FAST ALGORITHMS FOR COMBUSTION KINETICS CALCULATIONS: A COMPARISON*

Krishnan Radhakrishnan**
NASA Lewis Research Center

Many practical problems arising in chemically reacting flows require the simultaneous numerical integration of large sets of chemical kinetic rate equations of the type shown in figure 1. The initial value problem is that of finding the composition and temperature at the end of a prescribed time interval, given the initial mixture composition and temperature, the pressure, and the reaction mechanism. Multi-dimensional modeling of reactive flows requires the integration of the system of ordinary differential equations (ode's) given in figure 1 at several thousand grid points. To make such calculations practicable, it is necessary to have a very fast batch chemistry integrator.

To identify the fastest algorithm currently available for the numerical integration of chemical kinetic rate equations, several algorithms have been examined. In the present paper, we summarize our findings to date -- details are available in references (1) and (2). The algorithms examined in this work include two general-purpose codes EPISODE and LSODE (refs. 3 and 4), and three special-purpose (for chemical kinetic calculations) codes CHEMEQ (ref. 5), CREK1D (refs. 6 and 7), and GCKP84 (refs. 8 and 9). In addition, an explicit Runge-Kutta-Merson differential equation solver (ref. 10) (IMSL Routine DASCRU) is used to illustrate the problems associated with integrating chemical kinetic rate equations by a classical method. These methods are summarized in figure 3.

The algorithms summarized in figure 3 were applied to two test problems drawn from combustion kinetics. These problems, summarized in figure 4, included all three combustion regimes: induction, heat release and equilibration. Figures 5 and 6 present variations of the temperature and species mole fractions with time for test problems 1 and 2, respectively. Both test problems were integrated over a time interval of 1 ms in order to obtain near-equilibration of all species and temperature.

Of the codes examined in this study, only CREK1D and GCKP84 were written explicitly for integrating exothermic, non-isothermal combustion rate equations. These therefore have built-in procedures for calculating the temperature (T). For the other codes, two different methods, labeled as Methods A and B, were used to compute T. The following convention was adopted in naming these other codes: those using temperature method A were given the suffix-A (e.g. LSODE-A) and those using temperature method B were given the suffix-B (e.g. LSODE-B). In Method A, T was calculated from the mole numbers and the initial mixture enthalpy using an algebraic energy conservation equation (given in figure 7) and a Newton-Raphson iteration technique. In this method, the temperature is not an explicit independent variable, so the number of ode's is

[*] Work partially funded by NASA Grants NAG3-147 and NAG3-294.

[**] NRC-NASA Research Associate: on leave from The University of Michigan, Dept. of Mechanical Engineering and Applied Mechanics, Ann Arbor, MI 48109.

equal to the number, NS, of distinct chemical species in the mixture. The integrator therefore tracks only the solutions for the species mole numbers. In Method B, the temperature was treated as an additional independent variable and evaluated by integrating its time-derivative given in figure 7. In this method, the number of ode's is equal to NS+1, and the integrator tracks the solutions for both the temperature and the species mole numbers.

All codes were run on the NASA Lewis Research Center's IBM 370/3033 computer using single-precision accuracy, except GCKP84 which was in double-precision. A typical computational run consisted of initializing the species mole numbers, temperature, and CPU time. The integrator was then called with values for the necessary input parameters. On return from the integrator, the total CPU time required to solve the test problem was calculated. Other performance parameters were also recorded -- see reference (2) for details.

Figures 8 and 9 present the computational work (expressed as the CPU time in seconds) plotted against the local error tolerance, EPS, for test problems 1 and 2, respectively. For all codes except EPISODE, EPS is the local relative error tolerance. For EPISODE, EPS is a mixed error tolerance -- relative for species with initially nonzero mole numbers and for the temperature (method B) and absolute for species with initially zero mole numbers. Also shown on figures 8 and 9 are the CPU times required by the explicit Runge-Kutta method for one value of EPS. Note the excessive CPU times required by this technique. Its use would make multidimensional modeling of practical combustion devices prohibitively expensive.

For test problem 1, very small values for EPS had to be used for EPISODE (figure 8). For values of EPS $\geq 5 \times 10^{-6}$, EPISODE predicted little or no change in the composition and temperature after an elapsed time of 1 ms. Similar remarks apply to test problem 2 (figure 9), for which values of 10^{-4} and 10^{-3} had to be used for EPISODE-A and EPISODE-B, respectively. Although the runs with EPISODE-B and EPS $\geq 5 \times 10^{-4}$ were successfully completed, the solutions (especially for minor species) were significantly different from those given in figure 6. With GCKP84 and EPS = 10^{-2}, the solution for test problem 1 exhibited serious instability and so was terminated. A more detailed discussion of the accuracy of the codes tested in this study can be found in reference (2).

Examination of figure 8 shows that the difference in computational work required by methods A and B is small for test problem 1, with method B being more efficient. For test problem 2 (figure 9), the difference is small for large values of EPS. But for small values of EPS the difference is more marked, with method A being significanlty superior to method B.

Figures 8 and 9 show that LSODE and CREK1D are superior to the other codes. EPISODE is an attractive alternative, especially for test problem 2. However, in using EPISODE, a word of caution is in order. The computational work can be strongly dependent on the value for the initial steplength (H0) selected by the user. A poor guess for H0 can make EPISODE prohibitively expensive to use. Figure 10 illustrates this behavior for test problem 2. Note an order of magnitude increase in the CPU time for a change in H0 from 10^{-7} to 10^{-8} s. Although not shown here, a poor guess for H0 also resulted in inaccurate and unstable solutions. In addition, as discussed in reference (2), the error control performed by EPISODE is unsatisfactory for problems of the type eyamined in this study.

A simple method for increasing the efficiency of the algorithms as applied to the present problem was explored. This involved updating the rate constants k_j and k_{-j} (which was calculated from k_j and the concentration equilibrium constant) only for temperature changes greater than an amount ΔT. To avoid a trial and error search for the optimum value of ΔT -- defined as that value which results in minimum computational work -- an approximation for it was derived and is presented in figure 11. Comparisons of figure 12 with figures 7 and 8 show the significant reductions in computational work realized by use of the above approximation for ΔT.

References

1) Radhakrishnan, K., "A Comparison of the Efficiency of Numerical Methods for Integrating Chemical Kinetic Rate Eqautions," NASA TM, 1984. Presented at the 1984 JANNAF Propulsion Conference, Specialists Session, New Orleans, LA, 7-9 February, 1984.

2) Radhakrishnan, K., "A Comparison of Numerical Techniques for the Integration of Stiff ODE's arising in Combustion Chemistry," NASA TP, 1984.

3) Hindmarsh, A.C. and Byrne, G.D., "EPISODE: An Effective Package for the Integration of Systems of Ordinary Differential Equations," Lawrence Livermore Laboratory Report UCID-30122, Rev. 1, 1977.

4) Hindmarsh, A.C., "ODEPACK, A Systematized Collection of ODE Solvers," Lawrence Livermore Laboratory Report UCRL-88007, 1982.

5) Young, T.R. and Boris, J.P., "A Numerical Technique for Solving Stiff Ordinary Differential Equations Associated with the Chemical Kinetics of Reactive-Flow Problems," J. Physical Chemistry, 81, 1977, pp. 2424-2427.

6) Pratt, D.T., "CREK-1D: A Computer Code for Transient, Gas-Phase Combustion Kinetics," Paper 83-21, Proceeding of the Spring Technical Meeting of the Western States Section of the Combustion Institute, April 1983.

7) Pratt, D.T. and Radhakrishnan, K., "CREK-1D: A Computer Code for Transient, Gas-Phase Combustion Kinetics," to appear as a NASA Contractor Report, 1984

8) Zeleznik, F.J. and McBride, B.J., "Modelling the Internal Combustion Engine," NASA RP1094, 1984.

9) Bittker, D.A. and Scullin, V.J., "GCKP84-General Chemical Kinetics Code for Gas-Phase Flow and Batch Processes Including Heat Transfer," NASA TP, 1984.

10) Fox, L. (ed.), "Numerical Solution of Ordinary and Partial Differential Equations," Addison-Wesley Publishing Co., Inc., Reading, MA, 1962.

11) Pratt, D.T., "New Computational Algorithms for Chemical Kinetics," National Bureau of Standards Special Publication 561, 1979, pp. 1265-1279.

Adiabatic, Constant-Pressure, Gas-Phase Chemical Reaction

$$\frac{dn_i}{dt} = f_i(n_k,T) \quad ; \quad i,k = 1,NS$$

$$f_i = -\rho^{-1} \sum_{i=1}^{JJ} (\nu_{ij}' - \nu_{ij}'')(R_j - R_{-j})$$

$$R_j = k_j \prod_{k=1}^{NS} (\rho n_k)^{\nu_{kj}'}$$

$$R_{-j} = k_{-j} \prod_{k=1}^{NS} (\rho n_k)^{\nu_{kj}''}$$

$$k_j = A_j T^{N_j} \exp(-E_j/RT)$$

$$k_{-j} = A_{-j} T^{N_{-j}} \exp(-E_{-j}/RT)$$

In the above equations,

f_i = molar rate of formation of species i per unit mass of mixture, kmol-i/kg-mixture s
k_j, k_{-j} = forward and reverse rate constants for reaction j
n_i = mole number of species i, kmol-i/kg-mixture
A_j, A_{-j} = pre-exponential constants in forward and reverse rate equations for reaction j
E_j, E_{-j} = activation energy in forward and reverse rate equations for reaction j, cal/mole
JJ = number of distinct elementary reactions in mechanism
NS = number of distinct species in gas mixture
R = universal gas constant, 1.987 cal/mol K
R_j, R_{-j} = forward and reverse molar reaction rates per unit volume for reaction j, kmol/m^3 s
T = temperature, K
ρ = mixture mass-density, kg/m^3
ν_{ij}' ν_{ij}'' = stoichiometric coefficients of species i in reaction j as a reactant, and as a product, respectively

Figure 1 Governing Ordinary Differential Equations

Given

Initial Mixture Composition and Temperature

Pressure

Reaction Mechanism

Find, at the End of a Prescribed Time Interval

Mixture Composition and Temperature

Figure 2 Problem Statement

Method	Description
CCKP84	Details not yet available.
CREK1D	Variable-step, predictor-corrector method based on an exponentially-fitted trapezoidal rule; includes filtering of ill-posed initial conditions and automatic selection of Jacobi-Newton iteration or Newton iteration.
LSODE EPISODE	Variable-step, variable-order backward-differentiation method with a generalized Newton iteration*.
CHEMEQ	Variable-step, second-order predictor-corrector method with an asymptotic integration formula for stiff equations.
DASCRU	Variable-step, fourth-order, explicit Runge-Kutta-Merson solver.

*Other options are included in these packages.

Figure 3 Summary of Methods Studied

ORIGINAL PAGE IS
OF POOR QUALITY

Two Problems Describing Adiabatic, Constant Pressure Chemical Reactions

Test Problem 1:

Combustion of a Mixture of 33 % CO and 67 % H_2 with 100 % Theoretical Air
(taken from reference 11)

12 Reactions
11 Species + Temperature

Pressure = 10 atm.
Initial Temperature = 1000 K
Reaction Duration: 1 ms

Test Problem 2:

Combustion of a Stoichiometric Mixture of H_2 and Air
(taken from reference 9)

30 Reactions
15 Species + Temperature

Pressure = 2 atm.
Initial Mixture Temperature = 1500 K
Reaction Duration: 1 ms

Both Problems Include All Three Regimes of Combustion:
Induction, Heat Release and Equilibration.

Figure 4 Test Problems

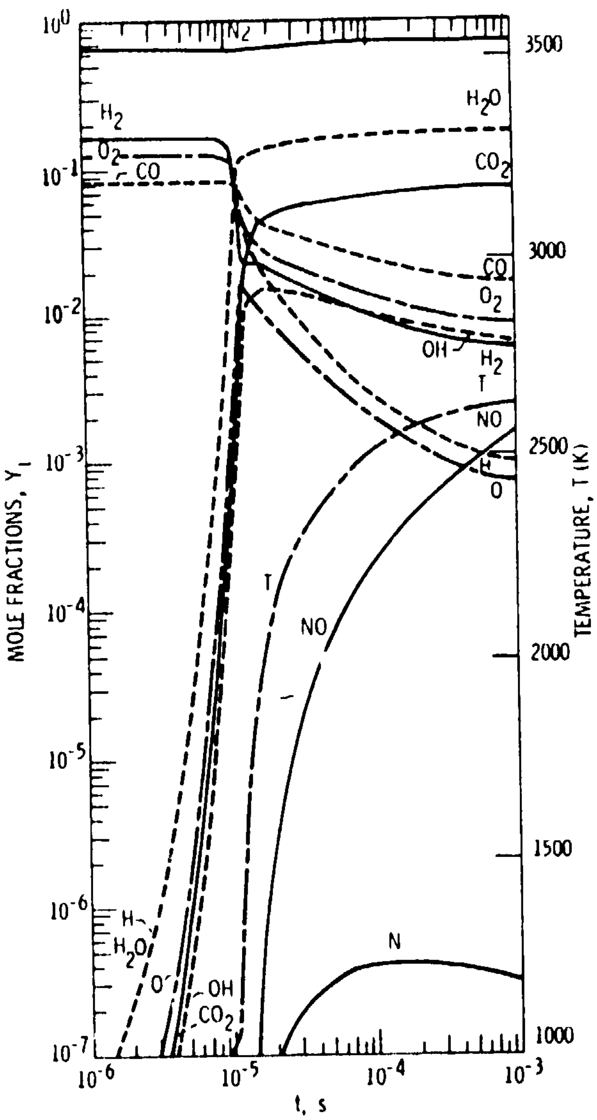

Figure 5 - Variation with time of temperature and species mole fractions for test problem 1. Solution generated with LSODE-B and EPS = 10^{-5}.

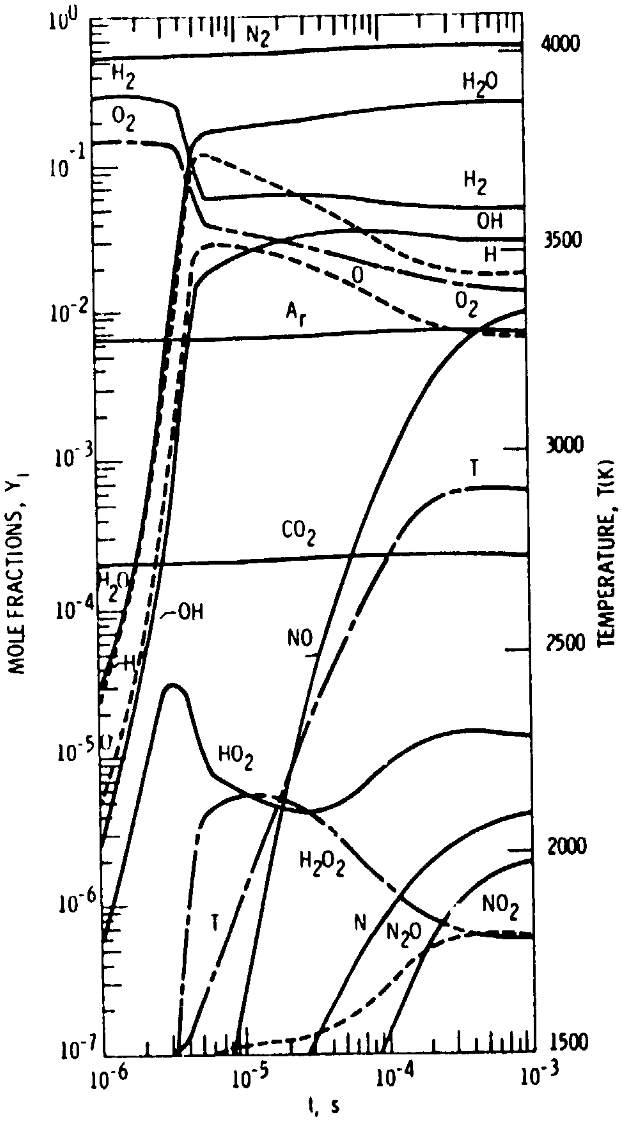

Figure 6 - Variation with time of temperature and species mole fractions for test problem 2. Solution generated with LSODE-B and EPS = 10^{-5}.

Method A

For adiabatic, constant-pressure combustion reaction, energy conservation gives

$$\sum_{i=1}^{NS} n_i h_i = h_0 = \text{constant} \qquad (1)$$

where, h_i = molal-specific enthalpy of species i (J/kmol)
and h_0 = mass-specific enthalpy of mixture (J/kg)

In this method, equation (1) was solved for the temperature using a Newton-Raphson iteration technique.

Method B

Differentiation of eq. (1) with respect to temperature (T) gives

$$\frac{dT}{dt} = - \frac{\sum_{i=1}^{NS} f_i h_i}{\sum_{i=1}^{NS} n_i c_{p_i}} \qquad (2)$$

where, c_{p_i} is the constant-pressure specific heat of species i (J/kmol K).

In this method, the temperature was evaluated by integrating equation (2).

Figure 7 Evaluation of Temperature
(for LSODE, EPISODE, and CHEMEQ)

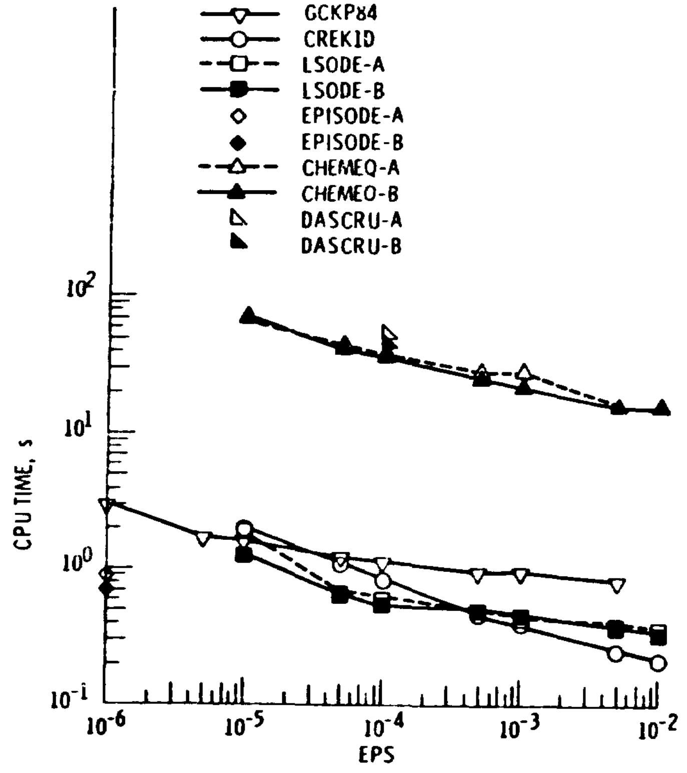

Figure 8 - Variation of the CPU time (s) with error tolerance, EPS, for test problem 1. All runs on IBM 370/3033.

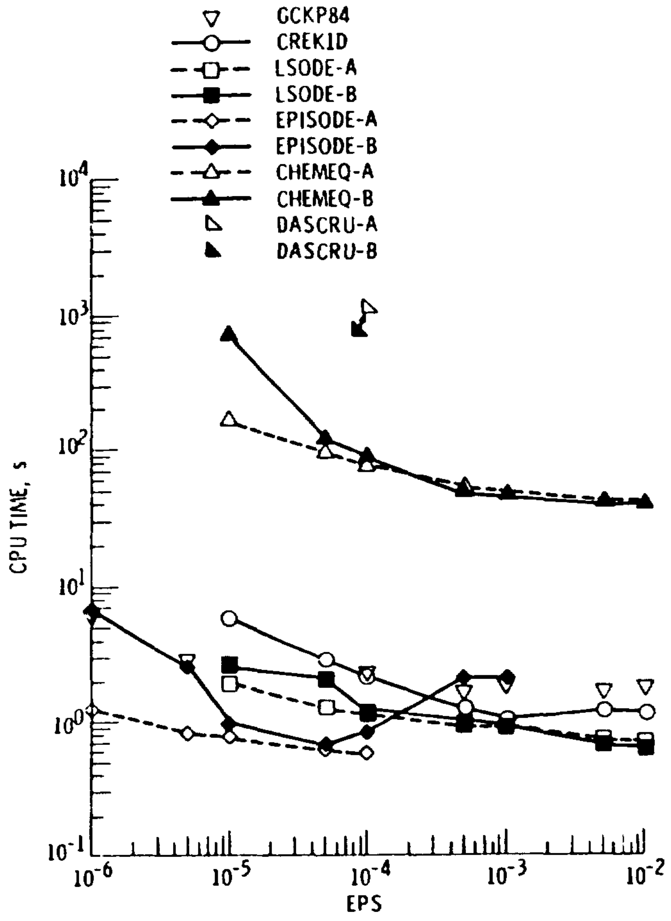

Figure 9 - Variation of the CPU time (s) with error tolerance, EPS, for test problem 2. All runs on IBM 370/3033.

HO (s)	CPU (s)
10^{-5}	0.786
10^{-6}	0.783
10^{-7}	0.791
10^{-8}	7.91
10^{-9}	8.04
10^{-10}	0.772

Figure 10 Example of Effect of Initial Steplength (HO) on Work Required by EPISODE-A (EPS = 10^{-8}) for Test Problem 2

An approximate expression for ΔT -- the maximum allowable temperature change allowed before the reaction rate constants k_j and k_{-j} are updated -- was derived by requiring that the maximum relative error in the resultant reaction rates does not exceed the local relative error tolerance (EPS) required of the numerical solution. The approximation for ΔT is given by

$$\Delta T = \frac{EPS \cdot T}{\max\limits_{j} \left| \frac{E_j}{RT} + N_j \; ; \; \frac{E_{-j}}{RT} + N_{-j} \right|} \qquad (3)$$

where, T is the current temperature, the bars $|\quad|$ denote absolute value, and the maximum is taken over all forward and reverse reactions.

Figure 11 Approximation for ΔT

Method	Test Problem	
	1	2
GCKP84	0.85	1.73
CREK1D	0.23	1.04
LSODE-A	0.31*	0.52*
LSODE-B	0.29*	0.51*
EPISODE-A	0.75*	0.54*
EPISODE-B	0.70*	0.67
CHEMEQ-A	6.41*	13.6*
CHEMEQ-B	5.69*	12.3*

*method incorporated eq. (3)

Figure 12 Minimum CPU Time (in seconds on IBM 370/3033 computer) Required for the Test Problems