Naval Research Laboratory
Washington, DC 20375-5000

NRL Memorandum Report 6555

FLIC — A Detailed, Two-Dimensional Flame Model

G. PATNAIK,* K.J. LASKEY,** K. KAILASANATH, E.S. ORAN AND T.A. BRUN†

Laboratory for Computational Physics and Fluid Dynamics

**Berkeley Research Associates*
Springfield, VA

***Carnegie-Mellon University*
Pittsburgh, PA

†Harvard University
Cambridge, MA

September 27, 1989

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 6555	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Research Laboratory 6b OFFICE SYMBOL (If applicable) Code 4410	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000	7b ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION NASA 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Cleveland, OH 44135	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: NASA PROJECT NO: NASA TASK NO: C-80001-G WORK UNIT ACCESSION NO:

11 TITLE (Include Security Classification)
FLIC - A Detailed, Two-Dimensional Flame Model

12. PERSONAL AUTHOR(S)
Patnaik,* G., Laskey,** K.J., Kailasanath, K., Oran, E. and Burn,+ T.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 10/86 TO present	1989 September 27	42

16 SUPPLEMENTARY NOTATION
*Berkeley Research Associates, Springfield, VA **Carnegie-Mellon University, Pittsburg, PA +Harvard University, Cambridge, MA

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Flames Numerical simulations
Flammability limits
Flame ignition

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes ***FLIC***, a detailed, two-dimensional model for flames. In order to describe a flame in enough detail to simulate its initiation, propagation, and extinction, ***FLIC*** combines algorithms for subsonic convective transport with buoyancy, detailed chemical reaction processes, and diffusive transport processes such as molecular diffusion, thermal conduction, and viscosity. Several new numerical techniques had to be developed specifically for multi-dimensional flame modelling,

(Continues)

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Kazhikathra Kailasanath	22b TELEPHONE (Include Area Code) (202) 767-2402 22c OFFICE SYMBOL Code 4410

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

19. ABSTRACTS (Continued)

and these are highlighted in this report. The numerical implementation of the flame model is described here, and a typical application is provided for illustration. ***FLIC*** is currently the state-of-the-art in detailed two-dimensional, time dependent flame models.

Contents

FLIC — A DETAILED, TWO-DIMENSIONAL FLAME MODEL

1. Introduction

This report describes the new computer code ***FL***ame with *I*mplicit *C*onvection (***FLIC***), a two-dimensional time-dependent program developed specifically to compute and study the behavior of flames and other subsonic chemically reactive flows. In order to describe a flame in enough detail to simulate its initiation, propagation, and extinction, ***FLIC*** combines algorithms for subsonic convective transport with buoyancy, detailed chemical reaction processes, and diffusive transport processes such as molecular diffusion, thermal conduction, and viscosity. Currently, we have not included algorithms for radiation transport or thermal diffusion, although these important processes can, in principle, be added in the same modular fashion as those physical processes that have been included.

FLIC solves the reactive-flow conservation equations for density, ρ, momentum, $\rho\vec{V}$, energy, E, and number densities of the individual species, n_k, $k = 1, \ldots, n_{sp}$, according to:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{\mathbf{V}}\right) = 0, \tag{1.1}$$

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot \left(\rho \vec{V} \vec{\mathbf{V}}\right) = -\nabla P + \vec{F} - \nabla \times \mu \nabla \times \vec{V} + \nabla \left(\frac{1}{3}\mu \nabla \cdot \vec{V}\right), \tag{1.2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(E \vec{\mathbf{V}}\right) = -\nabla \cdot \left(P \vec{\mathbf{V}}\right) + \nabla \cdot (\kappa \nabla T) - \sum_{k=1}^{n_{sp}} \nabla \cdot \left(n_k h_k \vec{V}_k\right) + \sum_{r=1}^{n_r} Q_r, \tag{1.3}$$

$$\frac{\partial n_k}{\partial t} + \nabla \cdot \left(n_k \vec{\mathbf{V}}\right) = -\nabla \cdot \left(n_k \vec{V}_k\right) + w_k. \tag{1.4}$$

Here $\vec{V}$ is the fluid velocity, P is the pressure, μ is the coefficient of shear viscosity, $\vec{F}$ is a body force, κ is the thermal conductivity of the mixture of gases, h_k is the enthalpy of species k, $\vec{V}_k$ is the diffusion velocity of species k, Q_r is the heat

Manuscript approved July 26, 1989.

an explicit method; so that the small timesteps it requires makes it inefficient for low-speed flows. The BIC-FCT method [7] was developed specifically to combine the accuracy of FCT with the efficiency of an implicit method for low-speed flows. BIC-FCT allows timesteps up to a hundred times larger than FCT and yet the calculation time of one BIC-FCT timestep is approximately equal to the calculation time of one timestep in the standard FCT module. Section 2 describes BIC-FCT in detail.

To solve the detailed combustion equations for hydrogen-oxygen chemistry involves solving coupled nonlinear ordinary differential equations for eight species and 48 chemical reactions representing the conversion of chemical species and chemical energy release into the system. This is the most expensive part of a reactive flow calculation because it requires integrating the set of equations at each computational cell for each timestep. The characteristic times of these differential equations vary by orders of magnitude, resulting in a set of very "stiff" equations. Then, because the cost of the calculation is approximately linear with the number of computational cells, the computational cost can be extreme in multidimensional computations. ***FLIC*** handles the cost of integrating ODEs in two ways: one, by not integrating the chemical reaction equations where there is nothing or essentially nothing happening, the other is by optimizing the integration procedure. We are using the CHEMEQ [8,9] method to solve stiff sets of ordinary differential equations, but in the TBA implementation that is fully optimized for the CRAY X-MP computer. TBA, described in section 3, allows speeds of up to 50 percent over VSAIM, the multidimensional implementation of CHEMEQ used to date.

Thermo-physical properties of the individual species and the mixture are required throughout the computation. These properties are modelled with high-order curve fits to values derived from more accurate calculations. The individual properties are combined where needed to obtain mixture properties through mixing rules. This simplified method is highy efficient yet sufficiently accurate. Section 4 describes the numerical solution of the diffusive transport processes.

The submodels representing the various physical processes are in independent modules that are coupled together. Several modifications have been made to the usual timestep splitting method in order to increase the stability limits and improve the efficiency of ***FLIC***. Section 5 describes the details of these improvements.

1.2 Comparison with FLAME1D

FLAME1D is a one-dimensional program used extensively to study the properties of the ignition and extinction of hydrogen flames [10]. It is useful to compare ***FLIC*** to FLAME1D because some FLAME1D algorithms are not obviously extendable to multidimensions and others are simply too expensive. The more important differences between these codes are described here.

1. ***FLIC*** uses an Eulerian representation of the convective transport instead of a Lagrangian representation. Specifically, ADINC, the one-dimensional Lagrangian algorithm [11] used in FLAME1D, is replaced by BIC-FCT [7], a very accurate multidimensional implicit Eulerian algorithm. A Lagrangian formulation, though preferable, is exceedingly difficult in multidimensions. However, unlike ADINC, BIC-FCT can be readily used to describe two-dimensional or three-dimensional flows.

2. Although the basic CHEMEQ algorithm is used in both codes, the VSAIM implementation used in FLAME1D was replaced by the TBA implementation. This algorithm is optimized for the CRAY X-MP, and can be retrofitted into CRAY versions of FLAME1D.

3. Another expensive part of the flame calculation is determining the amount that individual species diffuse. In FLAME1D, we used an iterative matrix expansion algorithm [12] that produces the diffusion flux of a species to arbitrary order, although we generally used it only up to second order. In ***FLIC***, we use a simpler technique which first evaluates a Fickian flux and then makes a correction. This approach is equivalent to the first order of the matrix expansion, cannot be made higher order, but it is computationally less intensive and certainly adequate for the flame problems treated to date.

4. In FLAME1D, thermal conduction is computed directly from expressions for the individual thermal conductivities of the species derived from molecular theory. In ***FLIC***, we use curve fits and mixture rules which has been benchmarked against the more exact calculations used in FLAME1D.

5. Whereas FLAME1D did not include algorithms for either viscosity or gravity, both of these effects are included in ***FLIC***.

6. At the moment, FLAME1D has a more flexible gridding algorithm. Because the basic convection algorithm in FLAME1D was Lagrangian, relatively non-diffusive cell splitting and merging routines are used to refine or coarsen the grid. The result is a rather general gridding capability. The general approach to adaptive gridding must be different in a multidimensional Eulerian code, and this is now being developed for ***FLIC***.

2. Convection

In this chapter, we describe the fluid convection algorithm, BIC-FCT, how it is used in ***FLIC***, and how the gravitational acceleration term is included. In detailed flame simulations that must resolve the individual species diffusion, the numerical diffusion that results from solving the convection equations numerically must be small enough so that we can resolve the physical diffusion processes. For high-speed flows, the Eulerian explicit monotone methods such as FCT [13] or most of the TVD [14] methods achieve this goal, and some even allow variable-order accuracy. Unfortunately, the timestep required by explicit methods must be small enough to resolve the sound waves in the system, otherwise the numerical method is unstable. There is little or no penalty paid for this small timestep in high-speed flow in which the physical phenomena evolve fast, but for low-speed flows, explicit methods are very inefficient and expensive. For example, resolving a microsecond of physical time with a timestep of 10^{-8} s requires 100 timesteps, but resolving one second requires 10^8 timesteps. For many low-speed flows this temporal accuracy is unnecessary; we need only to be able to resolve a millisecond of physical time with 100 timesteps. For this reason, implicit methods that allow large timesteps are usually used for calculations of low-speed flows.

One often-considered approach to eliminating numerical diffusion is to use Lagrangian methods, in which diffusion is totally absent by definition. We have found that this approach works well in one dimension, but there are a number of serious problems in multi-dimensions [15]. In complex flows, the multidimensional Lagrangian grid becomes distorted to the point where nearest neighbors are no longer connected by grid lines, a situation that leads to extremely inaccurate calculations. Eventually grid lines can even cross, which makes the solution unstable. Such problems are often avoided by a regridding procedure that actually adds diffusion to

the solution, or by using dynamically restructuring grid such as in SPLISH [16], which adds complexity. Except in one dimension, it has not yet been shown that Lagrangian methods are to be preferred because of the geometric complexities that arise.

Most common methods for solving convection problems use algorithms that produce ripples near steep gradients such as in a flame or shock front. The first high-order, nonlinear, monotone algorithm, Flux-Corrected Transport (FCT) [13], was designed to prevent these ripples by maintaining local positivity near steep gradients while keeping a high order of accuracy elsewhere. Other nonlinear methods have been reviewed by Woodward and Collela [14]. Although these methods are explicit, there are recent reports on implicit, nonlinear methods [17,18]. A major problem with applying these implicit methods to low-speed flows is that they are expensive even though they can be very accurate.

The Barely Implicit Correction to Flux-Corrected Transport, BIC-FCT [7], was designed to overcome the problem of numerical diffusion in low-velocity implicit methods. BIC-FCT combines an explicit high-order, nonlinear FCT method [5,6] with an implicit correction process. This method removes the timestep limit imposed by the speed of sound on explicit methods, retains the accuracy required to resolve the detailed features of the flow, and keeps the computational cost as low as possible.

2.1 The BIC-FCT Algorithm

BIC-FCT is based on an approach suggested by Casulli and Greenspan [19], who showed that it is not necessary to treat all of the terms in the gas-dynamic equations implicitly to be able to use longer timesteps than those dictated by explicit stability limits. Only those explicit terms which force a timestep limit due to the sound speed have to be treated implicitly. In a pure convection problem, the timestep is still limited by the fluid velocity, but for the low Mach number flow in flames, this results in a hundredfold increase in the timestep. The term "Barely Implicit Correction" emphasizes that only the minimal number of terms in the conservation equations are treated implicitly.

BIC-FCT solves the convective portion of the Navier-Stokes equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{\mathbf{V}}\right) = 0 , \tag{2.1}$$

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot \left(\rho \vec{V} \vec{\mathbf{V}}\right) = -\nabla P + \vec{F} , \tag{2.2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(E \vec{\mathbf{V}}\right) = -\nabla \cdot \left(P \vec{\mathbf{V}}\right) + S , \tag{2.3}$$

$$\frac{\partial n_k}{\partial t} + \nabla \cdot \left(n_k \vec{\mathbf{V}}\right) = 0 . \tag{2.4}$$

where $\vec{F}$ is a body force, usually gravity, and S is used to couple in the contributions to the change in energy due to other processes as discussed in chapter 5. In addition, note that convection of species is done at the same time.

BIC-FCT takes a predictor-corrector approach. The first explicit step uses FCT and a large timestep governed by a CFL condition on the fluid velocity,

$$\frac{\rho' - \rho^o}{\Delta t} = -\nabla \cdot \left(\rho^o \vec{V}^o\right), \tag{2.5}$$

$$\frac{\rho' \vec{V}' - \rho^o \vec{V}^o}{\Delta t} = -\nabla \cdot \left(\rho^o \vec{V}^o \vec{V}^o\right) - \nabla P^o, \tag{2.6}$$

$$\frac{E' - E^o}{\Delta t} = -\nabla \cdot (E^o + P^o) \left[\omega \vec{V}' + (1-\omega)\vec{V}^o\right] + S, \tag{2.7}$$

where the implicitness parameter $0.5 \leq \omega \leq 1.0$. This produces the intermediate values denoted by primes.

The second step is an implicit correction requiring the solution of one elliptic equation for the pressure correction, $\delta P \equiv \omega(P^n - P^o)$:

$$\frac{\delta P}{(\gamma - 1)\omega \Delta t} - \omega \Delta t \nabla \cdot \left(\frac{E^o + P^o}{\rho'} \nabla \delta P\right) = \frac{E' - E^o}{\Delta t} - \frac{\rho' \vec{V}'^2 - \rho^o \vec{V}^{o2}}{2\Delta t} \tag{2.8}$$

The elliptic equation (Eq. (2.8)) is solved by the multigrid method, MGRID [20]. This method is O(N) in both number of computations as well as storage. It is vectorized and extremely efficient on the CRAY X-MP. MGRID requires that the number of grid points in each direction be factorizible by a large power of two. While this has not proven to be very restrictive for ***FLIC***, it has been a problem in other applications. MGRID also has only a limited set of boundary conditions. Fortunately, the Neumann boundary conditions used in solving Eq. (2.8) has been implemented.

The final step is the correction of the provisional values for momentum, energy, and pressure:

$$\rho^n \vec{V}^n = -\Delta t \nabla \delta P + \rho' \vec{V}', \tag{2.9}$$

$$E^n = \frac{\delta P}{(\gamma - 1)\omega} + E^o, \tag{2.10}$$

$$P^n = \delta P + P^o. \tag{2.11}$$

Because BIC-FCT uses FCT for the explicit step, it has the high-order monotone properties that accurately treat sharp gradients. This accuracy combined with the savings that result from removing the sound-speed limitation on the timestep makes BIC-FCT a very cost effective convection algorithm. BIC-FCT takes about 15μs per point per computational timestep on the CRAY X-MP computer. This is as fast as the explicit FCT code currently in use. BIC-FCT has opened up the possibility of doing accurate, multidimensional, slow-flow calculations in which fluid expansion is important. Because the cost of BIC-FCT is modest even in two dimensions, reasonably detailed chemistry models as well as other physical processes can be included. Premixed flames, diffusion flames, and turbulent jet flames are some of the applications for which BIC-FCT is well suited.

2.2 Gravity

Buoyancy effects due to the force of gravity have been incorporated in ***FLIC***. The body force term $\vec{F}$ in Eq. (2.6) is used for this purpose and is given by

$$\vec{F} = \vec{g}(\rho - \rho_\infty) \tag{2.12}$$

where ρ_∞ is a suitable reference density, usually the cold ambient density. As currently implemented, the direction of the gravity vector is aligned with the flow direction, but this restriction can be removed trivially. Indeed, the gravity vector can be made time-dependent to simulate g-jitter in microgravity.

If the ambient density cannot be used, the hydrostatic head has to be included in its place. This second approach is not as suitable as the method given by Eq. (2.12) because of the need for very high precision calculations.

3. Chemistry

This chapter describes the integration package TBA that is used to solve the ordinary differential equations (ODE's) representing the chemical reactions and energy release. TBA is a fully vectorized FORTRAN subroutine for the CRAY X-MP. It is designed to replace VSAIM, the older vectorized version of CHEMEQ [8,9], an algorithm that solves a system of ODE's in a single computational cell. Both VSAIM and TBA are based on CHEMEQ and are designed to make the calculation of a large number of sets of ODE's more efficient. TBA is faster than VSAIM because it is designed to make specific use of the Cray X-MP's gather/scatter hardware and other capabilities.

The ODE's solved by these routines are of the form

$$\frac{dn_i}{dt} = F_i = Q_i - L_i n_i, \tag{3.1}$$

where n_i is the number density of species i, Q_i is the rate of formation of species i, and $L_i n_i$ is the rate of destruction of species i. Sometimes these equations can be solved by classical algorithms, sometimes they are stiff and need special techniques. TBA uses a different algorithm for each type of equation and gains efficiency by gathering all equations of a given type together and integrating them by groups.

A detailed hydrogen-oxygen reaction scheme has been implemented in ***FLIC***. This reaction scheme consists of forty-eight reversable reactions involving eight species. Nitrogen, acting as a diluent, is considered to be chemically inert. This reaction scheme, developed by Burks and Oran [21], has been used by Kailasanath *et al.* [22] in FLAME1D and is given in table 3.1. The total rate of formation and destruction of each species is obtained algebraically from the reaction rates of the individual chemical reactions and from the species concentrations. The reaction

rate for each chemical reaction r is assumed to follow the modified Arrhenius form:

$$k_r = AT^B e^{-C/T} \tag{3.2}$$

The computation of the rates in Eq. (3.2) is quite time consuming, mainly due to the calculation of the exponential term. Some time can be saved by computing the temperature dependence of the reaction rates only once per global timestep. The rates of formation and destruction of the species, which are also dependent on the species concentrations, is computed as often as needed by TBA, which updates them many times during each global timestep.

3.1 Numerical Method

TBA uses a second-order predictor-corrector method that is essentially the same numerical integration algorithm as CHEMEQ [9]. Normal ODE's are integrated using a simple classical scheme and stiff ODE's are integrated using an asymptotic method. Unlike CHEMEQ or VSAIM, TBA also recognizes equations that will approach equilibrium during the period of integration and handles them with a third scheme. TBA sorts the equations from every cell into one of three types — normal, stiff, or equilibrium, and then integrates all of the equations of each type together. A large number of cells are integrated simultaneously: as cells are completed, the results are returned to the control program and stored and data from new cells are read in and integrated.

The entire set of equations from each cell is integrated using the smallest timestep required by any equation in the set. The timestep is then increased or decreased based on the relative difference between the predictor and corrector stages. The predictor stages for the three types of equations are:

$$n_i' = n_i^o + \delta t F_i^o \ , \qquad \text{(Normal)} \tag{3.3}$$

$$n_i' = n_i^o + \frac{\delta t F_i^o}{1 + \delta t L_i^o} \ , \qquad \text{(Stiff)} \tag{3.4}$$

$$n_i' = Q_i / L_i \ . \qquad \text{(Equilibrium)} \tag{3.5}$$

The corrector stages are:

$$n_i^n = n_i^o + \frac{\delta t}{2} \left[F_i^o + F_i' \right] \ , \qquad \text{(Normal)} \tag{3.6}$$

Reaction Rate	A[a]	B	C[a]	Source
$H + HO \rightleftharpoons O + H_2$	1.40(−14)	1.00	3.50(+03)	[23]
	3.00(−14)	1.00	4.48(+03)	[23]
$H + HO_2 \rightleftharpoons H_2 + O_2$	4.20(−11)	0.00	3.50(+02)	[23]
	9.10(−11)	0.00	2.91(+04)	[23]
$H + HO_2 \rightleftharpoons HO + HO$	4.20(−10)	0.00	9.50(+02)	[23]
	2.00(−11)	0.00	2.02(+04)	[23]
$H + HO_2 \rightleftharpoons O + H_2O$	8.30(−11)	0.00	5.00(+02)	[24]
	1.75(−12)	0.45	2.84(+04)	$k_r = k_f/K_c$
$H + H_2O_2 \rightleftharpoons HO_2 + H_2$	2.80(−12)	0.00	1.90(+03)	[23]
	1.20(−12)	0.00	9.40(+03)	[23]
$H + H_2O_2 \rightleftharpoons HO + H_2O$	5.28(−10)	0.00	4.50(+03)	[23]
	3.99(−10)	0.00	4.05(+04)	$k_r = k_f/K_c$
$HO + H_2 \rightleftharpoons H + H_2O$	1.83(−15)	1.30	1.84(+03)	[25]
	1.79(−14)	1.20	9.61(+03)	[25]
$HO + HO \rightleftharpoons H_2 + O_2$	1.09(−13)	0.26	1.47(+04)	$k_f = k_r K_c$
	2.82(−11)	0.00	2.42(+04)	[26]
$HO + HO \rightleftharpoons O + H_2O$	1.00(−16)	1.30	0.00(+00)	[25]
	3.20(−15)	1.16	8.77(+03)	$k_r = k_f/K_c$
$HO + HO_2 \rightleftharpoons H_2O + O_2$	8.30(−11)	0.00	5.03(+02)	[27]
	2.38(−10)	0.17	3.69(+04)	$k_r = k_f/K_c$
$HO + H_2O_2 \rightleftharpoons HO_2 + H_2$	1.70(−11)	0.00	9.10(+02)	[23]
	4.70(−11)	0.00	1.65(+04)	[23]
$HO_2 + H_2 \rightleftharpoons HO + H_2O$	1.20(−12)	0.00	9.41(+03)	[26]
	1.33(−14)	0.43	3.62(+04)	$k_r = k_f/K_c$

Table 3.1 Chemical Reaction Rates for H_2-O_2 Combustion: $k_i = AT^B \exp(-C/T)$[b]

[a] Exponentials to the base 10 are given in parentheses: $1.00(-10) = 1.00 \times 10^{-10}$.

[b] Bimolecular reaction rate constants are in units of cm^3/ (molecule s).

Reaction Rate	A[a]	B	C[a]	Source
$HO_2 + HO_2 \rightleftharpoons H_2O_2 + O_2$	3.00(−11)	0.00	5.00(+02)	[24]
	1.57(−09)	−0.38	2.20(+04)	$k_r = k_f/K_c$
$O + HO \rightleftharpoons H + O_2$	2.72(−12)	0.28	−8.10(+01)	$k_f = k_r K_c$
	3.70(−10)	0.00	8.45(+03)	[23]
$O + HO_2 \rightleftharpoons HO + O_2$	8.32(−11)	0.00	5.03(+02)	[27]
	2.20(−11)	0.18	2.82(+04)	$k_r = k_f/K_c$
$O + H_2O_2 \rightleftharpoons H_2O + O_2$	1.40(−12)	0.00	2.12(+03)	[24]
	5.70(−14)	0.52	4.48(+04)	$k_r = k_f/K_c$
$O + H_2O_2 \rightleftharpoons HO + HO_2$	1.40(−12)	0.00	2.13(+03)	[24]
	2.07(−15)	0.64	8.23(+03)	$k_r = k_f/K_c$
$H + H + M \rightleftharpoons H_2 + M$	1.80(−30)	−1.00	0.00(+00)	[23]
	3.70(−10)	0.00	4.83(−04)	[23]
$H + HO + M \rightleftharpoons H_2O + M$	6.20(−26)	−2.00	0.00(+00)	[23]
	5.80(−09)	0.00	5.29(+04)	[23]
$H + O_2 + M \rightleftharpoons HO_2 + M$	4.14(−33)	0.00	−5.00(+02)	[23]
	3.50(−09)	0.00	2.30(+04)	[23]
$HO + HO + M \rightleftharpoons H_2O_2 + M$	2.50(−33)	0.00	−2.55(+03)	[23]
	2.00(−07)	0.00	2.29(+04)	[23]
$O + H + M \rightleftharpoons HO + M$	8.28(−29)	−1.00	0.00(+00)	[28]
	2.33(−10)	0.21	5.10(+04)	$k_r = k_f/K_c$
$O + HO + M \rightleftharpoons HO_2 + M$	2.80(−31)	0.00	0.00(+00)	[28]
	1.10(−04)	−0.43	3.22(+04)	$k_r = k_f/K_c$
$O + O + M \rightleftharpoons O_2 + M$	5.20(−35)	0.00	−9.00(+02)	[23]
	3.00(−06)	−1.00	5.94(+04)	[23]

Table 3.1 Continued Chemical Reaction Rates for H_2–O_2 Combustion: $k_i = AT^B \exp(-C/T)$[b]

[a] Exponentials to the base 10 are given in parentheses: $1.00(-10) = 1.00 \times 10^{-10}$.

[b] Bimolecular reaction rate constants are in units of cm^3/ (molecule s).

$$n_i^n = n_i^o + \frac{2\delta t\,[Q_i' - L_i^o n_i^o + F_i^o]}{4 + \delta t\,[L_i' + L_i^o]}\,, \qquad \text{(Stiff)} \qquad (3.7)$$

$$n_i^n = \frac{2Q_i'}{L_i' + L_i^o}\,. \qquad \text{(Equilibrium)} \qquad (3.8)$$

The original CHEMEQ report [9] describes the details of the timestep selection algorithm, stiffness criterion, and other important details. A detailed report on TBA [29] is currently under preparation.

3.2 Data-Handling Algorithm

TBA is designed to handle a large number of cells, each with an independent timestep. Thus, each cell is integrated with a different number of subcycles. Cells have to be constantly shuffled in and out of the integration routine. Whenever the integration of a cell is complete, it is put onto a list. At the end of the integration loop, this list is passed to another routine, CDATA, which stores the results and inserts new cells to be integrated into the places left by the completed cells. Towards the end of the integration procedure there are no additional cells to integrate and the arrays in the integrator will be only partially filled. When this occurs, we sort and move all completed cells to the ends of the arrays where they will not be integrated further. When the integration of the remaining cells is complete, all the data is written out in one operation. Thus we avoid both unnecessary shuffling and unnecessary calls to CDATA. The sort algorithm developed specificially for this application is O(N) and makes the minimum number of swaps. The sort is optimized by the use of CRAY assembly routines that make bitwise comparisons.

In the original VSAIM routine, all equations passed through one integration loop where they were tested for stiffness by `if` statements and either the stiff or normal calculations were performed. The CRAY X-MP vectorizes `if` statements by performing the calculations for both possibilities and then throwing away the results for the false condition, so this approach is wasteful. TBA creates index arrays for different types of equations and sets up a separate integration loop for each of the three types. This approach succeeds due to the speed of the CRAY X-MP gather/scatter hardware.

3.3 Programming Strategies

Many strategies or tricks were used to enhance the speed of TBA, most of which are in what is considered extremely poor programming style, but reflect certain idiosyncracies of CRAY programming in general. Some are documented in CRAY manuals, especially the *Optimization Guide* [30].

For example, many of the arrays are equivalenced as both one- and two-dimensional arrays. The CRAY will only vectorize the innermost nested loop, so wherever possible we looped over the one-dimensional equivalent array, to avoid an outer loop. Working space for TBA is supplied in a common block, as this proved substantially faster than passing it through the argument list in the calling sequence. Memory access is the bottleneck on the CRAY so scalar temporaries are used to reduce memory traffic. A full 50% speed up was achieved through their use. A power of two as the number of species results in memory conflicts that slow the program considerably. There are eight (2^3) chemical species in ***FLIC*** which would result in very poor performance. Thus a "dummy" species was put in, removing the memory conflict.

The following code fragment illustrates the importance of large vector lengths. Note that if the order of the loops were switched, the `if` could be taken out of the inner loop and the memory access made contiguous. However, a much shorter inner loop results, and the execution time is actually greater.

```
      do 250 i=1,numeqns
         do 240 j=1,numcells
            if (convchk(j)) then concent(i,j)=corr2d(i,j)
240      continue
250   continue
```

These sorts of tricks are higly specific to the CRAY X-MP computer and must be used to gain full advantage of its speed. Though TBA can be transported to other machines (it was developed on a VAX), it was written specifically for the CRAY X-MP with its gather/scatter hardware in mind.

4. Diffusive Transport Processes

The diffusive transport processes currently included in ***FLIC*** are molecular diffusion including the Dufour effect, thermal conduction, and viscosity, all processes that are crucial for describing flame structure. As yet, we have have not included the thermal effects on mass diffusion (thermal diffusion or the Soret effect), thermal dissipation due to viscosity, or radiation transport. Although it is a second-order effect, thermal diffusion may become important for near-limit flames and so should be included in the future. Thermal dissipation is negligible at the extremely low Mach numbers in the flows under consideration. Radiation effects are important in many flames, particularly hydrocarbon flames that form soot, but is not particularly important in hydrogen flames.

The differential equations describing these diffusive processes have been solved with a two-dimensional, explicit Eulerian scheme. Spatial derivatives have been approximated by central differences and a simple forward-Euler time marching scheme has been used for time advancement. This explicit method has a timestep limit roughly equal to the timestep required by BIC-FCT for the fluid convection. However, each diffusive transport process has its own stability condition, and on occasion it can require a timestep up to five times smaller. When this is the case, the approach we have taken is to subcycle the integration for that process. The alternate approach would be to use an implicit algorithm for selected terms, but this is generally more expensive than subcycling when fewer than ten subcycles are required.

4.1 Mass Diffusion

The part of the species conservation equations for species number density that describes mass diffusion are

$$\frac{\partial n_k}{\partial t} = -\nabla \cdot \left(n_k \vec{V}_k \right) \ , \quad n_k \ = \ 1, ... n_{sp} \ , \tag{4.1}$$

subject to the constraint

$$\sum_{k=1}^{n_{sp}} Y_k \vec{V}_k = 0 \ . \tag{4.2}$$

The effects of molecular diffusion in the energy equation (Dufour effect) appear as

$$\frac{\partial E}{\partial t} = -\nabla \cdot \left(\sum_{k=1}^{n_{sp}} n_k h_k \vec{V}_k \right) \ , \tag{4.3}$$

where h_k is the temperature-dependent enthalpy for each of the species. The values of $\{V_k\}$ are found by solving [31]

$$S_k = \sum_{\substack{j=1 \\ j \neq k}}^{n_{sp}} \frac{X_j X_k}{D_{kj}} (V_j - V_k) = \nabla X_k, \tag{4.4}$$

subject to

$$\sum_{k=1}^{n_{sp}} S_k = 0, \tag{4.5}$$

and Eq. (4.2), where X_k and Y_k are the mole and mass fractions of species k. The exact solution of this equation for the set $\{V_k\}$ can be obtained by solving the matrix equation implied by Eq. (4.4).

To avoid solving the full matrix equation at each location at each timestep, we find an approximation to the $\{V_k\}$ using Fick's Law and then correct this by the procedure described by Coffee and Heimerl [32] to ensure that the constraint Eq. (4.2) is met. The diffusion velocity according to Fick's law is

$$\widehat{V_k} = -\frac{1}{X_k} D_{km} \nabla X_k \ , \tag{4.6}$$

where D_{km} is the diffusion coefficient of species k in the mixture of gases. Equation 4.6 determines the diffusion velocities to within a constant. We then assume that a

constant $\vec{V}_c$ is added to all the raw diffusion velocities $\widehat{V}_k$ and require that the sum of the diffusion fluxes equal zero. Thus,

$$\sum_{k=1}^{n_{sp}} Y_k \vec{V}_k = \sum_{k=1}^{n_{sp}} Y_k \left(\widehat{V}_k + \vec{V}_c \right) = 0 \tag{4.7}$$

leads to

$$\vec{V}_c = - \sum_{k=1}^{n_{sp}} Y_k \widehat{V}_k \ . \tag{4.8}$$

The component of the corrected diffusion velocity defined by

$$\vec{V}_k = \widehat{V}_k + \vec{V}_c \tag{4.9}$$

is then used in Eq. (4.1).

The set of $\{V_k\}$ found in this way is algebraically equivalent to the first iteration of the DFLUX algorithm [12], an approach based on a matrix expansion that converges to arbitrary order. Values of $\{V_k\}$ obtained by the procedure described above were compared to the results of DFLUX to check their accuracy. The explicit finite differencing used to solve Eq. (4.1) has the numerical stability limit,

$$\max \left(D_{km} \Delta t / \Delta x^2 \right) < 1/2 \ .$$

Generally the mass-diffusion algorithm is subcycled within an overall timestep determined by the convection algorithm. However, the code is designed to decrease the overall timestep to below that required by the mass-diffusion stability limit if the number of subcycles required exceeds a specified maximum value. Subcycling becomes necessary when the temperature of the reacting flow becomes high and the diffusion coefficients increase accordingly.

Determining the diffusion velocities by Eq. (4.6) requires as input the set of diffusion coefficients of species k into the mixture, $\{D_{km}\}$. However, these quantities are difficult to obtain from first principles and are usually found by applying a mixture rule to the individual binary diffusion coefficients. Binary diffusion coefficients can be estimated theoretically [33] and sometimes measured experimentally [33,34]. Here we use the same approach as in FLAME1D (Kailasanath *et al.* [22]). Binary diffusion coefficients are expressed in the form

$$D_{kl} = \frac{A_{kl}}{n_{tot}} T^{B_{kl}} \tag{4.10}$$

	O	H_2	OH	H_2O	O_2	HO_2	H_2O_2	N_2
H	6.30(17)	8.29(17)	6.30(17)	6.70(17)	6.70(17)	6.70(17)	4.43(17)	6.10(17)
	7.28(-1)	7.28(-1)	7.28(-1)	7.28(-1)	7.32(-1)	7.32(-1)	7.28(-1)	7.32(-1)
O		3.61(17)	1.22(17)	2.73(17)	9.69(16)	9.69(16)	1.57(17)	9.69(16)
		7.32(-1)	7.74(-1)	6.32(-1)	7.74(-1)	7.74(-1)	6.32(-1)	7.74(-1)
H_2			3.49(17)	6.41(17)	3.06(17)	3.06(17)	4.02(17)	2.84(17)
			7.32(-1)	6.32(-1)	7.32(-1)	7.32(-1)	6.32(-1)	7.38(-1)
OH				2.73(17)	1.16(17)	9.69(16)	1.57(17)	9.69(16)
				6.32(-1)	7.24(-1)	7.74(-1)	6.32(-1)	7.74(-1)
H_2O					2.04(17)	2.04(17)	1.57(17)	1.89(17)
					6.32(-1)	6.32(-1)	6.32(-1)	6.32(-1)
O_2						8.74(17)	1.14(17)	8.29(16)
						7.24(-1)	6.32(-1)	7.24(-1)
HO_2							1.14(17)	8.85(16)
							6.32(-1)	7.74(-1)
H_2O_2								1.14(17)
								6.32(-1)

Table 4.1 Binary diffusion coefficients expressed in the form: $D_{jk} = A_{jk}\frac{T^{B_{jk}}}{N}$. For each pair of species, the upper term is A_{jk} and the lower term is B_{jk}, exponentials to the base 10 are given in parentheses.

where the sets $\{A_{kl}\}$ and $\{B_{kl}\}$ are tabulated [22] for each pair of species in the hydrogen–oxygen reaction system, and are given in table 4.1. These are then combined by a mixture rule [35,36]

$$D_{km} = \frac{1 - Y_k}{\sum_{\substack{l=1 \\ l \neq k}}^{n_{sp}} \frac{X_l}{D_{kl}}}, \tag{4.11}$$

which provides values of D_{km} to use in Eq. (4.6).

4.2 Thermal Conduction

The effects of thermal conduction are expressed in the energy equation as

$$\frac{\partial E}{\partial t} = -\nabla \cdot (\kappa \nabla T) \ , \tag{4.12}$$

where κ is the mixture thermal conductivity. Explicit finite differencing introduces a stability limit

$$\max\left(\kappa \Delta t / \rho c_p \Delta x^2\right) < 1/2 \ ,$$

where $\kappa/\rho c_p$ is the thermal diffusivity coefficient. Subcycling is used here in the same manner as it is for mass diffusion. However, this stability condition is less stringent than that for mass diffusion and typically only two or three subcycles are needed.

The mixture thermal conductivity κ is obtained by combining the thermal conductivities of the individual gases $\{\kappa_k\}$ that are in the mixture. The $\{\kappa_k\}$ are estimated theoretically and are a function of temperature. We have used the third-order polynomial fit in temperature determined by Laskey [37] and is presented in Table 4.2. The mixture thermal conductivity is then calculated using (Mathur *et al.* [38])

$$\kappa = \frac{1}{2}\left[\sum_{k=1}^{n_{sp}} X_k \kappa_k + \frac{1}{\sum_{k=1}^{n_{sp}} \frac{X_k}{\kappa_k}}\right] \ . \tag{4.13}$$

4.3 Viscosity

The viscosity terms in the Navier-Stokes equations are included in ***FLIC***. The stress-tensor term, which represents the effect of viscosity in the momentum conservation equations, is:

$$\frac{\partial \rho \vec{V}}{\partial t} = -\nabla \cdot \tau \ , \tag{4.14}$$

where

$$\tau = \left(\frac{2}{3}\mu - \lambda\right)\left(\nabla \cdot \vec{V}\right) \mathbf{I} - \mu\left[\left(\nabla \vec{V}\right) + \left(\nabla \vec{V}\right)^T\right] \ . \tag{4.15}$$

Species	A	B	C	D
H	4.710(3)	3.354(1)	-9.971(-3)	1.964(-6)
O	1.089(3)	1.038(1)	-3.739(-3)	8.251(-7)
H_2	6.306(3)	4.304(1)	-8.505(-3)	2.160(-6)
OH	1.679(3)	1.091(1)	-1.613(-3)	4.150(-7)
H_2O	-2.077(2)	1.603(1)	-7.932(-4)	1.530(-7)
O_2	3.862(2)	8.613(0)	-1.966(-3)	3.619(-7)
HO_2	1.576(2)	1.070(1)	-1.143(-3)	4.471(-8)
H_2O_2	-1.097(3)	9.895(0)	-1.779(-3)	1.396(-7)
N_2	7.024(2)	6.917(0)	-1.191(-3)	2.035(-7)

Table 4.2 Thermal conductivity of species $k, \kappa_k = A + BT + CT^2 + DT^3$, erg/cm-s-K. Exponentials to the base 10 are given in parentheses.

and μ is the dynamic viscosity coefficient. The quantity λ, the second coefficient of viscosity, is set to $-2/3\mu$. The stress tensor τ includes all the viscous terms which arise in the compressible Navier-Stokes equations.

Eq. (4.14) is solved explicitly in the same manner as the mass diffusion or thermal conduction equations. Thus there is a stability criterion given by

$$\max\left(\mu\Delta t/\rho\Delta x^2\right) < 1/2 \ ,$$

where μ/ρ is the kinematic viscosity. Subcycling is used so that the overall computational timestep is set by the convection stability limit and not by the viscosity stability limit. If the number of subcycles required for stability exceeds some maximum value, the global timestep is reduced. The viscous diffusion algorithm was tested using two test problems: 1) the boundary-layer growth over a flat plate parallel to the flow, and 2) the boundary-layer thickness on a flat plate normal to the flow (stagnation point flow).

For the parallel-plate test, the velocity profile of the parallel flow was initially uniform along the plate with a typical velocity profile that is valid for a boundary-layer thickness greater than three computational cells. This particular initialization was chosen because a physical boundary layer less than three cells wide would be

swamped by numerical diffusion from the FCT algorithm that creates a boundary layer at least this thick. Setting up the problem this way simulates the growth of a boundary layer away from the leading edge of the plate and minimizes any numerical effects on the solution. The result of this test is a boundary layer whose growth matches the Blasius solution.

The stagnation-flow test was initialized with a boundary layer of constant thickness and uniform velocity along the plate. Again, the initial boundary layer thickness was more than three cells to minimize numerical effects. The results showed the development of a constant-thickness boundary layer whose velocity profile very closely matched that predicted by theory.

For a gas containing a single species k, the dynamic viscosity μ_k can be otained from the kinetic theory of gases [33]. Over a suitable range of temperature, this can be expressed as a third-order polynomial in temperature. Laskey [37] has compared this polynomial fit, presented in Table 4.3, to the calculations and to tabulated values for the viscosity and found good agreement. The mixture dynamic viscosity is calculated using the expression (Wilke [39])

$$\mu = \sum_{k=1}^{n_{sp}} \frac{X_k \mu_k}{\sum_{j=1}^{n_{sp}} X_j \Phi_{kj}} , \tag{4.16}$$

where

$$\Phi_{kj} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_k}{M_j}\right)^{-\frac{1}{2}} \left(1 + \left(\frac{\mu_k}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_k}\right)^{\frac{1}{4}}\right)^2 . \tag{4.17}$$

This mixture viscosity, μ, is used in the stress tensor (Eq. (4.15)) to model the viscous portion of the Navier-Stokes equations.

Species	A	B	C	D
H	1.516(-5)	1.074(-7)	-3.178(-11)	6.255(-15)
O	4.504(-5)	5.355(-7)	-1.811(-10)	3.740(-14)
H_2	2.802(-5)	2.236(-7)	-6.958(-11)	1.399(-14)
OH	5.630(-5)	5.193(-7)	-1.678(-10)	3.413(-14)
H_2O	-8.597(-6)	6.608(-7)	-2.305(-10)	4.601(-14)
O_2	4.930(-5)	5.861(-7)	-1.983(-10)	4.093(-14)
HO_2	5.006(-5)	5.952(-7)	-2.013(-10)	4.156(-14)
H_2O_2	-9.109(-6)	3.966(-7)	-1.345(-10)	2.623(-14)
N_2	5.302(-5)	4.596(-7)	-1.464(-10)	2.969(-14)

Table 4.3 Viscosity of species $k, \mu_k = A + BT + CT^2 + DT^3$,dyne-s/cm^2. Exponentials to the base 10 are given in parentheses.

5. Model Integration

The conservation equations contain terms representing convection, buoyancy, thermal conduction, molecular diffusion, viscous diffusion, and chemical reactions. The approach that ***FLIC*** uses is to determine a global timestep, solve equations representing the individual physical processes separately for that timestep, and then couple the solutions. The coupling procedure is a variation of the standard timestep splitting method of Yanenko [40] and described for reactive flows by Oran and Boris [15]Chapters 4 and 13.

The usual explicit timestep-splitting approach assumes that in some predetermined global timestep, the effect of all the physical processes can be evaluated as a running sum of the effects of individual processes. Each physical process is integrated independently using the results of the previous process as initial conditions. This method is correct in the limit of small timesteps and works well in a practical sense when the changes in the variables during the global timestep are small. Using this appproach, the global timestep is often limited to the smallest timestep required by the stability limits of the integration algorithms for the various processes. This is the approach we have used in a number of programs in which the convection is solved by an explicit integration procedure. The global timestep is usually determined by the CFL condition on the sum of the sound speed and the fluid velocity. The chemistry integration is subcycled in the global timestep. However, subcycling can sometimes be used for individual processes if changes in variables due to that particular process are not too large.

Figure 5.1 summarizes the integration process in a typical ***FLIC*** timestep. The global timestep is first estimated based on the stability requirements of the convection algorithm, BIC-FCT. Then it is compared with the stability requirements of the particular physical processes which are allowed to subcycle if necesary to

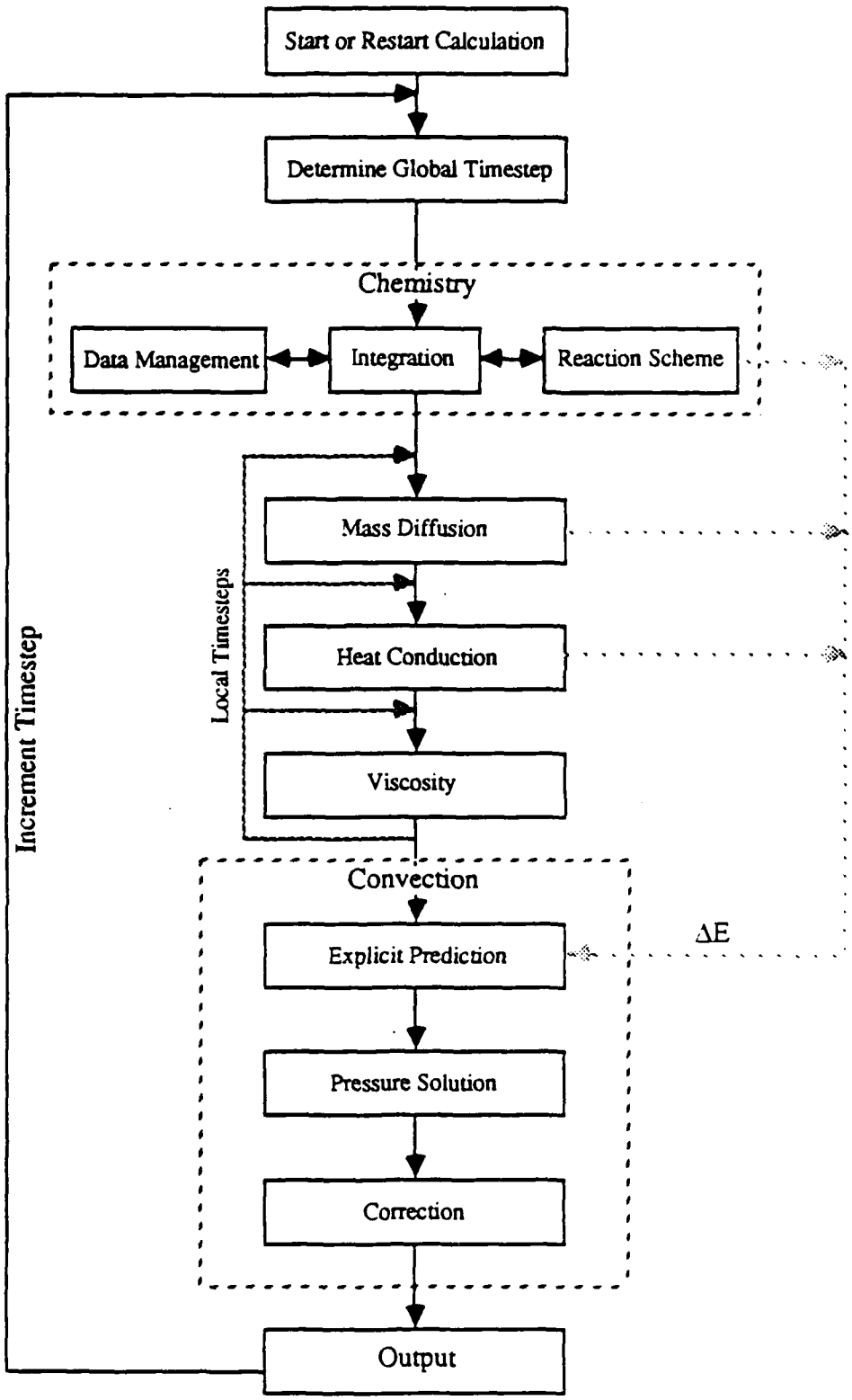

Figure 5.1 Flowchart of *FLIC*, indicating interprocess communication.

ensure stability. The overall timestep must be decreased sometimes to ensure that a physical variable does not change too much during a timestep. The most obvious difference between the scheme shown in Fig. 5.1 and the standard timestep splitting approach is that changes in the internal energy caused by each process are evaluated and added up at the end of a global timestep. This energy change is then used by the BIC-FCT convection algorithm (see Eq. (2.7)). This approach is quite similar to that used in FLAME1D [22], however, now changes in the internal energy are accumulated instead of changes in the pressure. The ADINC method [11], used in FLAME1D, does not solve an energy equation, and thus the only way other processes can be coupled is through the pressure. The BIC-FCT scheme used by ***FLIC*** does not require that energy changes be applied only during the convection step; this is merely a convenience that allows for larger global timesteps due to tighter coupling.

Some of the individual process integrations in ***FLIC*** are subcycled within a global timestep, including the ordinary differential equations representing the chemical reactions and the diffusive terms such as molecular diffusion and thermal conduction. For example, the timestep limit imposed by some chemical reactions may be orders of magnitude lower than that required by other physical processes, and so the chemistry integration is subcycled. Subcycling is built into the chemistry integration in an extremely sophisticated manner [9], so that the maximum allowable timestep at each computational cell is used, completely independent of the timestep in other cells. The chemistry is integrated up to the overall timestep before it is coupled to the other processes. However, if the energy release in an overall timestep due to chemistry is too large (typically greater than 10%), then the overall timestep must be decreased. Mass diffusion and thermal conduction are also subcycled, but only up to five times. The accuracy of the solution is generally tested by performing a separate calculation with a smaller timestep and noting whether the solution has converged.

All dependent variables, except for internal energy, are updated after each process integration, but dependent variables are not updated during subcycling of a process. This leads to a considerable savings, especially during the chemistry integration, because the evaluation of the temperature exponentials in Arrhenius expressions is done only once per global timestep. On the other hand, the global timestep may have to be decreased if the dependent variables change too much.

One consequence of a source term in the energy equation is the possibility of "ripples" in the pressure, which then manifest themselves in other variables as well. These ripples arise if a strong source is localized to a region only a few cells wide. These ripples are insignificant in ***FLIC*** where the flame zone is well resolved, but can be significant in other applications [37]. One way to avoid the ripples is to use a high-frequency filter such as

$$P^{filtered} = P + \alpha \nabla^4 P,$$

where α is a small constant. Other filters, including another FCT step, can also be used.

The particular advantages to using timestep splitting are that we can write very modular programs in which the integration of each physical process can be carried out with an optimum method, debugging is simpler, and explicit subcycling can be used to keep the costs down. Implicit methods can be used to avoid subcycling, but are more expensive when compared to explicit methods subcycled only a few times. In ***FLIC***, only five subcycles are allowed for each of the diffusion processes. Chemistry, on the other hand, subcycles thousands of times. At this point, it is not entirely clear if the extreme simplicity of the CHEMEQ scheme [9] results in faster integration of the chemistry equations than a more complicated implicit scheme. Disadvantages of timestep splitting are that the coupling process can be complicated, the algorithms and the overall program are less stable, and the timestep must be carefully controlled. However, we have found that the benefits of modular and fast programs outweigh potential disadvantages. This is discussed in some detail by Oran and Boris [15], pages 131–133.

6. Applications

FLIC has been written in a general manner so that it can be applied to solve a variety of slowly evolving problems involving chemistry and diffusive transport processes. It has already been applied to the study of multidimensional flames in premixed gases [1,41,42] and co-flowing diffusion flames [37,43]. A specific application of the code to the study of cellular flames in hydrogen-oxygen-nitrogen mixtures is described below to show what is required to perform a calculation with the code and to interpret the output from the code. Then other applications of the code are briefly discussed to bring out the generality of ***FLIC***.

6.1 A Sample Calculation

Flames in lean hydrogen-oxygen-nitrogen mixtures are known to exhibit a multidimensional cellular structure. The cellular structure is the result of a thermo-diffusive instability of a planar flame in the same mixture. Below we demonstrate how FLIC can be used to simulate the transition from a planar flame to a multidimensional cellular flame in a zero-gravity environment.

We need as input to the model:

A chemical reaction scheme involving all the species of interest (table 3.1),

Molecular diffusion coefficients for each pair of species (table 4.1),

For each species:

Molecular weights,

Thermal conductivity (Table 4.2),

Viscosity (Table 4.3),

Heats of formation [22],

Enthalpy coefficients [22].

To complete the specification of the problem, we also need the initial and boundary conditions such as composition, pressure, and temperature, in addition to the physical and chemical parameters given above. Figure 6.1 describes the configuration studied and gives the boundary conditions of the computational domain. Unburnt gas flows in from the left, and reaction products of the flame front flow out at the right. If the inlet velocity equals the burning velocity of the flame, the flame is fixed in space yielding a steady solution. Thus, transient effects from ignition can be eliminated. The initial conditions for the two-dimensional calculations were taken from one-dimensional calculations that gave the conditions for steady, propagating flames. The two-dimensional computational domain for this simulation was 2.0 cm $\times$ 4.5 cm, resolved by a 56 $\times$ 96 variably spaced grid. Fine zones, 0.36 mm $\times$ 0.15 mm, were clustered around the flame front.

The initial conditions specify a planar flame in a fuel-lean hydrogen-oxygen mixture diluted with nitrogen, H_2:O_2:N_2/1.5:1:10, a flame that showed multidimensional structure in the experiments by Mitani and Williams [4]. In order to study the evolution to cellular structure, the initial conditions were perturbed by displacing the center portion of the planar flame in the direction of the flow. The evolution to cellular structure is obtained by studying the output from the simulations.

The output from the calculation consists of the spatial and temporal distribution of all the species involved as well as the fluid density, temperature, pressure, internal energy, and momenta. Display of the data is not done by ***FLIC***, but is instead done as a post-processing operation. This cuts down the length and complexity of the ***FLIC*** code. A very useful diagnostic is contour plots showing isotherms and species distributions. Diagnostics, such as color-flood plots, allow visual interpretation of the data.

Figure 6.2 shows a sequence of isotherms from the calculation. The isotherms show that the temperature increases in the center portion of the flame, convex to the flow, and decreases in the the two adjacent concave regions, indicating more vigorous reaction in the convex region. The atomic hydrogen concentration increases in the convex and decreases in the concave regions. Also, the burning velocity in the

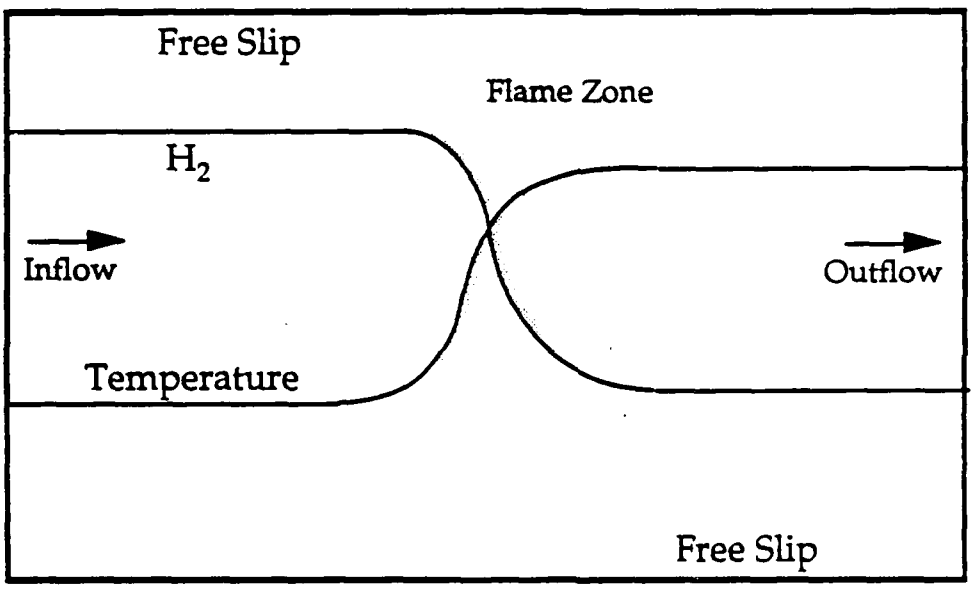

Figure 6.1 Initial and boundary conditions for the two-dimensional flame calculations.

convex region appears slightly higher than the burning velocity in a planar region, and the burning velocity in the concave regions were noticeably lower. Thus, this calculation shows that the planar lean one-dimensional hydrogen flame is unstable and evolves into a multidimensional flame having a cellular structure.

6.2 Applications of *FLIC*

The ***FLIC*** code has been used extensively to study the detailed evolution of cellular flames in premixed gases [1], to investigate the mechanisms which can lead to cellular structure [42], effects of gravity on flame structure and instabilities [41,42]. These studies have helped strengthen the prevailing theory of cellular instability and cast serious doubt on another theory which was also under consideration [1] ***FLIC*** has been geared primarily to these types of applications and several other related applications to premixed flames are planned for the near future.

FLIC can be converted to the study of diffusion flames extremely easily, and is begging for a suitable application in this area. A low-speed diffusion flame code [37,43] which has been used to study jet flames uses the same transport and diffusive packages as ***FLIC***. This code uses simplified chemistry, which was first calibrated by comparison to a detailed calculation with ***FLIC***.

6.3 Future Applications and Code Development

Calculations such as the one described earlier are time consuming (5 hours of CRAY X-MP time). There is interest to carry out these calculations in a larger domain for longer times. The cellular structure exhibited by flames is actually three-dimensional, so if these instabilities are to be studied fully, a three-dimensional version of ***FLIC*** is required. The computer time for these studies can quickly become intolerable; it is proposed to alleviate this by taking advantage of the multiprocessor architechure of the CRAY X-MP and the new Y-MP with its 16 processors. This will require some restructuring of the code and its algorithms to utilize microtasking appropriately. This restructuring may be simplified with the new "autotasking"

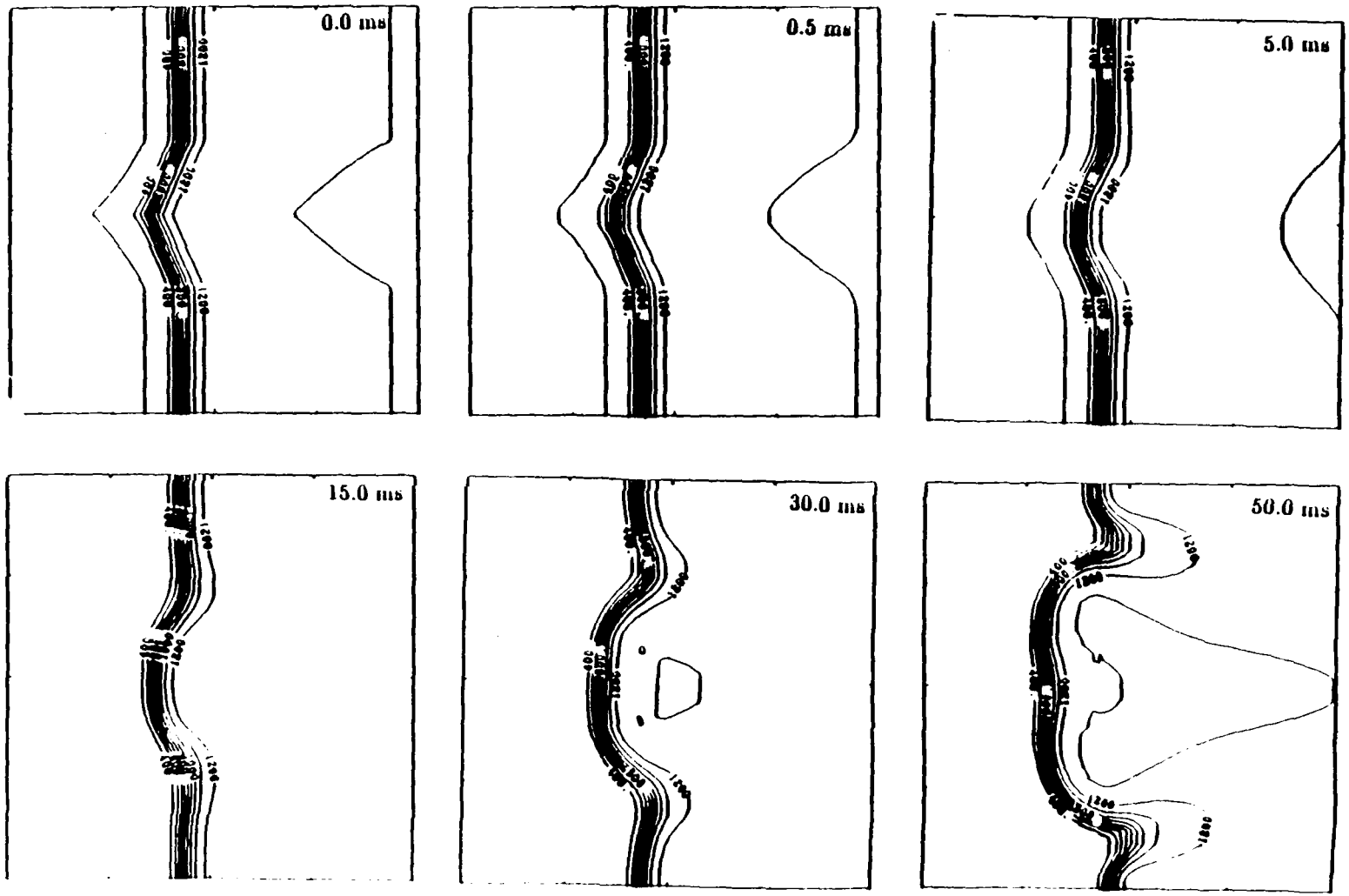

Figure 6.2 Evolution of isotherms in a fuel-lean hydrogen flame.

facility on the CRAY Y-MP.

One area of interest is the investigation of the behavior of flames near extinguishment. In particular, the prediction of flammability limits of flames in a flammability tube will provide a means for quantitative comparison with experiment. This complex task will require the addition of mechanisms which represent losses, including losses to the wall. The physics of the loss mechanisms of radicals to the walls is not yet fully understood. These additional mechanisms will need to be incorporated into ***FLIC***.

Detailed calculations will need to be performed for other more complex fuels of practical interest. The primary difficulty is in coping with the large number of species and chemical reactions that will be required for even the simplest hydrocarbon fuels. While the precise scaling of computer time with chemical complexity is not known, it is expected that the increase in the number of species will have the most dramatic effect. Thus, there is a need for reliable simplified chemistry models which have been first validated against a full model. It is anticipated that suitable simplified models for methane will become available soon. Radiation can not be neglected in hydrocarbon combustion. Thus a gas phase radiation model will have to be incorporated. Consideration of soot will have to wait until reliable models are available, which will not be in the near future.

All future applications are in areas that will require enormous amounts of computer time. One way to cut down on costs is to perform calculations only where they are strictly required. Calculations can be skipped in regions of low gradients, be it spatial or temporal gradients. This will require dynamic regridding or, more generally, re-discretization of the governing equations. A suitably efficient algorithm for this is essential, and its development is underway. Limitations and peculiarities of the target computer architecture will play a large role in determining the shape of these future versions of ***FLIC***.

Acknowledgments

This work was sponsored by NASA in the Microgravity Sciences Program and the Office of Naval Research through the Naval Research Laboratory. We also acknowledge some computational support from the Pittsburgh Supercomputing Center.

References

[1] G. Patnaik, K. Kailasanath, K. Laskey, and E. S. Oran. Detailed Numerical Simulations of Cellular Flames. In *Proceedings of the 22nd Symposium (International) on Combustion*, Pittsburgh, PA, 1989. The Combustion Institute.

[2] G. I. Barenblatt, Y. B. Zeldovich, and A. G. Istratov. On Diffusional Thermal Stability of Laminar Flame. *Zh. Prikl. Mekh. Tekh. Fiz.*, 4:21–26, 1962.

[3] G. I. Sivashinsky. Diffusional-Thermal Theory of Cellular Flames. *Combust. Sci. Tech.*, 15:137–146, 1977.

[4] T. Mitani and F. A. Williams. Studies of Cellular Flames in Hydrogen-Oxygen-Nitrogen Mixtures. *Combust. Flame.*, 39:169–190, 1980.

[5] J. P. Boris and D. L. Book. Solution of Convective Equations by the Method of Flux-Corrected Transport. In *Methods in Computational Physics*, volume 16, chapter 7, page 85. Academic Press, New York, 1976.

[6] J. P. Boris. Flux-Corrected Transport Modules for Solving Generalized Continuity Equations. Memorandum Report 3237, Naval Research Laboratory, 1976.

[7] G. Patnaik, R. H. Guirguis, J. P. Boris, and E. S. Oran. A Barely Implicit Correction for Flux-Corrected Transport. *J. Comput. Phys.*, 71:1–20, 1987.

[8] T. R. Young and J. P. Boris. A Numerical Technique For Solving Stiff Ordinary Differential Equations Associated with the Chemical Kinetics of Reactive Flow Problems. *J. Phys. Chem.*, 81:2424, 1977.

[9] T. R. Young. CHEMEQ — A Subroutine for Solving Stiff Ordinary Equations. Memorandum Report 4091, Naval Research Laboratory, 1980.

[10] K. Kailasanath and E. S. Oran. Time-Dependent Simulations of Laminar Flames in Hydrogen-Air Mixtures. Memorandum Report 5965, Naval Research Laboratory, 1987.

[11] J. P. Boris. ADINC: An Implicit Lagrangian Hydrodynamics Code. Memorandum Report 4022, Naval Research Laboratory, 1979.

[12] W. W. Jones and J. P. Boris. An Algorithm for Multispecies Diffusion Fluxes. *Comp. Chem.*, 5:139–146, 1981.

[13] J. P. Boris. A Fluid Transport Algorithm That Works. In *Computing as a Language of Physics*, pages 171–189. International Atomic Energy Agency, Vienna, 1971.

[14] P. Woodward and P. Colella. The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks. *J. Comp. Phys.*, 54:115–173, 1984.

[15] E. S. Oran and J. P. Boris. *Numerical Simulation of Reactive Flow.* Elsevier, New York, 1987.

[16] M. J. Fritts and J. P. Boris. The Lagrangian Solution of Transient Problems in Hydrodynamics Using a Triangular Mesh. *J. Comp. Phys.*, 31:173–215, 1979.

[17] B. A. Fryxell, P. R. Woodward, P. Colella, and K-H Winkler. An Implicit-Explicit Hybrid Method for Lagrangian Hydrodynamics. *J. Comput. Phys.*, 62, 1986.

[18] H. C. Yee and A. Harten. Implicit TVD Schemes for Hyperbolic Conservation Laws in Curvilinear Coordinates. In *AIAA 7th Computational Fluid Dynamics Conference*, pages 228–241, Washington, DC, 1985. AIAA.

[19] V. Casulli and D. Greenspan. Pressure Method for the Numerical Solution of Transient, Compressible Fluid Flows. *Int. J. Num. Methods Fluids*, 4:1001–1012, 1984.

[20] C. R. DeVore. Vectorization and Implementation of an Efficient Multigrid Algorithm for the Solution of Elliptic Partial Differential Equations. Memorandum Report 5504, Naval Research Laboratory, 1984.

[21] T. L. Burks and E. S. Oran. A Computational Study of the Chemical Kinetics of Hydrogen Combustion. Memorandum Report 4446, Naval Research Laboratory, 1981.

[22] K. Kailasanath, E. S. Oran, and J. P. Boris. A One-Dimensional Time-Dependent Model for Flame Initiation, Propagation, and Quenching. Memorandum Report 4910, Naval Research Laboratory, 1982.

[23] D. L. Baluch, D. D. Drysdale, D. G. Horne, and A. C. Lloyd. *Evaluated Kinetic Data for High Temperature Reactions*, volume 1. Butterworths, London, 1972.

[24] R. F. Hampson and D. Garvin. Chemical Kinetic and Photochemical Data for Modelling of Atmospheric Chemistry. NBS Technical Note 866, National Bureau of Standards, Washington, 1975.

[25] N. Cohen and K. R.. Westberg. Data Sheets. Technical report, The Aerospace Corporation, P. O. Box 92957, Los Angeles, CA, 1979.

[26] D. B. Olson and W. C. Gardiner Jr. An Evaluation of Methane Combustion Mechanisms. *J. Phy. Chem.*, 81:2514, 1977.

[27] A. C. Lloyd. Evaluated and Estimated Kinetic Data for Phase Reactions of the Hydroperoxyl Radical. *Int. J. Chem. Kinetics*, 6:169, 1974.

[28] G. S. Bahn. *Reaction Rate Compilation for H-O-N System.* Gordon and Breach, New York, 1968.

[29] T. A. Brun, T.R Young, and G. Patnaik. TBA — An Optimized Stiff ODE Solution Scheme. In preparation, 1989.

[30] P. J. Sydow. *Optimization Guide.* CRAY Research, Mendota Heights, 1983. CRAY Computer Systems Technical Note.

[31] S. Chapman and T. G. Cowling. *The Mathematical Theory of Non-Uniform Gases.* Cambridge University Press, Cambridge, 1970.

[32] T. P. Coffee and J. M. Heimerl. Transport Algorithms for Premixed Laminar Steady-State Flames. *Combust. Flame*, 43:273–289, 1981.

[33] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird. *Molecular Theory of Gases and Liquids.* John Wiley, New York, 1954.

[34] T. R. Marrero and E. A. Mason. Gaseous Diffusion Coefficients. *J. Phy. Chem. Ref. Data*, 1:3–118, 1972.

[35] C. F. Curtis and J. O. HIrschfelder. Transport Properties of Multicomponent Gas Mixtures. *J. Chem. Phys.*, 17:550, 1949.

[36] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport Phenomena.* John Wiley, New York, 1960.

[37] K. J. Laskey. *Numerical Study of Diffusion and Premixed Flames.* PhD thesis, Department of Mechanical Engineering, Carnegie-Mellon University, Pittsburgh, PA, 1989.

[38] S. Mathur, P. K. Tandon, and S. C. Saxena. Thermal Conductivity of Binary, Ternary, and Quaternary Mixtures of Real Gases. *Mol. Phys.*, 12:569, 1967.

[39] C. R. Wilke. A Viscosity Equation for GAs Mixtures. *J. Chem. Phys.*, 18:517, 1950.

[40] N. N. Yanenko. *The Method of Fractional Steps.* Springer-Verlag, Ney York, 1971.

[41] K. Kailasanath, G. Patnaik, and E. S. Oran. Effect of Gravity on Multi-Dimensional Laminar Premixed Flame Structure. IAF Paper 88-354, IAF, Paris, 1988.

[42] G. Patnaik, K. Kailasanath, and E. S. Oran. Effect of Gravity on Flame Instabilities in Premixed Gases. AIAA Paper 89-0502, AIAA, Washington, DC, 1989.

[43] J. L. Ellzey, K. J. Laskey, and E. S. Oran. Study of Low-Speed Diffusion Flames. In preparation, 1988.