The BiCGSTAB algorithm requires only matrix-vector products, specifically the product of $ASP(LU)^{-1}$ with arbitrary vectors $z$. Equation 62 is thus used to compute $v = SP(LU)^{-1}z$, and the output $v$ is sent to Equation 55. Applying the

preconditioner thus consists of repeatedly inverting Equation 62. In practice, the preconditioning factors need not be computed at the start of *each* Newton iteration, as the Jacobian may not change significantly, and preconditioning is, after all, approximate. The optimal frequency for updating the preconditioner rests on a balance of computational times: the additional time required to recalculate $L$, $U$, $S$, and $P$ versus the time this saves by speeding the convergence of BiCGSTAB.

In contrast to the matrix-free formulation of the Krylov subspace solver, computing the preconditioner by the methods described in this section requires an explicit Jacobian. At each preconditioner update, entries of the block-tridiagonal Jacobian are constructed according to the definition, $F'(x_k)_{ij} \equiv \partial F(x_k)_i / \partial x_{k,j}$. Most of these derivatives can be computed analytically, but those involving chemical source terms are computed via finite-difference approximation:

$$\frac{\partial \dot{w}_k}{\partial Y_{k*}} = \frac{\dot{w}_k(T, Y_1, \ldots, Y_{k*} + \delta Y_{k*}, \ldots, Y_K) - \dot{w}_k(T, Y_1, \ldots, Y_{k*}, \ldots, Y_K)}{\delta Y_{k*}} \tag{63}$$

where $\delta$ is usually chosen around $10^{-3}$; here the derivative with respect to species mass fraction was chosen as an example. To compute derivatives of $\dot{w}_k$ analytically would entail decomposing the chemical rate expressions within CHEMKIN—a difficult undertaking—and would compromise the modularity of the code with respect to kinetic schemes. In addition, derivatives of the transport coefficients with respect to $T$ or $Y_k$ cannot be computed easily, since they involve polynomial approximations within CHEMKIN for $C_p$. These derivatives are omitted from the explicit Jacobian; the resulting inaccuracy is generally insignificant, since $L$ and $U$ are approximate factors.

## 4.4 Initialization Conditions

Despite the robust convergence properties of the INB method, it will not converge quickly from a completely arbitrary profile. At every time step, the Newton solver requires a starting point for its iterations, an initial guess $x_o$. Ingenuity in creating these initialization conditions can yield large computational savings. We analyze this process in two cases: creating a starting condition for the code at $t = 0$, and selecting the inital guess at each time step while the code is running, i.e., for $t > 0$.

### 4.4.1 Starting Case

When applying the elemental flame code to a new chemical mechanism or a new flame configuration, an initial solution vector is assembled to approximate the expected flame structure as closely as possible. This guess is both $x$ at the $n = 0$ time step and the initial guess for $x^{n=1}$. A reasonable procedure for creating such an initial guess is as follows: Hyperbolic tangent functions are used to approximate $T(y)$, $U(y)$, and the mass fraction profiles of major species. Gaussians are used to approximate the profiles of minor species. In a premixed flame, an equilibrium code such as STANJAN provides product-side boundary conditions on $T$ and $Y_k$, and in all cases, a steady-state version of the continuity equation (31) is integrated to find $V$. The thickness and height of each profile must be estimated based on knowledge of the kinetic mechanism and of typical flame structure at the desired strain rate.

Clearly, this ad hoc scheme cannnot predict the flame structure very closely. To create a more accurate starting condition for the elemental flame, the initial guess is subjected to a *continuation* scheme for gradually introducing the chemical source terms. Chemistry is the root of stiffness in the governing equations, and thus is the primary cause of difficult convergence. The continuation scheme places damping terms—arbitrary scalar factors $\theta$—in front of the chemical source terms in the energy and species equations (29 and 28), converting the elemental flame into a weakly-reacting or non-reacting mixture as follows:

$$\rho\frac{\partial Y_k}{\partial t} + V\frac{\partial Y_k}{\partial y} + \frac{\partial}{\partial y}\left(\rho Y_k \mathcal{V}_k\right) - \theta\dot{w}_k W_k = 0 \tag{64}$$

$$\rho\frac{\partial T}{\partial t} + V\frac{\partial T}{\partial y} - \frac{1}{c_p}\frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) + \frac{1}{c_p}\left(\sum_k^K \rho c_{p,k} Y_k \mathcal{V}_k\right)\frac{\partial T}{\partial y} + \frac{\theta}{c_p}\sum_k^K \dot{w}_k H_k = 0 \tag{65}$$

The initial guess is used to start the code in a weakly-reacting form, and as time integration proceeds, the damping terms $\theta$ are gradually increased to unity. Once the solution vector reaches steady-state, it becomes a physically accurate starting case.

The continuation scheme typically begins with damping terms on the order of $10^{-3}$ and brings them to 1.0 in about 10 ms of integration. Precise values depend on the stiffness of the kinetic scheme, the accuracy of the initial guess, and the flame time scale.

### 4.4.2 Mass Flux Projection Method

When $t > 0$, creating the initial guess for the solution vector at each time step is much simpler; the inital guess at $t = n + 1$ is just $x^n$. This procedure creates difficulty in the case of unsteady strain, however. Recall that to aid in discretization, the boundary condition on $V$ is imposed at $y = -\infty$. The strain rate $\epsilon$ can easily vary one or two orders of magnitude in a given simulation, however; such a change in strain, with a fixed boundary value on the mass flux through the flame, causes the flame to translate rapidly with respect to the grid. The flame nears the boundary of the computational domain, and successive regriddings become necessary. This outcome is not only cumbersome, but computationally taxing; when the flame translates quickly, $x^n$ is a poor guess for $x^{n+1}$.

To avoid this situation while retaining the boundary condition $V(y = -\infty)$, we implement a projection method to update the mass flux profile in the case of unsteady strain. At the start of time step $n + 1$, an initial guess for $V^{n+1}$ is obtained by integrating the continuity equation with $\epsilon^{n+1}$, $U^n$, and $\rho^n$:

$$\frac{\rho_j^n - \rho_j^{n-1}}{\Delta t} + \frac{V_j^{n+1,\,guess} - V_{j-1}^{n+1,\,guess}}{h_{j-1}} + \rho_j^n U_j^n \epsilon^{n+1} = 0 \qquad (66)$$

In one step, this projection updates the boundary value on $V$ at $y = -\infty$ and generates a new guess for $V^{n+1}$. Updating the boundary value on $V$ minimizes flame translation for fast convergence.

# 5 Flame Structure: Numerical Results

This section is essentially a "gallery" of numerical results designed to illustrate various aspects of strained flame structure and unsteady flame response.

## 5.1 Steady Strained Flame Structure—Profiles

Salient features in Figures 4–7:

- Flame thicknesses, distance from stagnation point.
- Zonal structure of flame (preheat, reaction, oxidation); relation to transport and hydrocarbon kinetics.

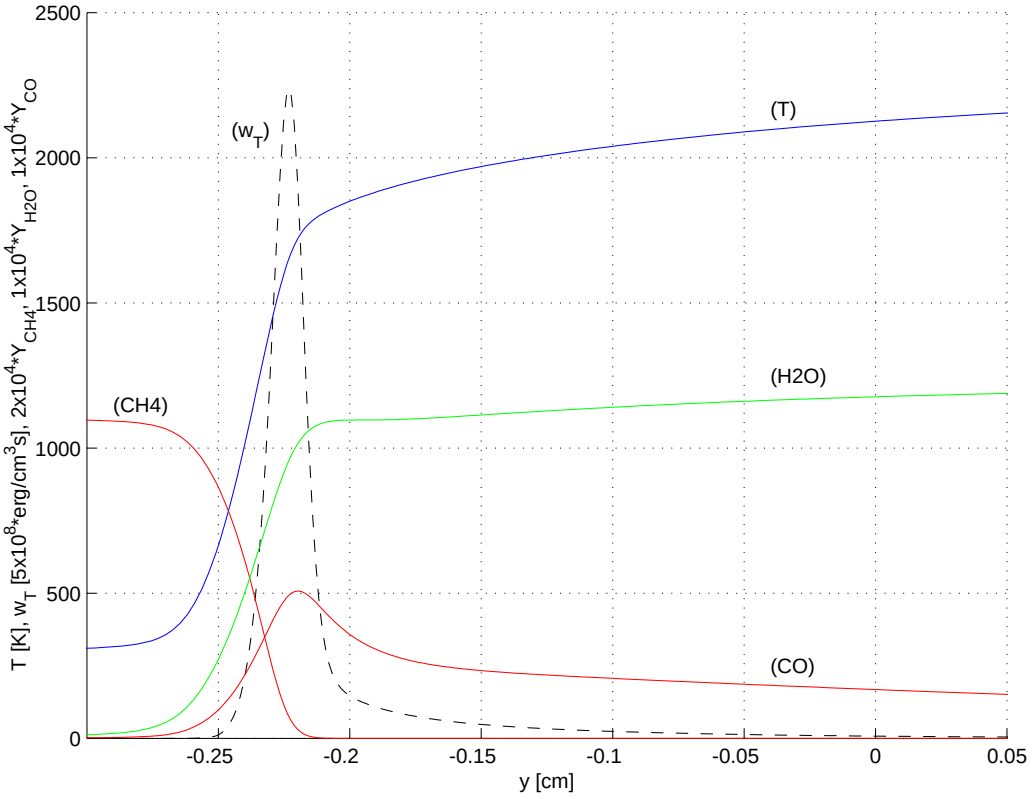


Figure 4: Steady strained flame, $\epsilon = 100$ s$^{-1}$, $\phi = 1.0$: major species profiles.

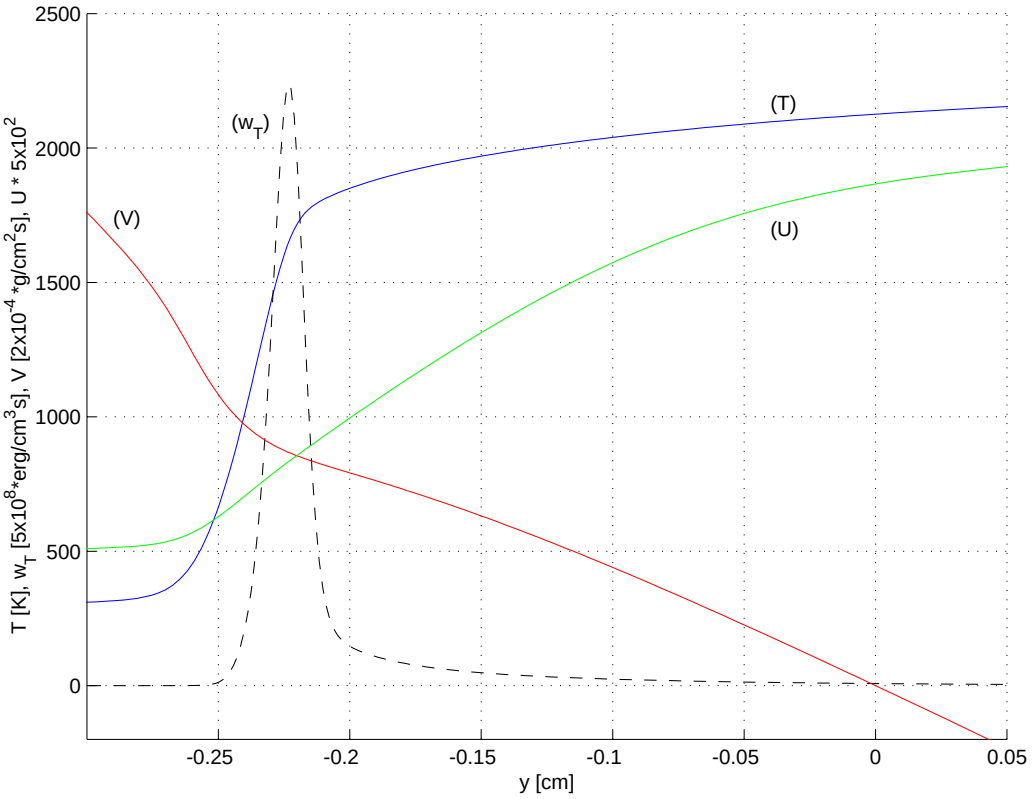


Figure 5: Steady strained flame, $\epsilon = 100$ s$^{-1}$, $\phi = 1.0$: mass flux and strain rate variation.

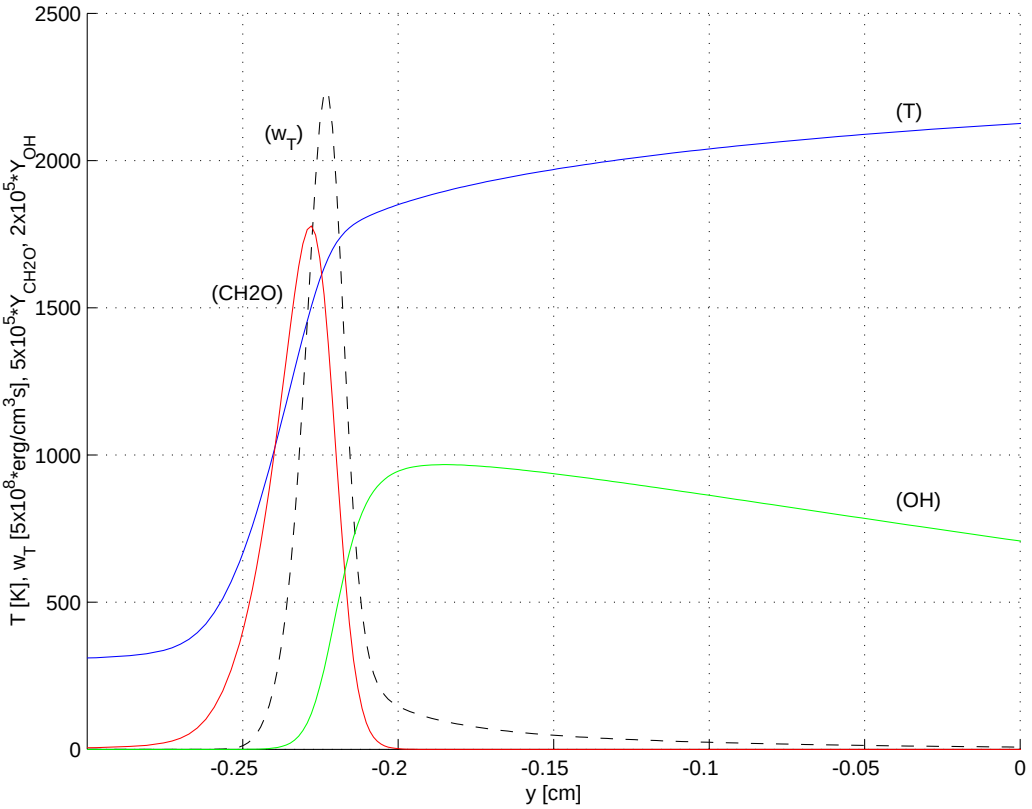


Figure 6: Steady strained flame, $\epsilon = 100\ \mathrm{s}^{-1}$, $\phi = 1.0$: minor species profiles.

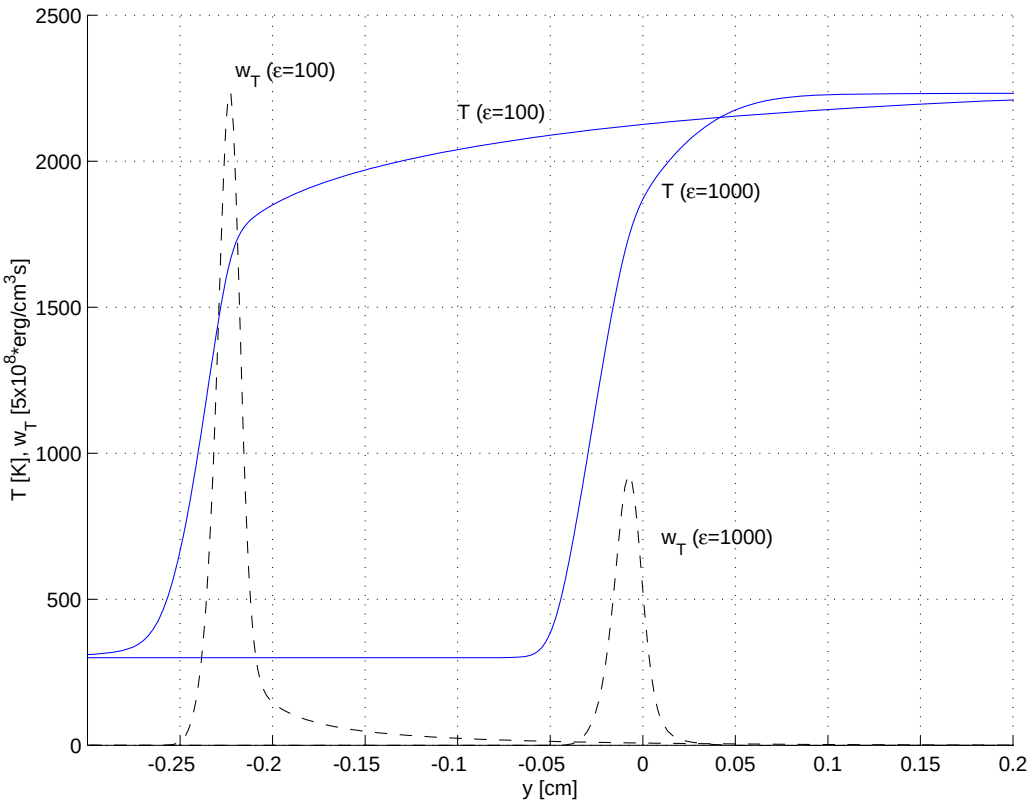


Figure 7: Comparison of two steady strained flames at $\phi = 1.0$; $\epsilon = 100$ s$^{-1}$, versus $\epsilon = 1000$ s$^{-1}$.

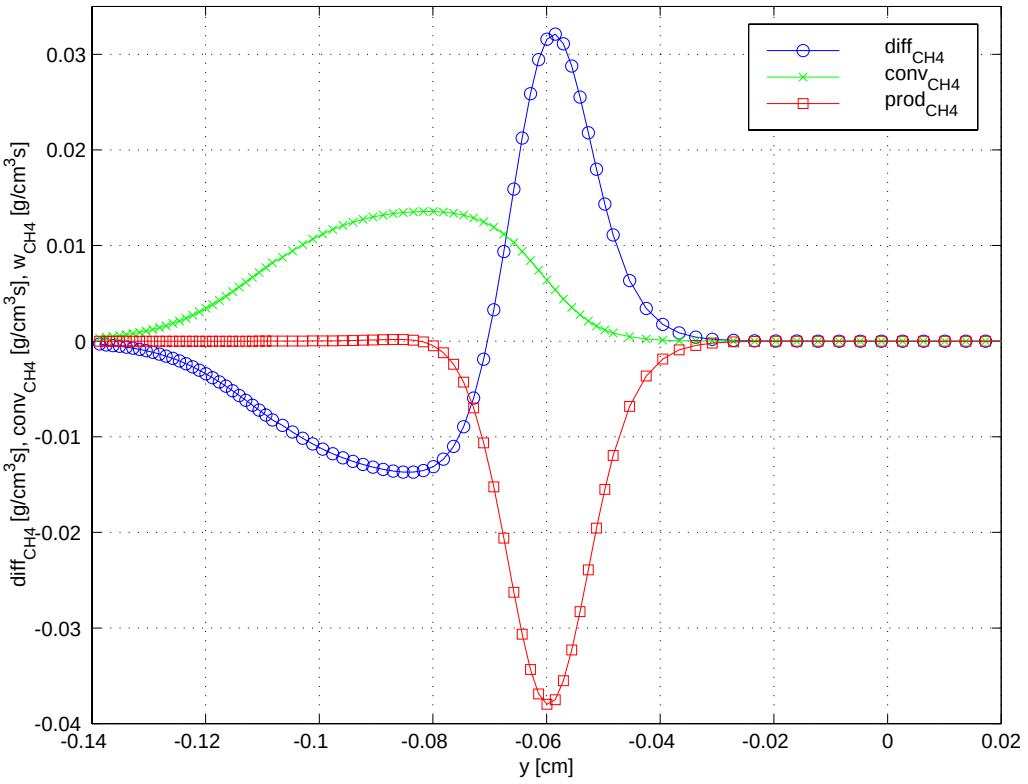


Figure 8: Net flux of CH4 due to convection, diffusion, and reaction; $\epsilon = 300$ $s^{-1}$, $\phi_u = 0.6$, $\phi_b = 1.0$.

- Variation (in value and slope) of $V$ across flame.
- Effect of strain on integrated heat release rate.

## 5.2 Steady Strained Flame—Fluxes

For the steady flame in Figure 8, note distinct balancing of convection-diffusion and reaction-diffusion (for this species).

## 5.3 Flame Reponse to Unsteady Strain

The details of Figure 9 are not the focus here; rather the two-dimensional simulation can be seen merely as the source of a "realistic" time-varying

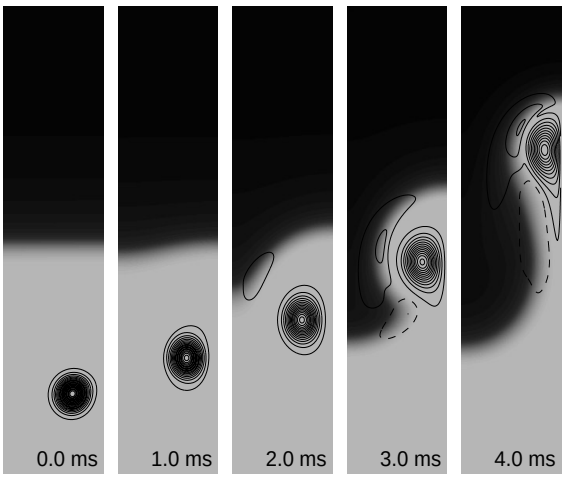


Figure 9: Direct numerical simulation of premixed flame interaction with a counter-rotating vortex pair. The vertical right-hand-side edge of each frame is the centerline of the vortex pair, only half of which is computed. Grayscale indicates gas temperature, with darker shading corresponding to burned combustion products. Solid/dashed contours delineate levels of positive/negative vorticity.

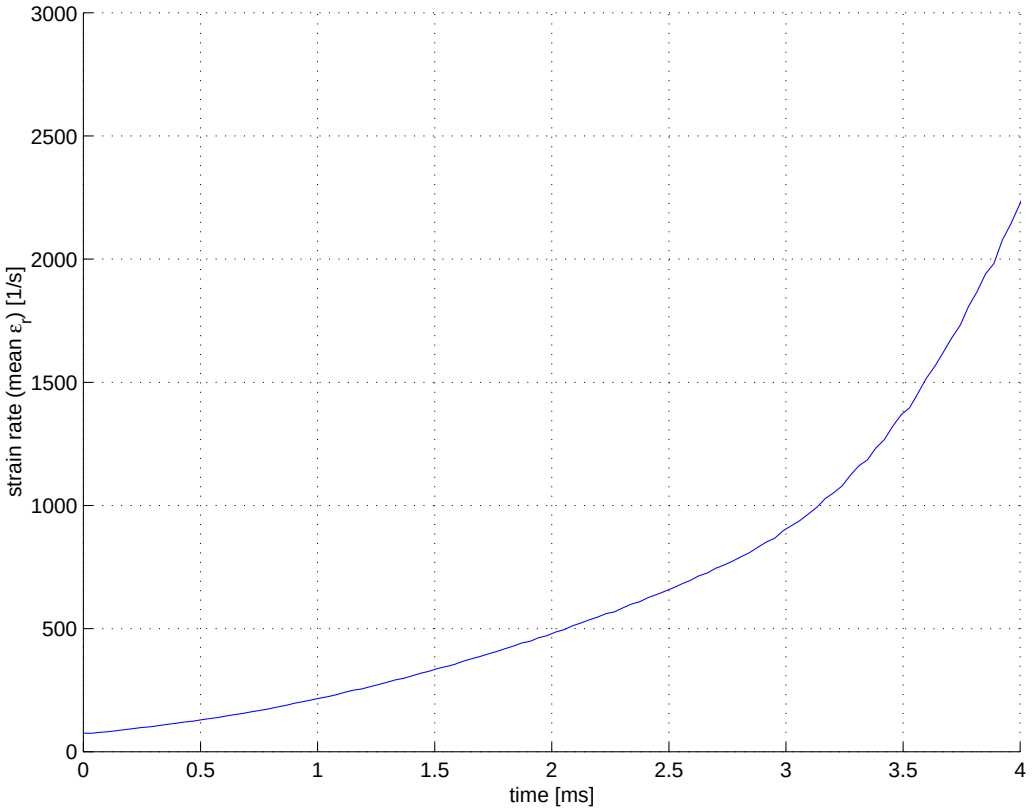


Figure 10: Strain history extracted from centerline of flame-vortex interaction.

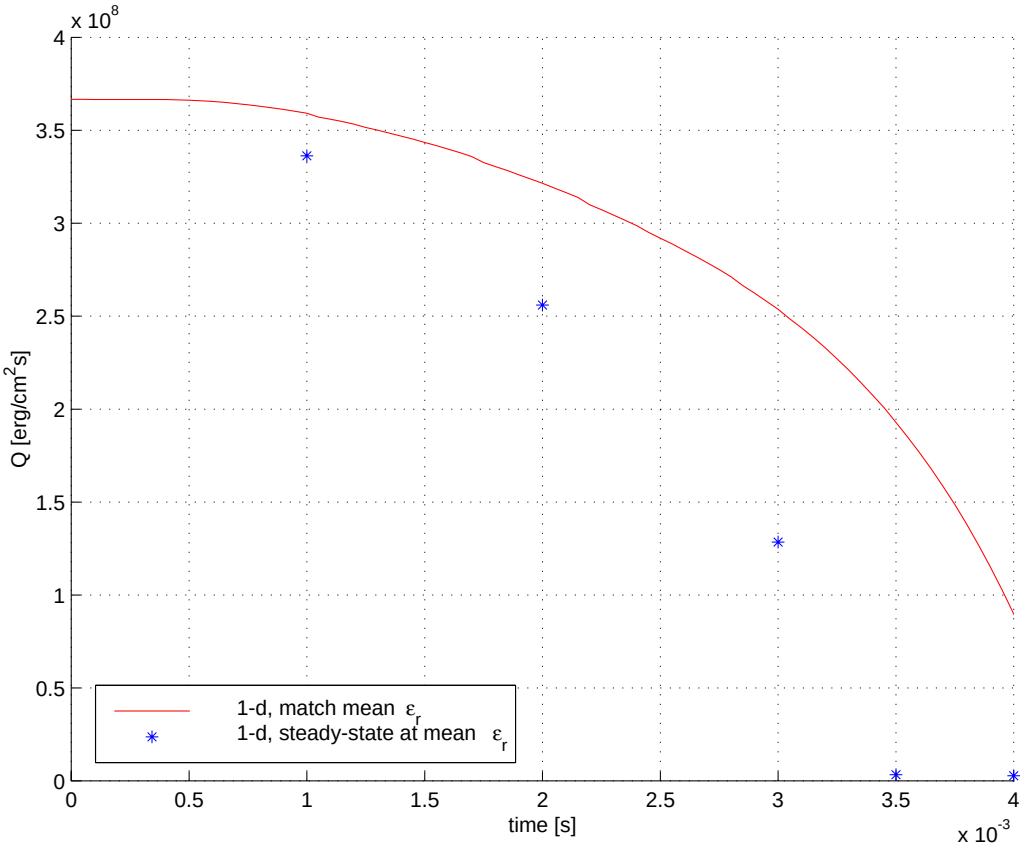


Figure 11: Integrated heat release rate history of the unsteadily-strained laminar flame. Asterisks correspond to the steady-state $\dot{Q}$ at the same average strain rate.

strain rate (Figure 10). As shown in Figure 11, the total heat release decreases with increasing strain, but the unsteady response differs markedly from the steady-state values.

# References


[1] N. Darabiha. Transient behaviour of laminar counterflow hydrogen-air diffusion flames with complex chemistry. *Combustion Science and Technology*, 86:163–181, 1992.

[2] F. N. Egolfopoulos. Dynamics and structure of unsteady, strained, laminar premixed flames. In *25th Symposium (International) on Combustion*, pages 1365–1373. The Combustion Institute, 1994.

[3] S. C. Eisenstat and H. F. Walker. Globally convergent inexact Newton methods. *SIAM Journal on Optimization*, 4:393–422, 1994.

[4] A. F. Ghoniem, M. C. Soteriou, O. M. Knio, and B. Cetegen. Effect of steady and periodic strain on unsteady flamelet combustion. In *24th Symposium (International) on Combustion*, pages 223–230. The Combustion Institute, 1992.

[5] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, third edition, 1996.

[6] H. G. Im, J. K. Bechtold, and C. K. Law. Counterflow diffusion flames with unsteady strain rates. *Combustion Science and Technology*, 106:345–361, 1995.

[7] R. J. Kee, J. A. Miller, G. H. Evans, and G. Dixon-Lewis. A computational model of the structure and extinction of strained opposed-flow premixed methane-air flames. In *22nd Symposium (International) on Combustion*, pages 1479–1494. The Combustion Institute, 1988.

[8] J. S. Kistler, C. J. Sung, T. G. Kreutz, C. K. Law, and M. Nishioka. Extinction of counterflow diffusion flames under velocity oscillations. In *26th Symposium (International) on Combustion*, pages 113–120. The Combustion Institute, 1996.

[9] 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 Laboratories, February 1998.

[10] M. Pernice and H. F. Walker. NITSOL: A Newton iterative solver for nonlinear systems. *SIAM Journal on Scientific Computing*, 19:302–318, 1998.

[11] C. Petrov. *Numerical Simulation of Reacting Flows with Complex Chemistry Using Flame Embedding*. Ph.D. thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, February 1997.

[12] C. Petrov and A. Ghoniem. An unsteady strained flame model for turbulent combustion simulations. In *32nd Aerospace Sciences Meeting and Exhibit*, number AIAA-94-0776. AIAA, January 1994.

[13] C. Petrov and A. Ghoniem. The transient response of strained laminar-premixed flames. *Combustion and Flame*, 102:401–417, 1995.

[14] C. Petrov and A. Ghoniem. An unsteady strained flame model with multi-step chemical kinetics for turbulent combustion simulations. In *33rd Aerospace Sciences Meeting and Exhibit*, number AIAA-95-0380. AIAA, January 1995.

[15] C. Petrov and A. Ghoniem. A uniform strain model of elemental flames in turbulent combustion simulations. *Combustion and Flame*, 111:47–64, 1997.

[16] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, Boston, 1996.

[17] M. D. Smooke, editor. *Reduced Kinetic Mechanisms and Asymptotic Approximations for Methane-Air Flames*. Lecture Notes in Physics. Springer-Verlag, Berlin, 1990.

[18] G. Stahl and J. Warnatz. Numerical investigation of time-dependent properties and extinction of strained methane- and propane-air flamelets. *Combustion and Flame*, 85:285–299, 1991.

[19] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[20] F. A. Williams. *Combustion Theory*. Addison-Wesley, Reading, MA, second edition, 1985.