NASA Technical Paper 3315

1993

A Critical Analysis of the Accuracy of Several Numerical Techniques for Combustion Kinetic Rate Equations

Krishnan Radhadrishnan
Lewis Research Center
Cleveland, Ohio

National Aeronautics and Space Administration
Office of Management
Scientific and Technical Information Program

Summary

A detailed analysis of the accuracy of several techniques recently developed for integrating stiff ordinary differential equations is presented. The techniques include two general-purpose codes, EPISODE and LSODE, which were developed for an arbitrary system of ordinary differential equations, and three specialized codes, CHEMEQ, CREK1D, and GCKP84, which were developed specifically to solve chemical kinetic rate equations. The accuracy study is made by applying these codes to two practical combustion kinetics problems. Each problem describes adiabatic, homogeneous, gas-phase chemical reactions at constant pressure and includes all three combustion regimes: induction, heat release, and equilibration. To illustrate the error variation in the different combustion regimes, the species are divided into three types: reactants, intermediates, and products. Error versus time plots are presented for each species type and the temperature. These plots show that CHEMEQ is the most accurate code during induction and early heat release. During late heat release and equilibration, however, the other codes are more accurate. A single global quantity, a mean integrated root-mean-square error, that measures the average error incurred in solving the complete problem is used to compare the accuracy of the codes. Among the codes examined, LSODE is the most accurate for solving chemical kinetics problems. It is also the most efficient code, in the sense that it requires the least computational work to attain a specified accuracy. An important finding is that use of the algebraic enthalpy conservation equation to compute the temperature can be more accurate and efficient than integrating the temperature differential equation.

Introduction

Many practical problems arising in chemically reacting flows require the simultaneous solution of large sets of coupled ordinary differential equations (ODE's) which describe the time rate of change of chemical species concentrations and temperature. Examples of such problems include the development and validation of reaction mechanisms, combustion of fuel-air mixtures, and pollutant formation and destruction.

The main difficulty in using classical methods, such as the popular explicit Runge-Kutta method (e.g., ref. 1), to solve large sets of chemical kinetic rate equations is that of "stiffness." The property of stiffness arises in chemical kinetics because of the widely varying time constants for different species. For free radicals the relaxation time is on the order of microseconds, whereas the nitric oxide formation time is on the order of milliseconds. To satisfy the stability requirements that errors in the numerical solution remain bounded as the calculation proceeds in time, classical methods must use extremely small step sizes, as illustrated in references 2 and 3 for the explicit Runge-Kutta method in solving combustion kinetics problems. Consequently, these methods require prohibitive amounts of computer time to solve a practical chemical kinetics problem.

Numerous approaches have been proposed for stiff ODE's to remove the stability restriction on the step size. In Part I of this effort (ref. 2) and other recent publications (refs. 3 to 5), several techniques were examined, and detailed comparisons of their computational work requirements for solving combustion kinetic rate equations were made. The methods examined in these studies include the general-purpose packages EPISODE and LSODE (refs. 6 to 9), which were developed for an arbitrary system of ODE's, and the specialized codes CHEMEQ (ref. 10), CREK1D (refs. 11 to 14), and GCKP84 (ref. 15), which have all been developed specifically to integrate chemical kinetic rate equations. In the present work the accuracy of these techniques in solving combustion kinetic rate equations is examined.

In general, numerical methods generate approximate solutions to the governing ODE's at discrete points in time. To maintain accuracy of the numerical solution, they require that the estimated error incurred on each time step be less than a user-specified local error tolerance. This result is usually achieved by restricting the size of the time step. Some solvers, in addition, adjust the order of the numerical approximation when appropriate. In either case, only the estimated local error, that is, the estimate of the error incurred in advancing the numerical solution by one time step, is controlled. However, the quantity that is of interest to the user is the global error, which is the deviation of the numerical approximation from the exact solution and which generally accumulates in a nontrivial manner from the local errors.

In the present paper, a detailed study of the estimated global error incurred by the above techniques in solving combustion kinetic rate equations is presented. Also presented is a study of the variation of the global error with the user-specified local tolerance and an examination of the computational cost, measured by the required CPU execution time, associated with

attaining desired accuracy. The paper concludes with two appendixes: Appendix A describes the methods examined in this study, and appendix B describes the procedure used to solve the algebraic enthalpy conservation equation for the temperature.

Symbols

A_j, A_{-j}	pre-exponential constants in forward and reverse rate coefficients for reaction j (eqs. (6) and (7)), units depend on reaction type
ATOL_i	local absolute error tolerance for i^{th} component, required by LSODE (eq. (20)).
ATOLSP	local absolute error tolerance used with LSODE for all species mole numbers
B_j	exponent-on-ten in pre-exponential constant for forward rate coefficient of reaction j, where $B_j = \log_{10} A_j$, arbitrary units
C_G	local error test constant used in GCKP84 (eq. (21))
$c_{p,i}$	constant-pressure molar specific heat of species i, J/kmole K
$d_{i,n}$	estimated local truncation error in i^{th} component at t_n
$E_{i,n}$	cumulative difference between converged and predicted values of (dY_i/dt) at t_n, used by GCKP84, units depend on component i
E_j, E_{-j}	activation energy in forward and reverse rate coefficients for reaction j (eqs. (6) and (7)), cal/mole
$\mathcal{E}_{\text{rms}}$	mean integrated root-mean-square global error (eq. (34))
EPS	for EPISODE and GCKP84: local relative error tolerance for species with initially nonzero mole numbers and temperature, and local absolute error tolerance for species with initially zero mole numbers; for LSODE: local relative error tolerance for all components; for CHEMEQ and CREK1D: local relative convergence criterion for all components.
ERMAX	relative error tolerance for Newton-Raphson iteration for temperature
EWT_i	local error weight for i^{th} component, used by LSODE (eqs. (19) and (20))
e_i	estimated global error in i^{th} component (eq. (24))
$e_{i,j}$	estimated global error for i^{th} species of type j
e_{rms}	root-mean-square norm of the estimated global errors for all variables (eq. (27))
$e_{\text{rms},j}$	root-mean-square norm of the estimated global errors for species of type j (eq. (26))
e_T	estimated global error in temperature (eq. (25))
$\mathfrak{F}$	generalized algorithm for Y_n (eq. (14))
f_i	time rate of change of i^{th} component, units depend on i
H_0	initial mixture mass-specific enthalpy, J/kg
$\hat{h}_i$	molar-specific enthalpy of species i, J/kmole
h_n	step size used on the n^{th} step, s
h_0	initial step length to be attempted by integrator, s
IERROR	error control indicator for EPISODE
ITOL	error control indicator for LSODE
k_j, k_{-j}	forward and reverse rate coefficients for reaction j (eqs. (6) and (7)), units depend on reaction type
N	total number of first-order ordinary differential equations
N_j, N_{-j}	temperature exponent in forward and reverse rate coefficients for reaction j (eqs. (6) and (7))
N_R	total number of elementary chemical reactions in reaction mechanism
N_S	total number of chemical species in reacting gas mixture
P1,P2	test problems 1 and 2, respectively
p	pressure, $\mathrm{N/m^2}$
R	universal gas constant in cal/mole K
R_a	universal gas constant in J/kmole K
R_j, R_{-j}	molar forward and reverse rates per unit volume for reaction j (eqs. (4) and (5)), $\mathrm{kmole/m^3\ s}$
RTOL	local relative error tolerance for LSODE
T	temperature, K
ΔT	maximum temperature change allowed before reaction rate coefficients and thermodynamic properties are updated in CREK1D, K
T_{ST}	standard solution value for temperature, K
TINY	minimum species mole number values allowed in CHEMEQ and CREK1D
t	reaction time, s
t_{end}	final time (≥ 1 ms) at which numerical solution is generated, s
t_n	time reached on the n^{th} integration step, s
t_0	initial time, s
V	reacting gas velocity, m/s
X_i	chemical symbol for i^{th} species
x_i	mole fraction of species i
$x_{i,st}$	standard solution value for mole fraction of species i
$x_{\min}$	mole fraction value corresponding to $\sigma_{\min}$ (eq. (33))
$Y_{i,n}$	numerical solution of the i^{th} component at t_n, units depend on i
$Y_{i,n}^{[m]}$	value obtained for $Y_{i,n}$ on m^{th} iteration, units depend on i

$Y_{i,n}^{[m]}$ value obtained for $Y_{i,n}$ on m^{th} iteration, units depend on i

$Y_{i,n}^{[0]}$ predicted value of $Y_{i,n}$, units depend on i

$Y_{i,n}^{*}$ numerical solution of the i^{th} component at t_n generated by using exact past values, units depend on i

$Y_{\max,i}$ local weight for i^{th} component, used by EPISODE and GCKP84 (eqs. (17), (18), and (21))

y_i exact solution for the i^{th} component, units depend on i

α constant in generalized algorithm for Y_n (eq. (14))

ϵ_n global error at t_n (eq. (15))

ν_{ij}', ν_{ij}'' stoichiometric coefficients of species i in forward and reverse reaction j (eq. (1)); number of kilomoles of species i in elementary reaction j as a reactant and as a product, respectively

ρ mixture mass density, kg/m^3

σ_i mole number of species i, kmole species i/kg mixture

$\sigma_{\min}$ mole number value at which local error control in LSODE is equally relative and absolute (eq. (32))

Governing Differential and Algebraic Equations

The ordinary differential equations describing homogeneous gas-phase chemical reactions of the type

$$\sum_{i=1}^{N_S} \nu_{ij}' X_i \rightleftharpoons \sum_{i=1}^{N_S} \nu_{ij}'' X_i \qquad j = 1,\ldots,N_R \tag{1}$$

are as follows:

$$\frac{d\sigma_i}{dt} = f_i(\sigma_k, T) \qquad i,k = 1,\ldots,N_S \tag{2}$$

$$\sigma_i(t_0) = \text{given}$$

$$T(t_0) = \text{given}$$

where f_i, the total formation rate of species i, is given by

$$f_i = -\rho^{-1} \sum_{j=1}^{N_R} (\nu_{ij}' - \nu_{ij}'')(R_j - R_{-j}) \tag{3}$$

The molar reaction rates per unit volume, R_j and R_{-j}, are given by the law of mass action (e.g., ref. 16):

$$R_j = k_j \prod_{l=1}^{N_S} (\rho\sigma_l)^{\nu_{lj}'} \tag{4}$$

and

$$R_{-j} = k_{-j} \prod_{l=1}^{N_S} (\rho\sigma_l)^{\nu_{lj}''} \tag{5}$$

where the forward (k_j) and reverse (k_{-j}) rate coefficients are given by the modified Arrhenius expressions:

$$k_j = A_j T^{N_j} \exp(-E_j/RT) \tag{6}$$

$$k_{-j} = A_{-j} T^{N_{-j}} \exp(-E_{-j}/RT) \tag{7}$$

In equations (1) to (7), ν_{ij}' and ν_{ij}'' are the stoichiometric coefficients of species i (with chemical symbol X_i) in reaction j as a reactant and as a product, respectively; N_S is the total number of distinct chemical species (reacting and inert) in the gas mixture; N_R is the total number of independent reactions in the mechanism; σ_i is the mole number of species i (in kilomoles of species i per kilogram of mixture); t is the time (in seconds); ρ is the gas mixture mass density (in kilograms per cubic meter); T is the temperature (in kelvins); R is the universal gas constant (in calories per mole per kelvin); and A_j, A_{-j}, N_j, N_{-j}, E_j, and E_{-j} are constants in the modified Arrhenius expressions for k_j and k_{-j}. The reverse rate coefficient parameters are calculated from the forward rate coefficient parameters and the concentration equilibrium constants by using the principle of detailed balancing (ref. 16).

In this paper, as in the companion paper (ref. 2), attention is restricted to adiabatic, constant-pressure chemical reactions. For such problems, the following enthalpy conservation equation constitutes an algebraic constraint on equations (2) to (7):

$$\sum_{i=1}^{N_S} \sigma_i \hbar_i = H_0, \tag{8}$$

where $\hbar_i$ is the molar-specific enthalpy of species i (in joules per kilomole) and H_0 is the initial mixture mass-specific enthalpy (in joules per kilogram). Equation (8) can be differentiated with respect to time to give the following ODE for the temperature

$$\frac{dT}{dt} = -\frac{\sum_{i=1}^{N_S} f_i \hbar_i}{\sum_{i=1}^{N_S} \sigma_i c_{p,i}} \tag{9}$$

where $c_{p,i}$ is the constant-pressure molar-specific heat of species i (in joules per kilomole per kelvin). Either equation (8) or (9) can be included in the equation set. We explore the use of both these equations and examine their effects on solution accuracy and computational cost.

The mass density of the mixture is given by the ideal gas equation of state

$$\rho = p/(R_u T \sigma_m) \tag{10}$$

where p is the absolute pressure (in newtons per square meter), R_u is the universal gas constant (in joules per kilomole per kelvin), and σ_m, the reciprocal of the mean molar mass of the mixture, is given by

$$\sigma_m = \sum_{i=1}^{N_s} \sigma_i \tag{11}$$

For constant pressure problems the following density ODE can be obtained from equation (10) by differentiating it with respect to time and then rearranging terms in the resulting expression:

$$\frac{d\rho}{dt} = -\rho\left(\frac{1}{T}\frac{dT}{dt} + \frac{1}{\sigma_m}\sum_{i=1}^{N_s}\frac{d\sigma_i}{dt}\right) \tag{12}$$

Either equation (10) or (12) can be used to compute the density. The code GCKP84, which allows the pressure to vary, solves for ρ by integrating its ODE (eq. (12)). With the other codes, however, we obtain ρ by using equation (10). Indeed, ρ is implicitly replaced by the right-hand side of equation (10) and does not appear as a variable. We, therefore, exclude density from our discussion, including statement of the problem, and restrict attention to solving for the other $N_S + 1$ quantities.

Problem Statement

The initial value problem may be stated as follows: Given (1) at time $t = t_0$, values for the species mole numbers, σ_i ($i = 1,\ldots,N_S$), and the temperature, T, (2) the pressure, p, and (3) the reaction mechanism, find, at the end of a prescribed time interval, the mixture composition and temperature.

Methods and Codes Examined

The codes examined in this study include the general-purpose packages EPISODE and LSODE (refs. 6 to 9) and the specialized techniques CHEMEQ (ref. 10), CREK1D (refs. 11 to 14), and GCKP84 (ref. 15). The methods used in these codes are summarized below and are discussed in detail in appendix A.

The packages EPISODE and LSODE consist of a variable-order, variable-step implicit Adams method (suitable for nonstiff problems) and a variable-order, variable-step backward differentiation formula method (suitable for stiff problems; e.g., refs. 1 and 17). Both methods use a standard predictor and a variety of corrector formulas—from functional iteration to a modified Newton iteration—is included. The Jacobian matrix $\partial f/\partial y$, where y is the vector of dependent variables and $f = dy/dt$, is computed either numerically or with a user-supplied subroutine. In part I of this investigation (ref. 2) all options relevant to the problem of chemical kinetics were attempted, and the stiff method with Newton iteration and user-supplied analytical Jacobian matrix was found to be the fastest. Therefore, only this option is used in examining the accuracy of EPISODE and LSODE.

The general chemical kinetics program GCKP84 uses the integration technique developed by Zeleznik and McBride (ref. 18). The algorithm is essentially a revised version of the GEAR package (ref. 19), which contains the same two integration methods as EPISODE and LSODE and several iteration techniques. For reasons given above we restrict attention to the stiff method with Newton iteration using an analytical Jacobian matrix. GCKP84 includes corrective actions if the physically impossible situation of negative concentrations, temperature, density, or velocity arises.

In CHEMEQ, at the start of each time step the ODE's are separated into two classes: stiff and normal. For equations classified as normal, a classical second-order predictor-corrector method, the trapezoidal rule, is used. For the stiff equations a simple stable asymptotic integration formula is used.

The code CREK1D is based on the exponentially fitted trapezoidal rule developed by Liniger and Willoughby (ref. 20) and Brandon (refs. 21 and 22). This code includes special treatment of ill-posed initial conditions and automatic selection of Jacobi-Newton iteration or Newton iteration.

Error Considerations

In this section the error controls used in the different codes examined are discussed. In general, numerical methods replace the differential equations with difference equations and solve them step by step. Starting with the known initial conditions $y(t_0)$ at t_0 numerical approximations Y_n to the exact solution $y(t_n)$ of the ODE's are generated at discrete points in time (t_n ($n = 1,2,\ldots$)), until the end of the integration interval is reached. At each t_n the numerical method provides a rule for generating the approximate solution Y_n in terms of computed quantities at one or more previous times.

For the scalar differential equation,

$$\left.\begin{aligned} \frac{dy}{dt} &= f(y) \\ y(t_0) &= \text{given} \end{aligned}\right\} \tag{13}$$

the algorithms used for Y_n in all of the codes can be written as

$$Y_n = \alpha Y_{n-1} + h_n \mathfrak{F}(h_n, Y_n, f_n, Y_{n-1}, f_{n-1}, \ldots) \qquad (14)$$

where α is a constant, Y_{n-j} is the approximate solution at t_{n-j}, h_n (equal to $t_n - t_{n-1}$) is the step size used on the step $[t_{n-1}, t_n]$ and $f_{n-j} = f(Y_{n-j})$. Because equation (14) involves the unknown quantity Y_n its solution generally requires an iterative procedure. Starting with the predicted value (an initial guess), denoted by $Y_n^{[0]}$, improved estimates $Y_n^{[m]}$ $(m = 1,2,\ldots)$ are generated until the iteration converges, that is, until the difference in two successive approximations approaches zero within a specified accuracy.

During the calculation procedure, errors called discretization or truncation errors are introduced into the numerical solution because of the approximation of the ODE's by difference equations. Two measures can be defined for this error, which is a property of the numerical method (e.g., ref. 23). The global discretization error ϵ_n at any t_n is the difference between the computed approximation Y_n and the exact solution $y(t_n)$:

$$\epsilon_n = Y_n - y(t_n) \qquad (15)$$

It is the quantity that the user wants to know and control. The local truncation error d_n at t_n is the error in the numerical approximation Y_n^* that is generated on the step $[t_{n-1}, t_n]$ by using exact past values:

$$d_n = Y_n^* - y(t_n) \qquad (16)$$

It is the quantity that ODE solvers generally control. The two discretization errors are illustrated in figure 1 for a single ODE.

The codes examined in this study require the user to specify values for one or more local tolerance parameters, which control the accuracy of the numerical solution. Now, as discussed below, the same error control is not used in all codes. Nevertheless, for convenience, for all codes the same notation, EPS, is used to denote the local tolerance quantity, or the primary one if several are required. In EPISODE the local truncation error vector d_n satisfies the inequality

$$\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{d_{i,n}}{Y_{\max_i}} \right)^2 \right)^{1/2} \leq \text{EPS} \qquad (17)$$

where N is the number of ODE's, $d_{i,n}$ is the estimated local truncation error in the i^{th} component at t_n, and for the error control used

$$Y_{\max_i} = \max \{|Y_{i,n-1}|, |Y_{i,n-2}|\} \quad \text{for } i \text{ that satisfy } y_i(t_0) \neq 0,$$
$$= \max \{1, |Y_{i,n-1}|\} \quad \text{for } i \text{ that satisfy } y_i(t_0) = 0, \qquad (18)$$

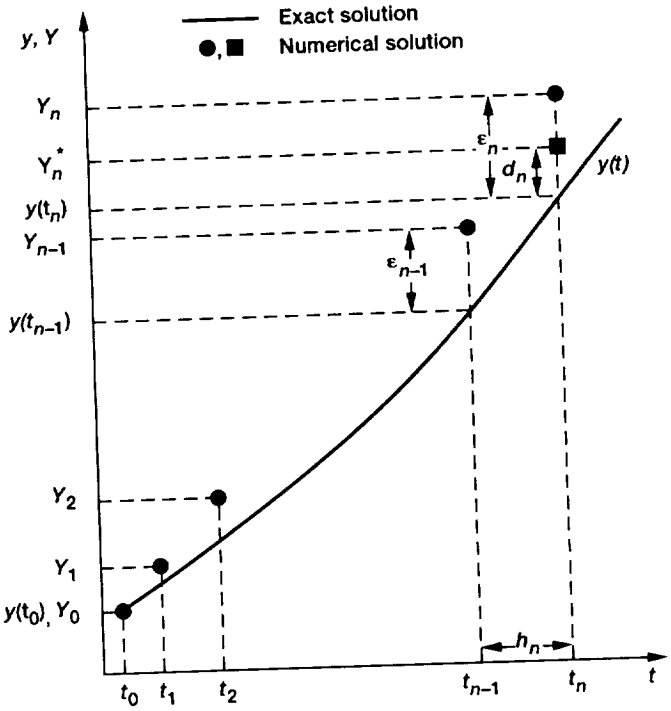

Figure 1.—Numerical solutions and truncations error types for the single ODE $dy/dt = f(y)$. The exact solution to the ODE is denoted by $y(t)$. The numerical solutions obtained with the initial condition $Y_0 = y(t_0)$ are denoted by solid circles. The solid square denotes the numerical solution obtained at t_n by using exact past values. The local truncation error is denoted by d, the global truncation error by ϵ, and the step length by h.

where the vertical bars denote absolute value. The error control selected to be performed by LSODE is given by

$$\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{d_{i,n}}{EWT_i} \right)^2 \right)^{1/2} \leq 1 \qquad (19)$$

where

$$\text{EWT}_i = \text{EPS}\,|Y_{i,n-1}| + \text{ATOL}_i \qquad (20)$$

where ATOL_i is the user-supplied local absolute error tolerance for the i^{th} component.

In GCKP84 the local error test satisfies the inequality

$$\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{E_{i,n}}{Y_{\max_i}} \right)^2 \right)^{1/2} \leq C_G\,\text{EPS} \qquad (21)$$

where, $E_{i,n}$ contains the cumulative difference between the converged and predicted values of the derivative (dY_i/dt) at t_n and where C_G is a constant. The quantity $Y_{\max,i}$ has the same meaning as in EPISODE (see eq. 18)).

The codes CHEMEQ and CREK1D do not control the estimated local truncation error. The solution is accepted when the magnitude of the normalized difference in successive

estimates $(Y_n^{[m+1]} - Y_n^{[m]})$ is less than a specified amount. Therefore, these codes control only the error in the solution to the difference equation of the method. In CHEMEQ each component i satisfies the inequality

$$\frac{|Y_{i,n}^{[m+1]} - Y_{i,n}^{[m]}|}{\min\{|Y_{i,n}^{[m+1]}|, |Y_{i,n}^{[m]}|\}} \leq \text{EPS} \tag{22}$$

The convergence criterion used in CREK1D is given by

$$\left(\frac{1}{N_S}\sum_{i=1}^{N_S}\left(\frac{Y_{i,n}^{[m+1]} - Y_{i,n}^{[m]}}{Y_{i,n}^{[m]}}\right)^2\right)^{1/2} \leq \text{EPS} \tag{23}$$

It is clear from the above discussion that the user-supplied local tolerance EPS does not have the same meaning for all codes. In LSODE it is the local relative error tolerance for all variables and is a measure of the number of accurate significant figures in the numerical solution. In EPISODE and GCKP84, however, as discussed in the section "Computational Procedure," EPS is the local relative error tolerance for only variables with nonzero initial values, such as the temperature. For species with zero initial mole numbers EPS is the local absolute error tolerance and is a measure of the largest number that may be neglected. In contrast to these three codes, CHEMEQ and CREK1D do not control the local truncation error, and EPS is the local relative convergence criterion, or error in the solution to the difference equation. However, as described in appendix A, although CREK1D does not test that the estimated local truncation error is within a prescribed bound, the step length calculation procedure attempts to achieve this result. The step length to be attempted next is selected such that the current estimate of the local truncation error normalized by the solution is at most equal to EPS. Because of these differences in the meanings of EPS it will be referred to as simply the local tolerance.

Evaluation of Temperature

Of the codes examined in the present study, only GCKP84 and CREK1D were written explicitly for nonisothermal chemical reactions. These methods, therefore, have built-in procedures for calculating the temperature. For the other codes, however, the temperature has to be calculated along with the mixture composition. In the present study (as in ref. 2), the temperature was computed using one of the two methods outlined below.

In method A the temperature was calculated from the initial mixture mass-specific enthalpy H_0 and the solution for the species mole numbers returned by the integrator by using the algebraic enthalpy conservation equation (8). This equation was solved for the temperature by using a Newton-Raphson iterative technique, with a user-supplied local relative error tolerance, ERMAX (as described in appendix B). In this method, the temperature is not an explicit dependent variable, so the number of ODE's is equal to the number (N_S) of species and the Jacobian matrix is of size $N_S \times N_S$. The integrator, therefore, tracks only the solution for the species mole numbers. The temperature was also computed when the species time derivatives and the Jacobian matrix were evaluated.

In method B the temperature was treated as an additional dependent variable and evaluated by solving its ODE (eq. (9)). In this method, the number of ODE's is equal to $N_S + 1$, the Jacobian matrix is of size $(N_S + 1) \times (N_S + 1)$, and the integrator tracks the solutions for both the species mole numbers and the temperature.

The following naming convention was adopted. Techniques using method A were given the suffix A (EPISODE-A, etc.), and those using method B were given the suffix B (EPISODE-B, etc.).

The code GCKP84 allows for heat transfer between the reacting gas mixture and its surroundings and must therefore use an ODE to solve for the temperature. It also includes the density and velocity, V, of the gas mixture as dependent variables and evaluates them by integrating their ODE's. (For the static test problems used in this study the velocity ODE is given trivially by $dV/dt = 0$, $V(t_0) = 0$.) Consequently, the number of ODE's solved by GCKP84 is equal to $N_S + 3$, and the Jacobian matrix is of size $(N_S + 3) \times (N_S + 3)$.

CREK1D computes the temperature by solving the algebraic enthalpy conservation equation (8). However, the calculation procedure is different from that used in method A. In CREK1D the mixed differential-algebraic system of equations (2) and (8) is solved simultaneously, whereas method A solves equation (8) after the species ODE's have been integrated over a time step. Thus, although the number of ODE's solved by CREK1D is equal to N_S, the Jacobian matrix is of size $(N_S + 1) \times (N_S + 1)$.

Test Problems

The algorithms examined in the present study were applied to the same two test problems used in our previous work (ref. 2). Both problems describe adiabatic, constant pressure, transient, batch chemical reactions and include all three combustion regimes: induction, heat release, and equilibration.

Test problem 1 describes the ignition and subsequent combustion of a mixture of 33 percent carbon monoxide and 67 percent hydrogen with 100 percent theoretical air at an initial temperature of 1000 K and a pressure of 10 atm. It comprises 12 reactions which describe the temporal evolution of 11 reacting species (CO, CO_2, H, H_2, H_2O, N, NO, N_2, O, OH, and O_2). Test problem 2 describes the ignition and subsequent combustion of a stoichiometric hydrogen-air mixture at a pressure of 2 atm and an initial temperature of 1500 K. It involves 30 reactions among 15 species (Ar, CO_2, H, HO_2, H_2, H_2O, H_2O_2, N, NO, NO_2, N_2, N_2O, O, OH, and O_2), of which two (Ar and CO_2) are inert. The reaction mechanisms and forward rate coefficient parameters for the two test problems are given in tables I and II.

TABLE I.—REACTION MECHANISM AND FORWARD RATE COEFFICIENT PARAMETERS USED FOR TEST PROBLEM 1

[Rate coefficient $k_j = 10^{B_j} T^{N_j} \exp(-E_j/RT)$.]

Reaction number, j	Reaction	Rate coefficient parameters		
		B_j	N_j	E_j, kcal/mole
1	$CO + OH = CO_2 + H$	11.49	0	0.596
2	$H + O_2 = O + OH$	14.34	↓	16.492
3	$H_2 + O = H + OH$	13.48	↓	9.339
4	$H_2O + O = OH + OH$	13.92	↓	18.121
5	$H + H_2O = H_2 + OH$	14.0	↓	19.870
6	$N + O_2 = NO + O$	9.81	1.0	6.250
7	$N_2 + O = N + NO$	13.85	0	75.506
8	$NO + M = N + O + M$	20.60	−1.5	149.025
9	$H + H + M = H_2 + M$	18.00	−1.0	0
10	$O + O + M = O_2 + M$	18.14	−1.0	0.340
11	$H + OH + M = H_2O + M$	23.88	−2.6	0
12	$H_2 + O_2 = OH + OH$	13.00	0	43.000

TABLE II.—REACTION MECHANISM AND FORWARD RATE COEFFICIENT PARAMETERS USED FOR TEST PROBLEM 2

[Rate coefficient $k_j = 10^{B_j} T^{N_j} \exp(-E_j/RT)$.]

Reaction number, j	Reaction	Rate coefficient parameters		
		B_j	N_j	E_j, kcal/mole
1	$H + O_2 = OH + O$	14.342	0	16.790
2	$O + H_2 = OH + H$	10.255	1.0	8.900
3	$H_2 + OH = H_2O + H$	13.716	0	6.500
4	$OH + OH = O + H_2O$	12.799	↓	1.093
5	$H + O_2 + M = HO_2 + M$	15.176	↓	−1.000
6	$O + O + M = O_2 + M$	13.756	↓	−1.788
7	$H + H + M = H_2 + M$	17.919	−1.0	0
8	$H + OH + M = H_2O + M$	21.924	−2.0	0
9	$H_2 + HO_2 = H_2O + OH$	11.857	0	18.700
10	$H_2O_2 + M = OH + OH + M$	17.068	↓	45.500
11	$H_2 + O_2 = OH + OH$	13.000	↓	43.000
12	$H + HO_2 = OH + OH$	14.398	↓	1.900
13	$O + HO_2 = OH + O_2$	13.699	↓	1.000
14	$OH + HO_2 = H_2O + O_2$	13.699	↓	1.000
15	$HO_2 + HO_2 = H_2O_2 + O_2$	12.255	↓	0
16	$OH + H_2O_2 = H_2O + HO_2$	13.000	↓	1.800
17	$O + H_2O_2 = OH + HO_2$	13.903	↓	1.000
18	$H + H_2O_2 = H_2O + OH$	14.505	↓	9.000
19	$HO_2 + NO = NO_2 + OH$	13.079	↓	2.380
20	$O + NO_2 = NO + O_2$	13.000	↓	0.596
21	$NO + O + M = NO_2 + M$	15.750	↓	−1.160
22	$NO_2 + H = NO + OH$	14.462	↓	0.795
23	$N + O_2 = NO + O$	9.806	1.0	6.250
24	$O + N_2 = NO + N$	14.255	0	76.250
25	$N + OH = NO + H$	13.602	↓	0
26	$N_2O + M = N_2 + O + M$	14.152	↓	51.280
27	$O + N_2O = N_2 + O_2$	13.794	↓	24.520
28	$O + N_2O = NO + NO$	13.491	↓	21.800
29	$N + NO_2 = NO + NO$	12.556	↓	0
30	$OH + N_2 = N_2O + H$	12.505	↓	80.280

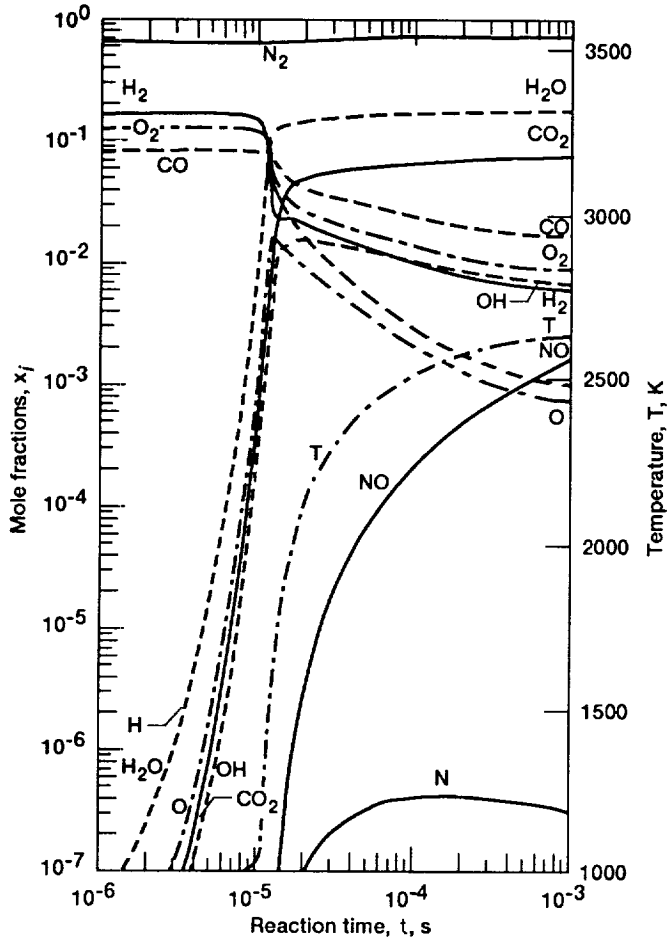

Figure 2.—Variation with reaction time of temperature and species mole fractions for test problem 1.

Figure 3.—Variation with reaction time of temperature and species mole fractions for test problem 2.

Figures 2 and 3 present the variations with time of the chemical species mole fractions and temperature for test problems 1 and 2, respectively. These solutions were generated with LSODE-B using a small value (10^{-5}) for the local relative error tolerance. Both test problems were integrated over a time interval of 1 ms in order to obtain near-equilibration of all chemical species and the temperature.

Computational Procedure

For each method, global errors in solutions generated with a certain value for the local tolerance EPS were estimated by comparing them with results obtained with the same method and a reduced tolerance. The solutions used as a basis for comparison were the most accurate generated and are referred to as standard solutions. For example, for CREK1D solutions used as standards were generated with CREK1D and EPS = 10^{-5}. These standard solutions were used to estimate the global errors in results produced with CREK1D and EPS = 10^{-2}, 10^{-3} and 10^{-4}. The above procedure for estimating global errors is reliable provided the technique is effective in the sense that reducing the local tolerance actually reduces the global error (ref. 24). In any case, in the absence of exact solutions the only method for assessing the accuracy of an algorithm is to compare the solutions that it produces with those obtained with a reduced local tolerance using either the same algorithm or a different one. The use of solutions generated by each technique as a standard of comparison only for itself ensures that the accuracy comparison is not biased in favor of any one method or code.

A typical computational run was performed by first initializing the time (t, set equal to zero), species mole numbers, and temperature. The integrator was then called with values for the necessary input parameters, including the local tolerance and the elapsed time (equal to 1 ms for both problems) at which the integration was to be terminated. After each step successfully executed by the integrator, the current time and the solutions for the species mole fractions and temperature were saved. This procedure was repeated until the time reached by the integrator was greater than or equal to 1 ms. The saved time

values served as input data for the output stations at which the standard solution was to be generated. At each of these discrete time values global errors in the species mole fractions and temperature were estimated by comparisons with the standard solutions as follows:

$$e_i(t) = \frac{x_i(t)}{x_{i,ST}(t)} - 1 \qquad i = 1,\ldots, N_s \tag{24}$$

$$e_T(t) = \frac{T(t)}{T_{ST}(t)} - 1 \tag{25}$$

where $e_i(t)$ and $e_T(t)$ are, respectively, the estimated global errors in the mole fraction $x_i(t)$ of species i and the temperature $T(t)$ at time t and where $x_{i,ST}(t)$ and $T_{ST}(t)$ are, respectively, the standard solution values for the mole fraction of species i and the temperature at time t. To prevent the possibility of requiring accuracy in species with immeasurably small concentrations, global errors were not measured for species whose standard solution mole fractions were less than 0.1 ppm. For such species $e_i(t)$ was set equal to zero. In this way time histories of the global errors in species mole fractions and temperature were generated.

For each technique, standard solutions were generated with a small value for EPS. In addition to EPS and the elapsed time at which the integration was to be terminated, other input parameters were required by all codes examined. In this paper only those input parameters that affect the accuracy of each code are discussed. A more detailed discussion of these parameters can be found in part I (ref. 2).

The user-supplied parameters relevant to solution accuracy that are required by LSODE are the error control flag, ITOL, which indicates the type of local error control to be performed, and the local relative, RTOL, and absolute, ATOL, error tolerances. Both RTOL and ATOL can be specified either as (1) a scalar, so that the same local error tolerance is used for all variables, or (2) an array, so that different values of the local error tolerance are used for different variables. In the present work the error control given by ITOL = 2 (for scalar RTOL (equal to EPS) and array ATOL, see appendix A) was used for reasons given below. Since the same number of accurate significant figures is acceptable for all solution components, RTOL was specified as a scalar. Now, for the test problems examined in this study, the species mole fractions and temperature vary widely (figs. 2 and 3), so relative error control is appropriate and is the reason for designating the local relative error tolerance as the primary tolerance (eq. (20)). Pure relative error control can be achieved by specifying a value of zero for the local absolute error tolerances. However, since many of the species had zero initial concentrations, pure relative error control could not be used. To make the error control mostly relative, small values were specified for the absolute error tolerances for the species mole numbers; for convenience the same value (equal to ATOLSP) was used for all species. Since the temperature can never be zero, pure relative error control was used for this variable, that is, the local absolute error tolerance for temperature was set equal to zero. Thus, ATOL was specified as an array.

The values used for ATOLSP were those obtained in part I of this study (ref. 2) for LSODE-B as follows: With EPS = 10^{-5}, ATOLSP was progressively decreased until the temperature-time trace showed essentially no change with a further decrease. The values obtained for ATOLSP by using this procedure were 10^{-14} and 10^{-11}, respectively, for test problems 1 and 2. For consistency, the same EPS and ATOLSP were used with LSODE-A. They were, however, checked to ensure that reductions in ATOLSP resulted in essentially the same solutions. For reasons given in the next section ERMAX was set equal to EPS.

To make accuracy comparisons among the codes meaningful, the same value of EPS (i.e., 10^{-5}) was used to generate standard solutions for GCKP84, CHEMEQ-A, CHEMEQ-B, and CREK1D. With EPISODE, however, an EPS value of 10^{-6} was used because larger values produced physically meaningless results for test problem 1—little or no change from initial values after an elapsed time of 1 ms. The error control to be performed by this code is selected by means of the flag IERROR (appendix A). For the reasons given above, pure relative error control (option IERROR = 2) could not be used and the option IERROR = 3 was used, instead. This error control is semirelative (see eqs. (17) and (18)). It is relative for a variable that is initially nonzero. But for a variable that is initially zero, it is absolute until the variable reaches unity in magnitude, when it becomes relative. Since none of the mole numbers attains a value of unity, the error control is always absolute for species with zero initial mole numbers.

The solution generated with EPISODE depended on the value specified for the initial step length (h_0) to be attempted by the integrator. In generating standard solutions with this code, h_0 was progressively decreased (with EPS = 10^{-6}) until the temperature-time trace showed essentially no change with a further decrease. The values obtained for h_0 by using this procedure were 10^{-9} and 10^{-8} s, respectively, for test problems 1 and 2, for both EPISODE-A and EPISODE-B. However, an h_0 value of 10^{-9} s was used for test problem 2 because it resulted in smaller execution times, as shown in table III. For EPISODE-A the savings were modest, but for EPISODE-B they were significant.

GCKP84 uses the same error control as that selected to be performed by EPISODE. It also requires the user to specify h_0. Since details of the integration technique used in GCKP84 were not known, a default value of $h_0 = 10^{-6}$ s was used in our previous work (refs. 2 to 5). However, Bittker and Scullin (ref. 15) have since then set the default value for h_0 at 5×10^{-8} s. Nevertheless an h_0 value of 10^{-6} s was used in this study to be consistent with part I (ref. 2). In addition, as shown in the next section, the 10^{-6} value generally produced more accurate results than the new default value, while requiring comparable execution times for all EPS used in this study.

TABLE III.—EFFECTS OF INITIAL STEP LENGTH ON EXECUTION TIMES REQUIRED BY EPISODE-A AND -B (EPS = 10^{-6}) FOR TEST PROBLEM 2

Method	Initial step length, h_0, s	CPU execution time, s
EPISODE-A	10^{-8}	3.1
	10^{-9}	3.0
EPISODE-B	10^{-8}	14
	10^{-9}	7.8

In contrast to EPISODE and GCKP84, the other codes automatically compute the h_0 value to be attempted by the integrator. In LSODE the calculation procedure for h_0 employs the user-specified values for the first output station and the local error tolerances. The computed initial step length can have an adverse effect on both the computational work and the solution generated by the code (ref. 3). The calculation procedures used for h_0 in CHEMEQ and CREK1D are based on the problem physics (see appendix A) and the computed h_0 did not cause the above difficulties.

Both CHEMEQ and CREK1D use a relative convergence criterion (eqs. (22) and (23)). The difficulty of applying the test when the solution vanishes is avoided by setting mole numbers less than a suitably small value, TINY, to be equal to TINY. In this study a value for TINY of 10^{-20} was used. The only user-specified parameter required by CREK1D that affects its accuracy is ΔT, which is the maximum temperature change allowed before the reaction rate coefficients and the thermodynamic properties k_i and $c_{p,i}$ are updated. Use of this parameter increases the efficiency of numerical techniques in solving combustion kinetic rate equations (refs. 2 and 3). To ensure that the most accurate solutions were used as standards, a value of $\Delta T = 0$ K was used for both test problems.

Results and Discussion

The procedure described in the previous section was used to study the global errors incurred by the different techniques in solving the two test problems. All results presented herein were generated on the NASA Lewis Research Center's IBM 370/3033 computer using single-precision accuracy, except GCKP84 which uses double-precision accuracy.

Both temperature calculation methods A and B were attempted with EPISODE, LSODE, and CHEMEQ. The error control used in method A is pure relative, and the local relative error tolerance is equal to ERMAX (see appendix B). In EPISODE-B and CHEMEQ-B the error or convergence control for the temperature is pure relative and the local relative tolerance is equal to EPS (eqs. (17), (18), and (22)). For reasons given previously in the section "Computational Procedure" the above remarks apply to LSODE-B also. To make accuracy comparisons between the two temperature calculation methods meaningful, ERMAX was set equal to EPS, thereby imposing the same local accuracy requirements on both methods. Thus, both methods A and B used the same error control (i.e., pure relative) and the same local tolerance.

To facilitate accuracy comparisons among the different techniques, the species were divided into three types: reactants (R), intermediates (I), and products (P). At each discrete time at which global errors had been computed, root-mean-square (rms) errors, $e_{\mathrm{rms},j}(t)$ ($j = R,I,P$), were computed for all three species types as follows:

$$e_{\mathrm{rms},j}(t) = \left(\frac{1}{N_j} \sum_{i=1}^{N_j} e_{i,j}^2(t) \right)^{1/2} \qquad j = R,\ I,\ P \tag{26}$$

where $e_{i,j}(t)$ ($i = 1,\ldots,N_j$) is the estimated global error at time t in the mole fraction of the i^{th} species of type j and where N_j is the number of species of type j. The values of N_j (and the species that comprise each subset) are as follows: For test problem 1, $N_R = 4$ (CO, H_2, N_2, and O_2), $N_I = 4$ (H, N, O, and OH), and $N_P = 3$ (CO_2, H_2O and NO). For test problem 2, $N_R = 5$ (Ar, CO_2, H_2, N_2, and O_2), $N_I = 6$ (H, HO_2, H_2O_2, N, O, and OH), and $N_P = 4$ (H_2O, NO, NO_2, and N_2O). Although Ar and CO_2 are inert species, so that their mole numbers do not change during the course of the reaction, they are classified as reactant species because they participate in three-body reactions as catalysts and their concentrations affect the rates of these reactions.

In addition to the rms error for each species type, a single rms error for all variables, $e_{\mathrm{rms}}(t)$, was computed at each time t by using

$$e_{\mathrm{rms}}(t) = \left(\frac{\sum_{i=1}^{N_S} e_i^2(t) + e_T^2(t)}{N_S + 1} \right)^{1/2} \tag{27}$$

Figures 4 to 9 present the variations with time of the percent rms error in reactants, intermediates, and products and the percent error in temperature for test problem 1. Similar information is presented for test problem 2 in figures 10 to 17. For brevity, test problems 1 and 2 are hereinafter referred to as P1 and P2, respectively. Note that for clarity the actual errors have been magnified in some of the figures. The maximum percent rms errors incurred and the reaction times at which they occurred are given in tables IV and VI, along with the values used for the input parameters discussed in the previous section. For each code (except GCKP84) and EPS, these input parameters, obtained in part I of this study (ref. 2) by a trial-and-error procedure, minimized the execution time required to solve the problem. To prevent the possibility of generating physically meaningless results by using too large a value of

Continued on p. 29

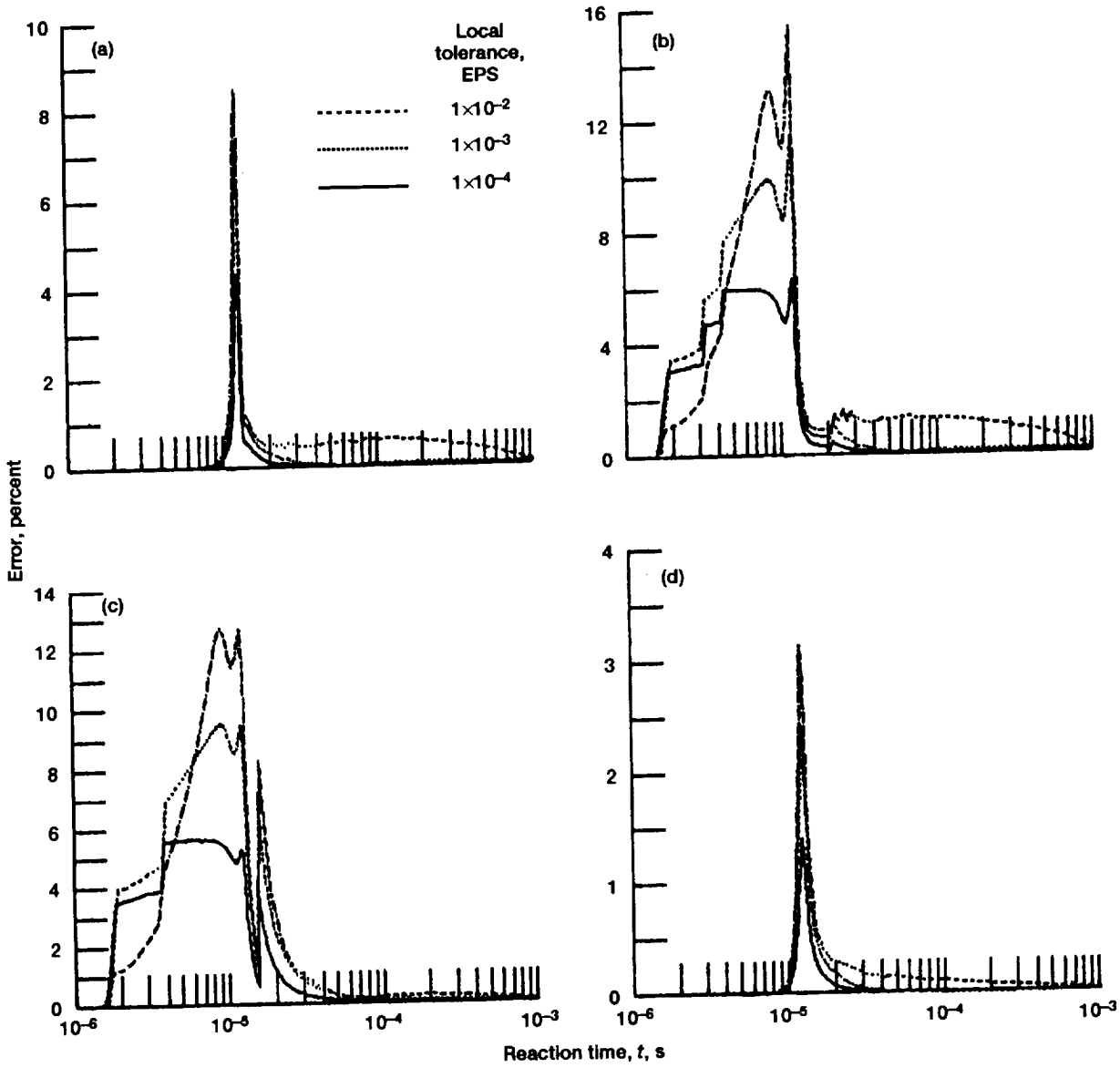

Figure 4.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 1 results generated using LSODE-A.

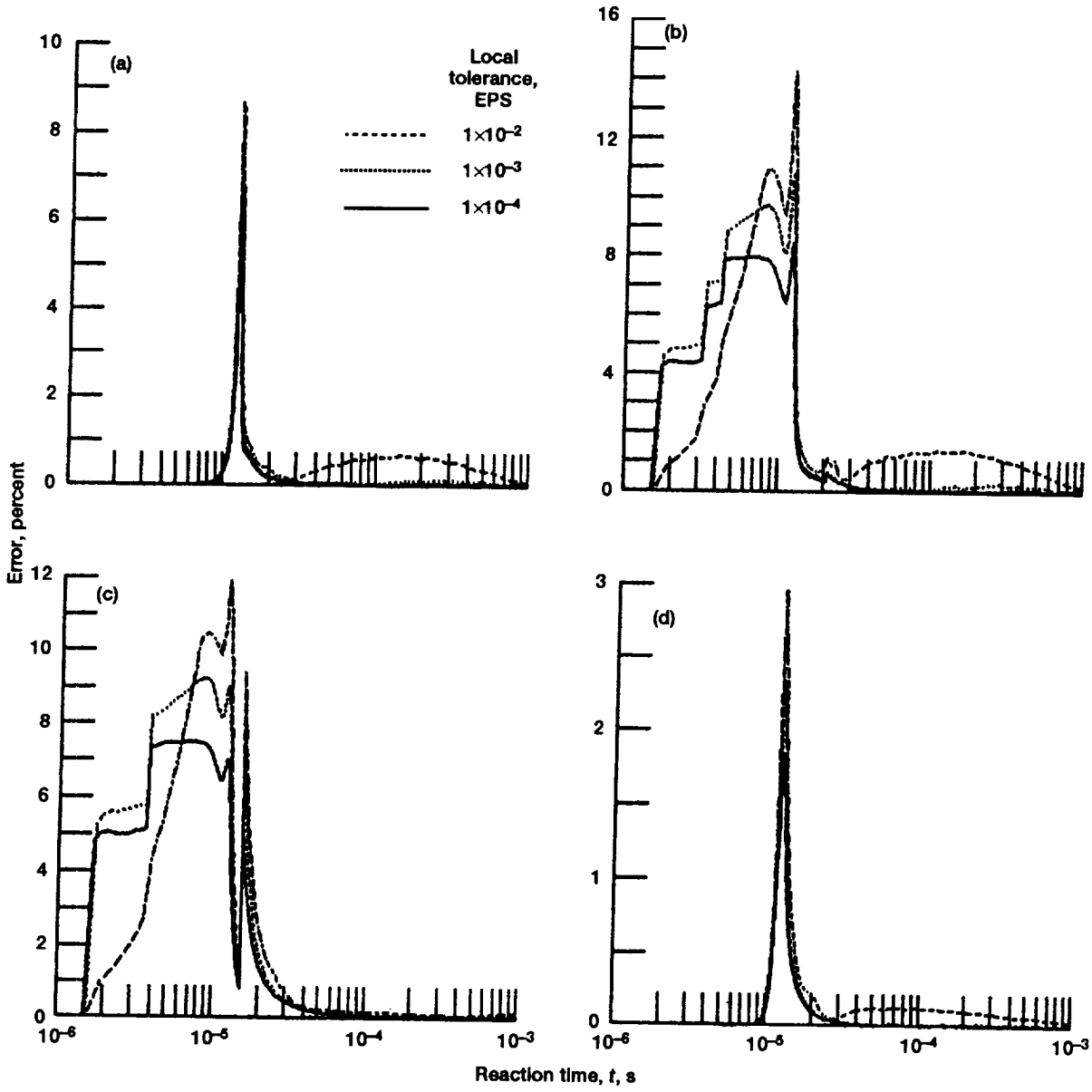

Figure 5.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 1 results generated using LSODE–B.

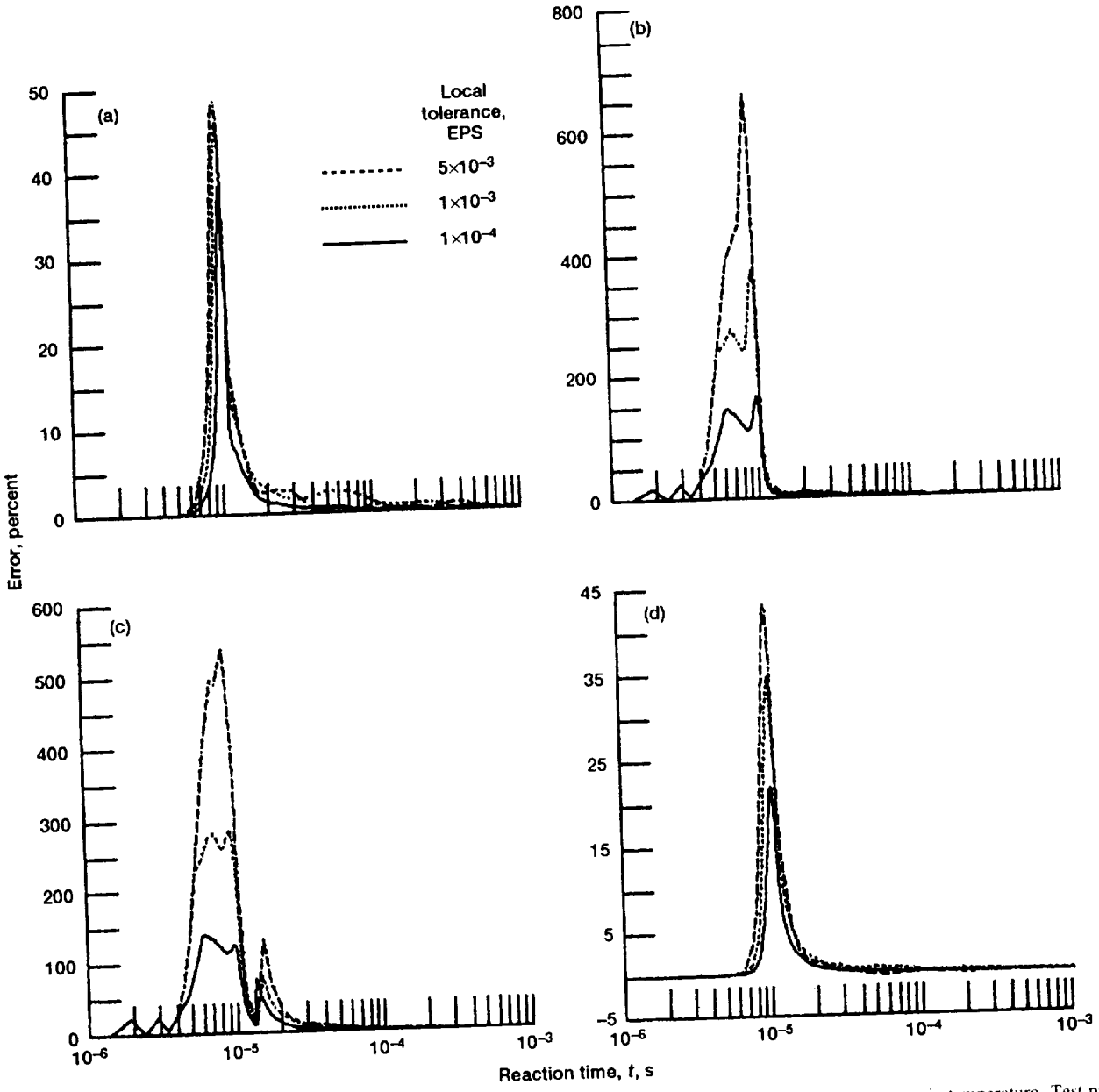

Figure 6.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 1 results generated using GCKP84.

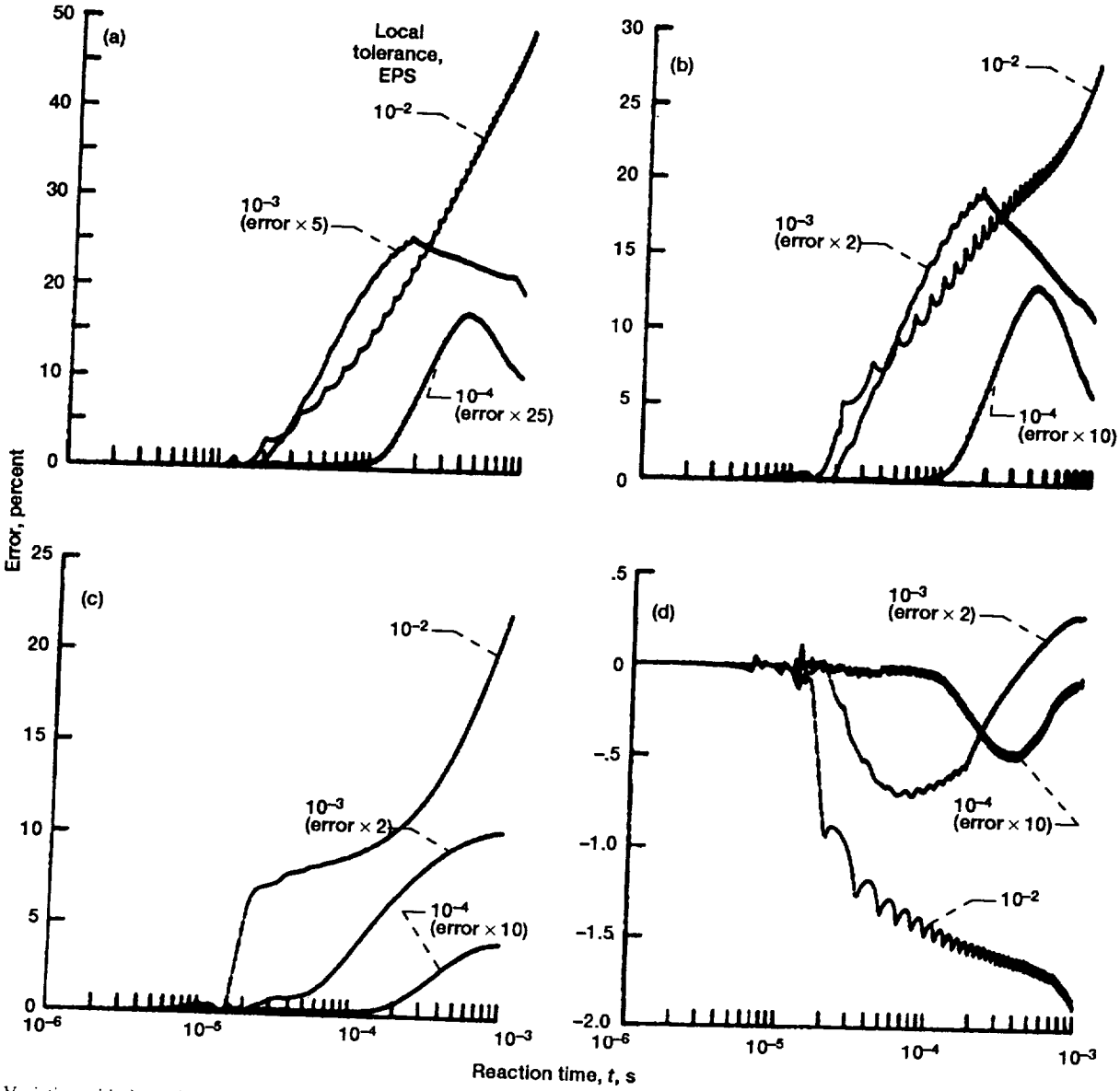

Figure 7.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 1 results generated using CHEMEQ–A.

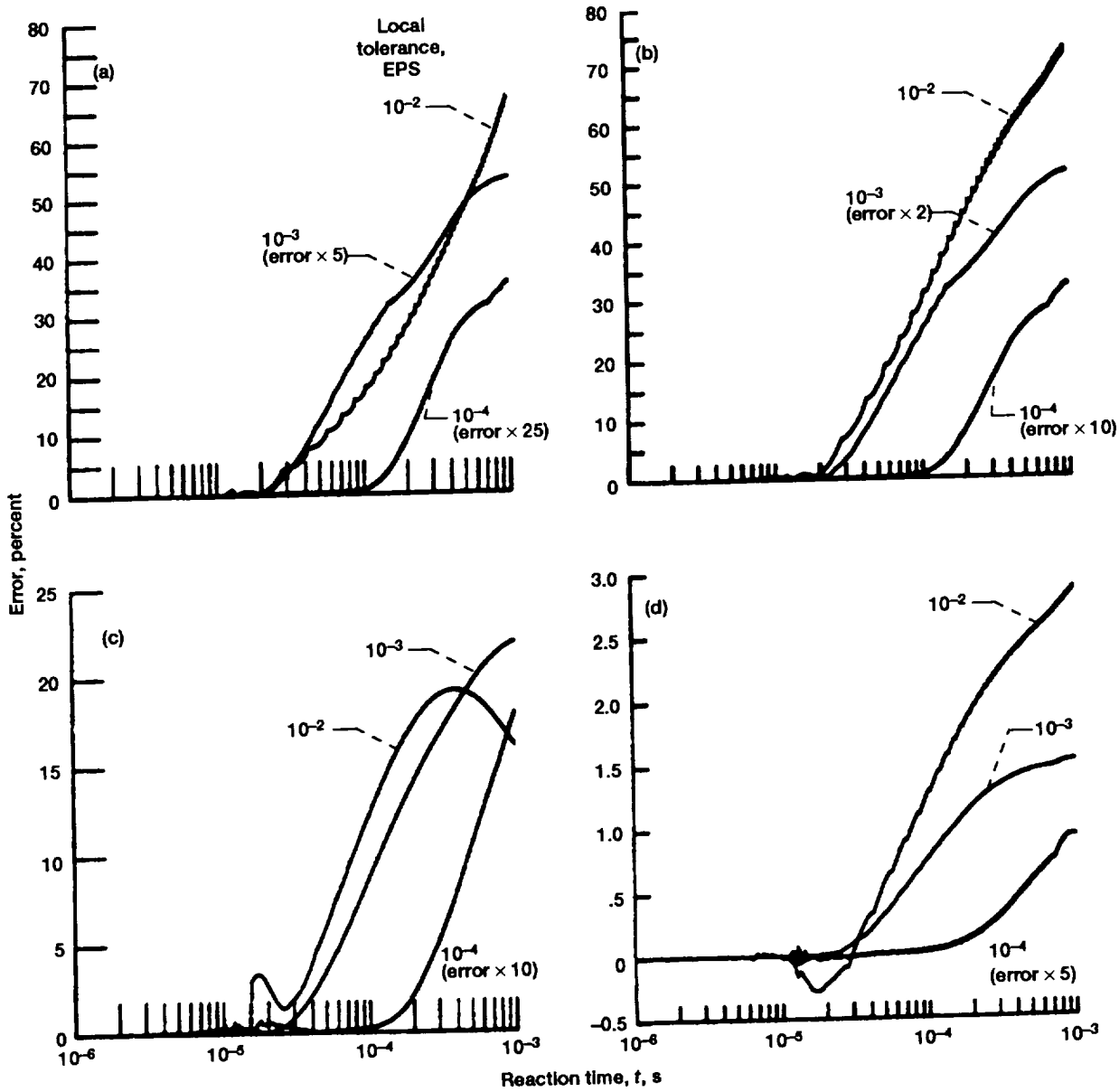

Figure 8.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 1 results generated using CHEMEQ–B.

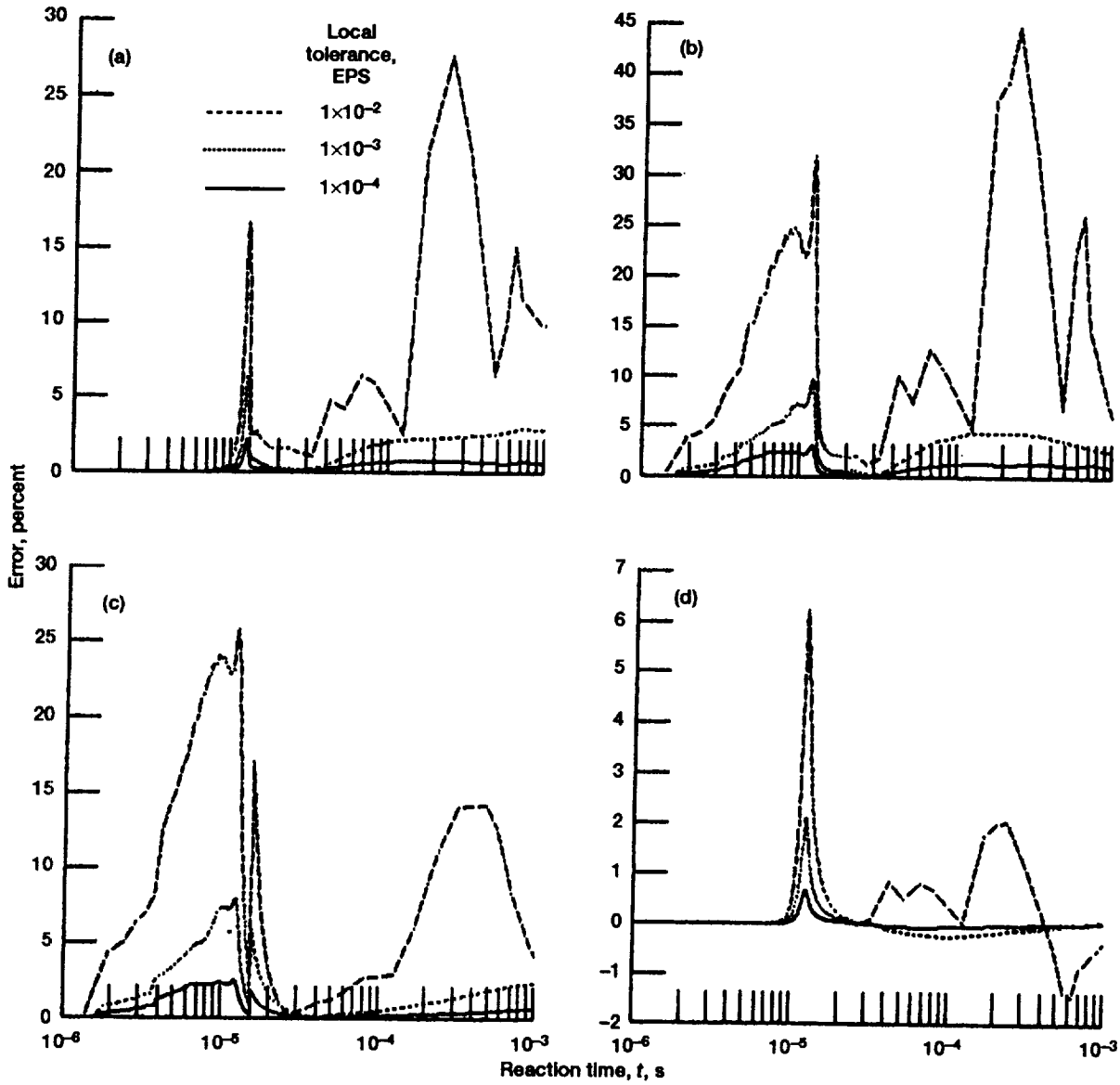

Figure 9.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 1 results generated using CREK1D.

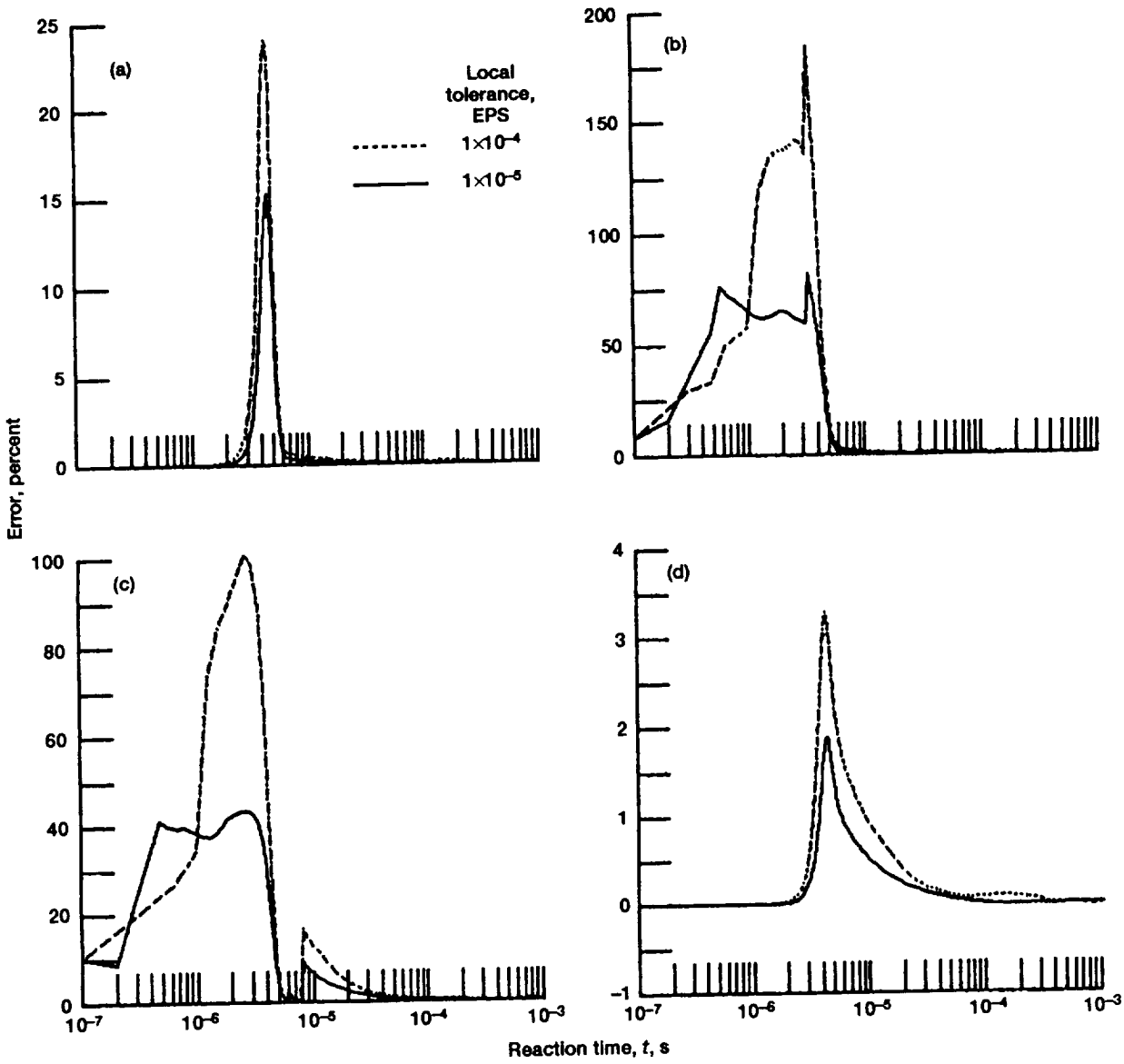

Figure 10.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using EPISODE-A.

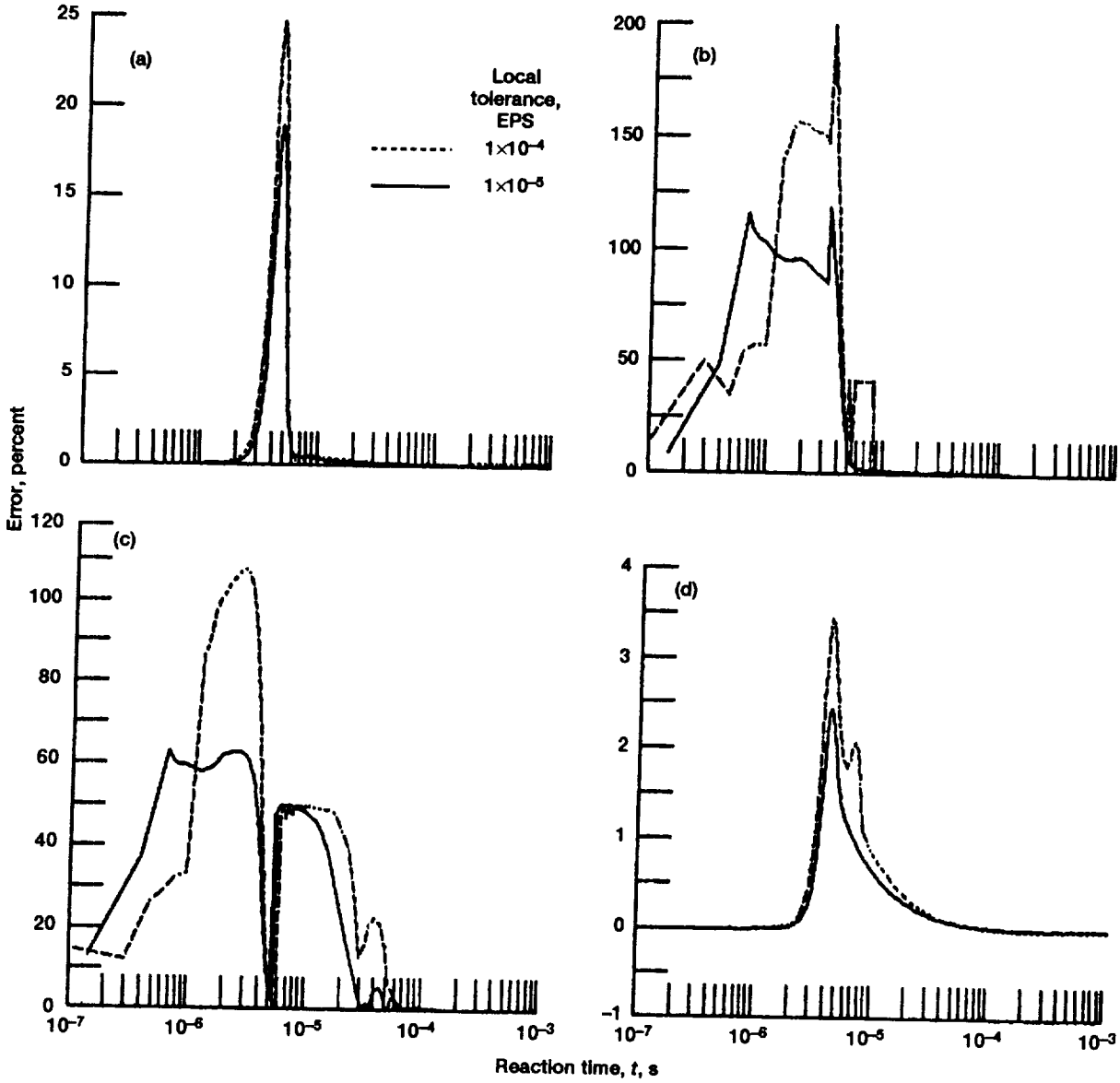

Figure 11.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using EPISODE-B.

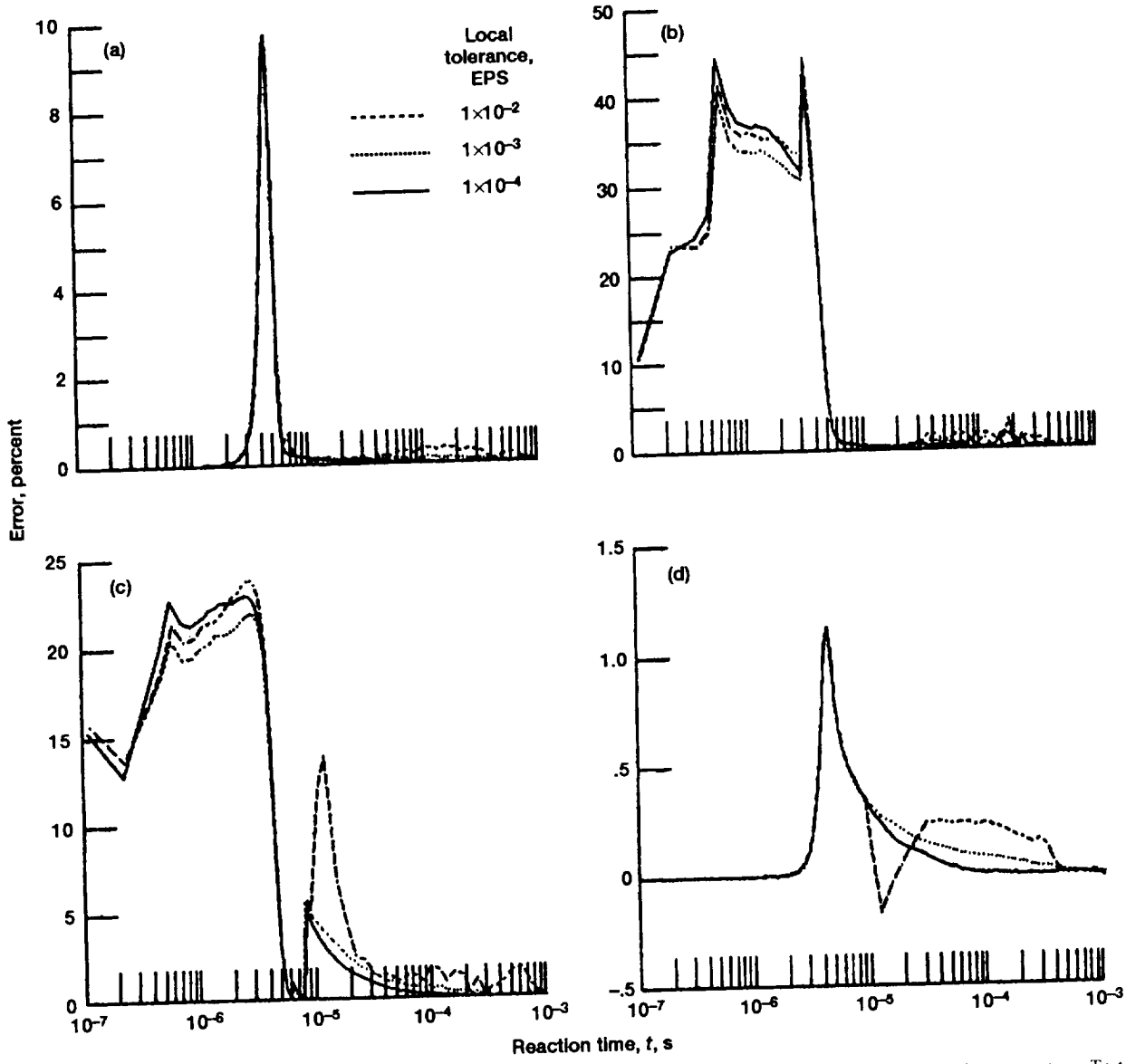

Figure 12.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using LSODE-A.

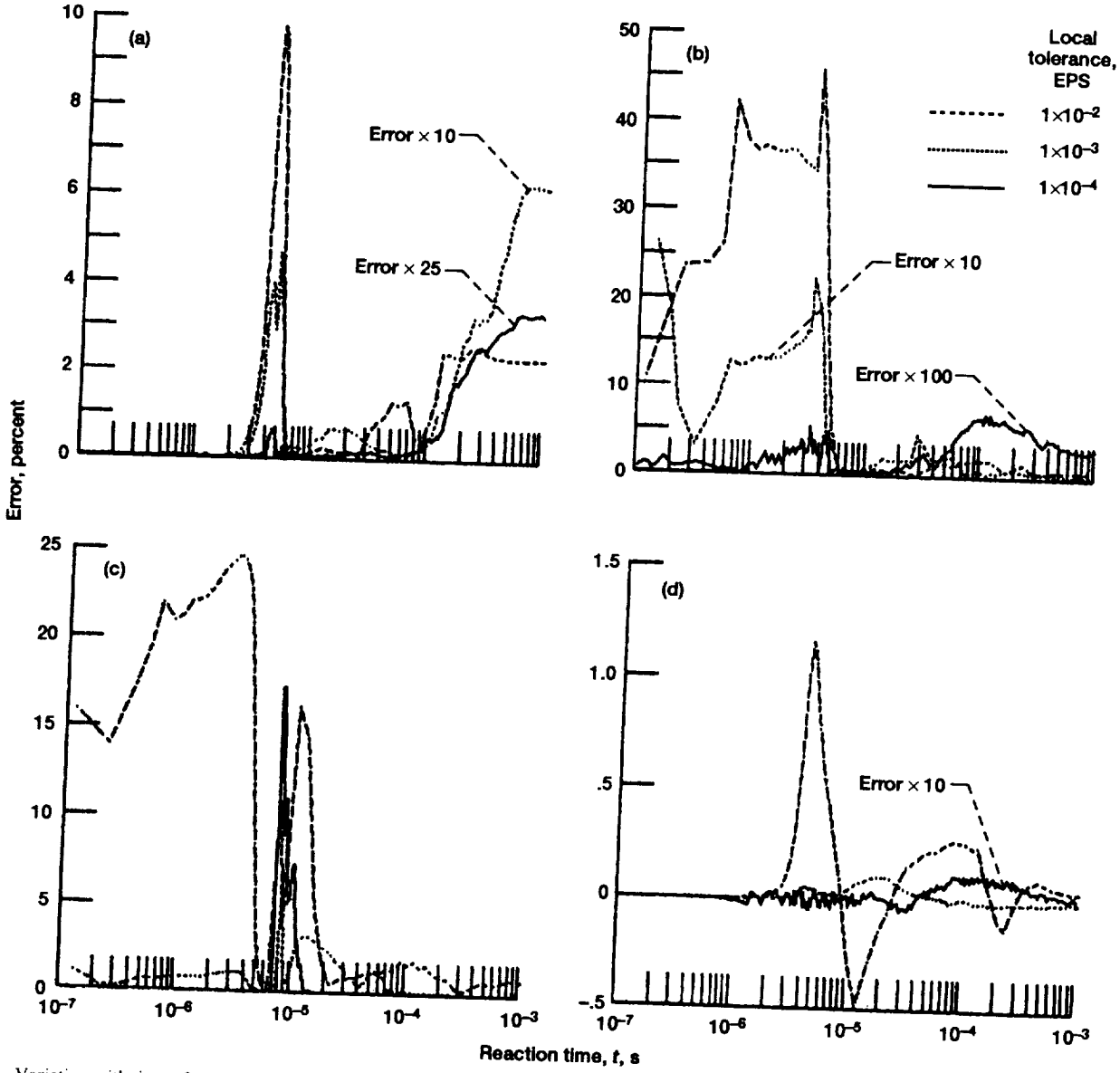

Figure 13.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using LSODE-B.

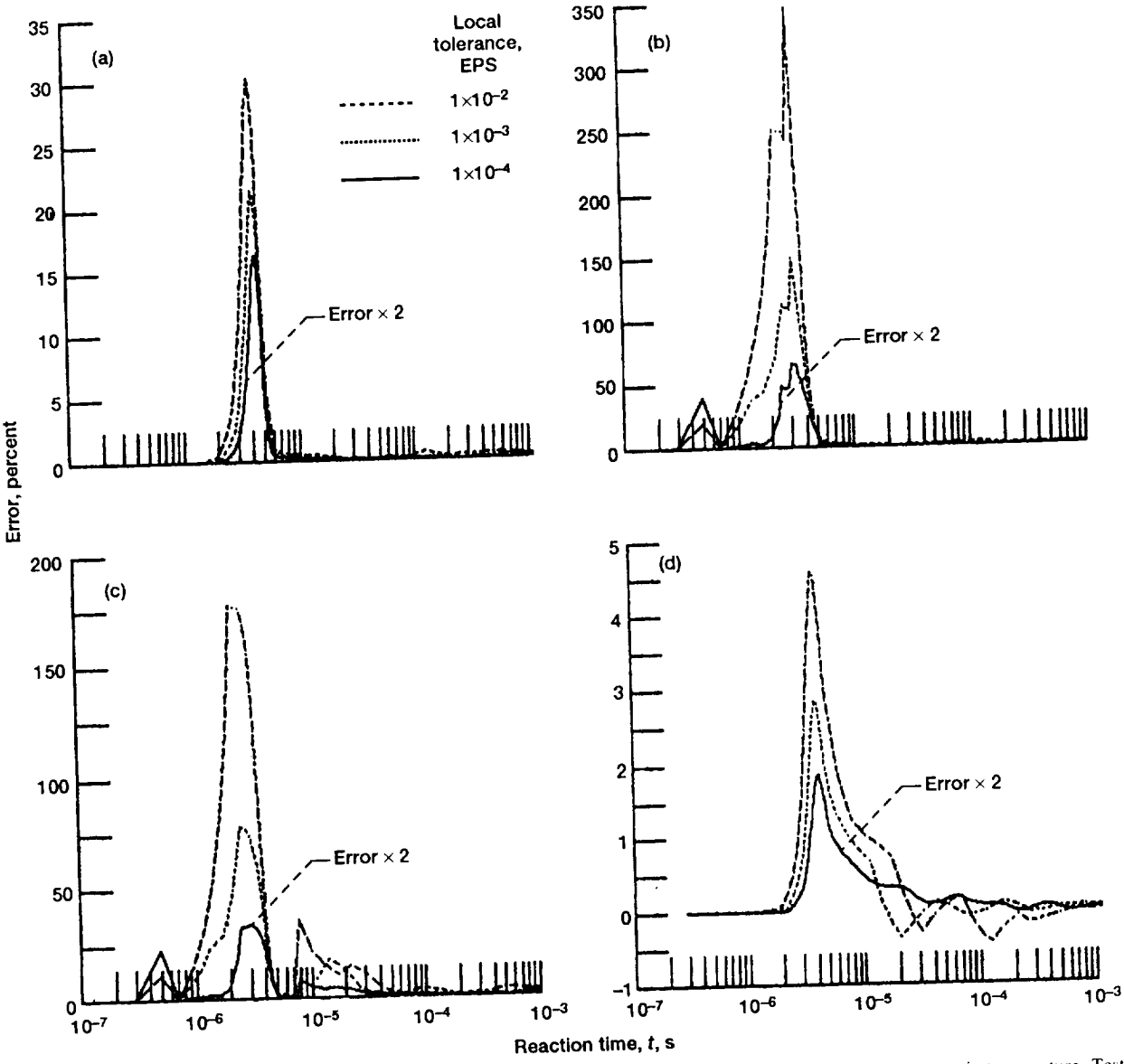

Figure 14.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using GCKP84.

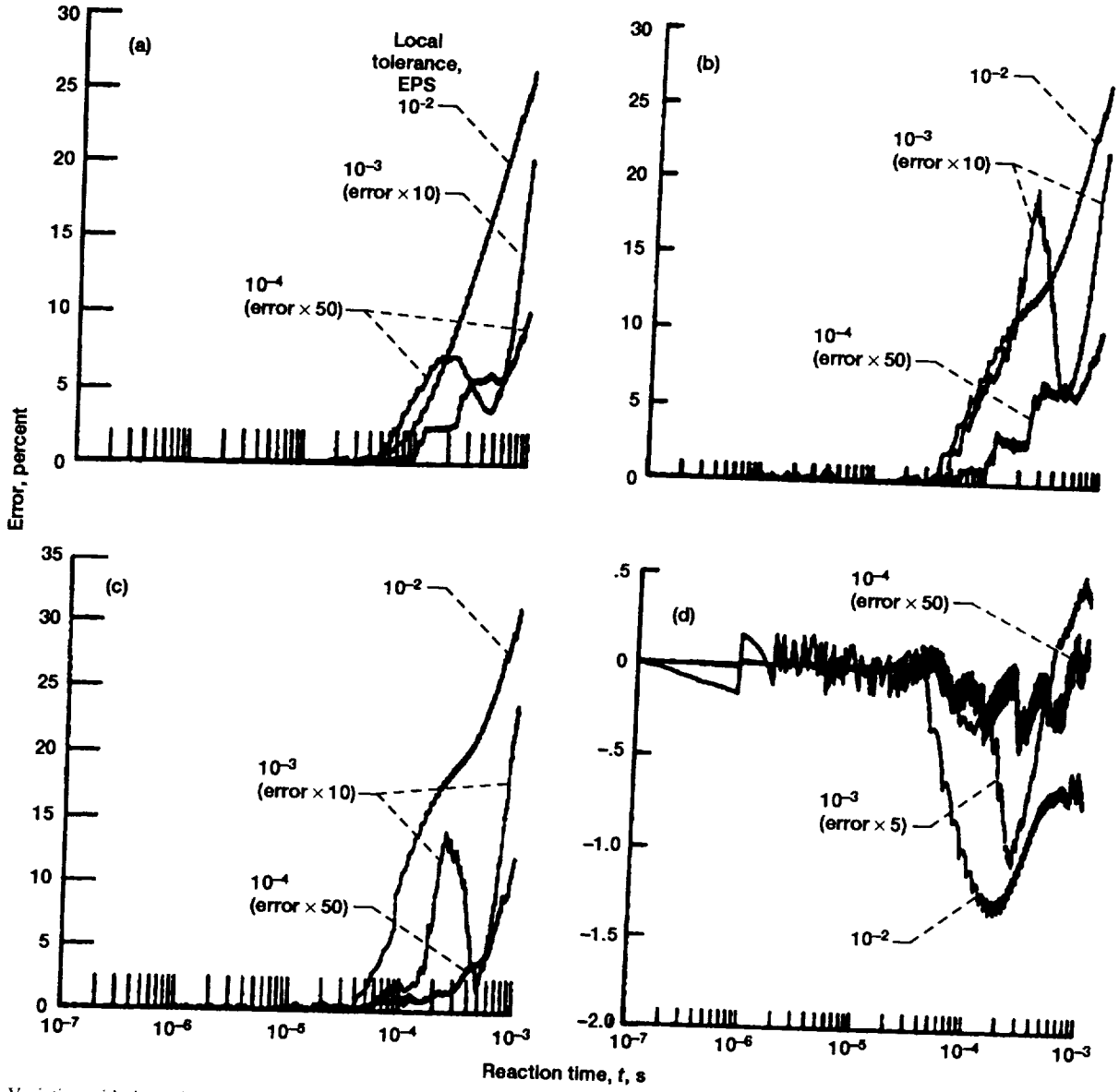

Figure 15.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using CHEMEQ-A.

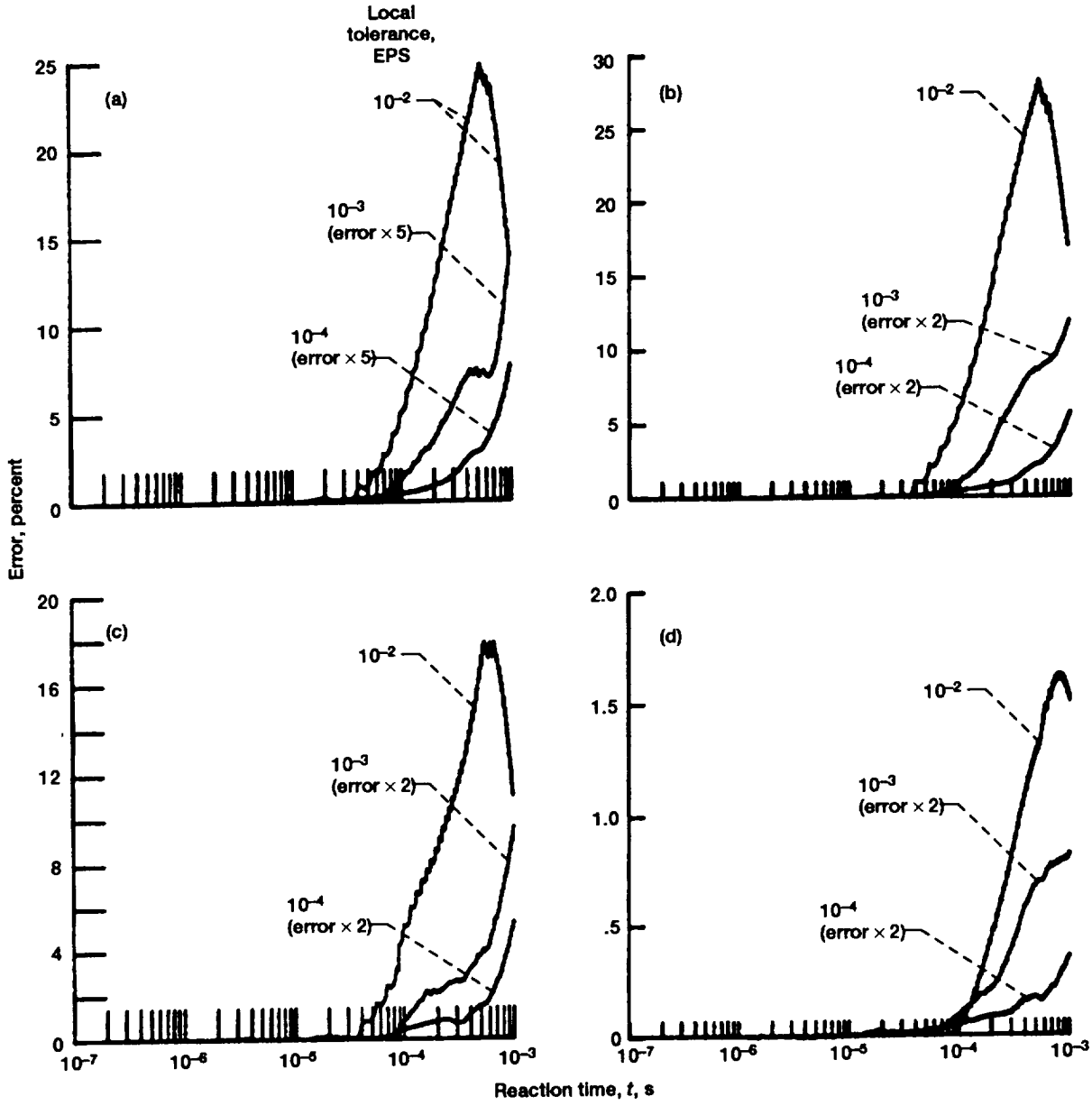

Figure 16.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using CHEMEQ-B.

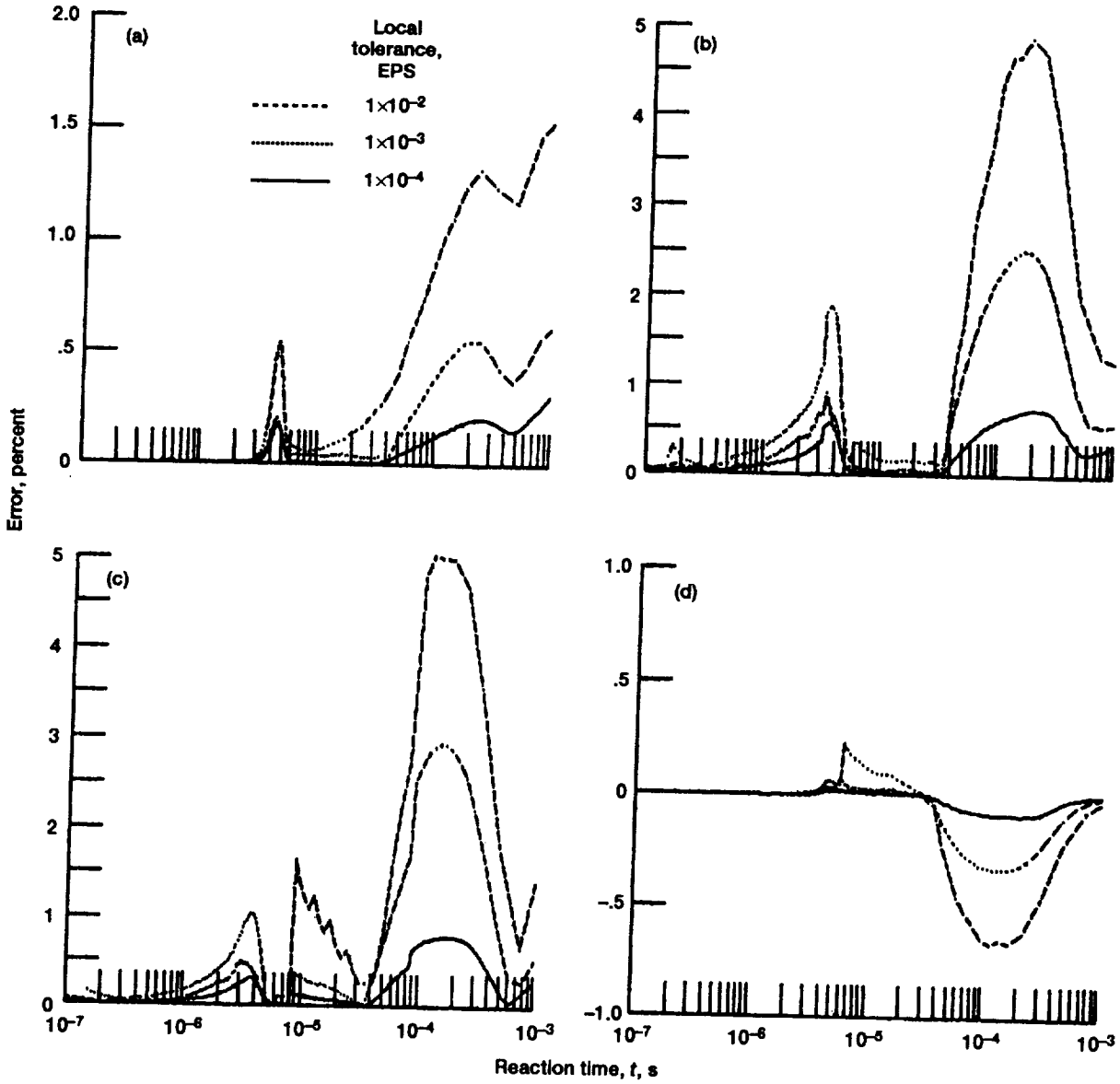

Figure 17.—Variation with time of percent rms error in (a) reactants, (b) intermediates, and (c) products, and of (d) percent error in temperature. Test problem 2 results generated using CREK1D.

TABLE IV.—ROOT-MEAN-SQUARE ERRORS FOR TEST PROBLEM 1

[Numbers in parentheses designate powers of 10.]

Method	Local tolerance, EPS	Local absolute error tolerance for species mole numbers, ATOLSP	Initial step length, h_0, s	Maximum temperature change between updates of rate coefficients and thermodynamic properties, ΔT, K	RMS errors in species concentration						RMS error in species and temperature	
					Reactants		Intermediates		Products			
					Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s
EPISODE-A	5×10^{-6}	(a)	10^{-7}	(a)	1.424(3)	1.483(−3)	1.000(2)	(b)	1.000(2)	(b)	8.257(2)	1.483(−3)
EPISODE-B	5×10^{-6}	(a)	10^{-7}	↓	1.424(3)	1.604(−3)	1.000(2)	(b)	1.000(2)	(b)	8.259(2)	1.604(−3)
LSODE-A	1×10^{-2}	1×10^{-12}	(a)		8.490	1.246(−5)	1.539(1)	1.205(−5)	1.269(1)	9.166(−6)	1.156(1)	1.221(−5)
	1×10^{-3}	1×10^{-11}	↓		7.209	1.268(−5)	1.148(1)	1.206(−5)	9.513	8.991(−6)	8.596	1.222(−5)
	1×10^{-4}	1×10^{-11}			4.331	1.262(−5)	6.328	1.206(−5)	5.632	6.355(−6)	4.786	1.234(−5)
LSODE-B	1×10^{-2}	1×10^{-12}			8.648	1.256(−5)	1.423(1)	1.207(−5)	1.190(1)	1.207(−5)	1.057(1)	1.207(−5)
	1×10^{-3}	1×10^{-11}			7.064	1.266(−5)	1.077(1)	1.220(−5)	9.233	8.275(−6)	8.167	1.220(−5)
	1×10^{-4}	1×10^{-11}			5.742	1.262(−5)	8.410	1.204(−5)	7.496	7.158(−6)	6.366	1.229(−5)
GCKP84	[c]1×10^{-2}	(a)	10^{-6}		4.811(1)	9.699(−6)	6.616(2)	8.900(−6)	5.386(2)	8.900(−6)	4.679(2)	8.900(−6)
	5×10^{-3}	↓	↓		4.842(1)	9.823(−6)	6.616(2)	8.900(−6)	5.386(2)	8.900(−6)	4.679(2)	8.900(−6)
	1×10^{-3}				4.588(1)	1.008(−5)	3.722(2)	9.407(−6)	2.845(2)	9.407(−6)	2.586(2)	9.407(−6)
	1×10^{-4}				3.834(1)	1.031(−5)	1.645(2)	9.790(−6)	1.405(2)	6.240(−6)	1.150(2)	9.790(−6)
CHEMEQ-A	1×10^{-2}		(a)		4.903(1)	9.892(−4)	2.815(1)	9.897(−4)	2.231(1)	1.000(−3)	3.447(1)	9.895(−4)
	1×10^{-3}		↓		5.139	1.718(−4)	9.831	1.719(−4)	5.203	9.755(−4)	6.580	1.719(−4)
	1×10^{-4}				6.951(−1)	4.279(−4)	1.313	4.278(−4)	4.223(−1)	9.856(−4)	8.685(−1)	4.278(−4)
CHEMEQ-B	1×10^{-2}				6.691(1)	9.815(−4)	7.273(1)	9.822(−4)	1.922(1)	4.033(−4)	5.761(1)	9.820(−4)
	1×10^{-3}				1.066(1)	9.829(−4)	2.580(1)	9.834(−4)	2.188(1)	1.000(−3)	1.948(1)	9.834(−4)
	1×10^{-4}				1.411	9.848(−4)	3.232	9.646(−4)	1.795	9.995(−4)	2.219	9.840(−4)
CREK1D	1×10^{-2}			2.0	2.757(1)	2.390(−4)	4.460(1)	2.390(−4)	2.580(1)	1.207(−5)	3.079(1)	2.390(−4)
	1×10^{-3}			2.0	6.307	1.259(−5)	9.555	1.216(−5)	7.893	1.216(−5)	7.218	1.228(−5)
	1×10^{-4}	↓	↓	1.5	2.110	1.271(−5)	3.032	1.216(−5)	2.515	1.216(−5)	2.311	1.236(−5)

[a]Not needed.

[b]See text.

[c]Terminated at $t \cong 3.7\times10^{-5}$ s because of instability.

TABLE V.—ERRORS FOR TEST PROBLEM 1
[Numbers in parentheses designate powers of 10.]

Method	Local tolerance, EPS	Errors in species concentration												Error in temperature		
		Reactants				Intermediates				Products						
		Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Reaction time, s	Standard solution, K
EPISODE-A	5×10^{-6}	2.510(3)	H_2	1.483(−3)	6.377(−3)	−1.000(2)	(a)	(a)	(a)	−1.000(2)	(a)	(a)	(a)	−6.194(1)	1.483(−3)	2627.7
EPISODE-B	5×10^{-6}	2.507(3)	↓	1.604(−3)	6.383(−3)	−1.000(2)	(a)	(a)	(a)	−1.000(2)	(a)	(a)	(a)	−6.194(1)	1.604(−3)	2627.4
LSODE-A	1×10^{-2}	−1.541(1)	↓	1.271(−5)	4.300(−2)	2.254(1)	OH	1.221(−5)	2.515(−3)	1.719(1)	CO_2	1.221(−5)	3.810(−3)	3.121	1.246(−5)	1288.3
	1×10^{-3}	−1.320(1)	↓	1.268(−5)	4.543(−2)	1.669(1)	↓	1.222(−5)	2.558(−3)	1.280(1)	↓	1.233(−5)	4.482(−3)	2.408	1.245(−5)	1284.2
	1×10^{-4}	−7.920	↓	1.262(−5)	5.012(−2)	9.115	↓	1.206(−5)	1.956(−3)	7.104	↓	1.243(−5)	5.089(−3)	1.404	1.252(−5)	1309.9
LSODE-B	1×10^{-2}	−1.571(1)	↓	1.256(−5)	5.473(−2)	2.070(1)	↓	1.207(−5)	1.977(−3)	1.625(1)	↓	1.231(−5)	4.362(−3)	2.951	1.250(−5)	1301.6
	1×10^{-3}	−1.293(1)	↓	1.266(−5)	4.645(−2)	1.573(1)	↓	1.220(−5)	2.449(−3)	1.224(1)	↓	1.233(−5)	4.478(−3)	2.341	1.247(−5)	1292.5
	1×10^{-4}	−1.049(1)	↓	1.262(−5)	5.018(−2)	1.219(1)	↓	1.217(−5)	2.331(−3)	9.425	↓	1.242(−5)	5.004(−3)	1.845	1.252(−5)	1307.9
GCKP84	[b]1×10^{-2}	−7.918(1)	↓	9.699(−6)	1.242(−1)	1.056(3)	↓	8.900(−6)	3.650(−4)	7.434(2)	↓	8.900(−6)	7.291(−4)	4.257(1)	9.699(−6)	1086.5
	5×10^{-3}	−7.938(1)	↓	9.823(−6)	1.184(−1)	1.056(3)	↓	8.900(−6)	3.650(−4)	7.434(2)	↓	8.900(−6)	7.291(−4)	4.311(1)	9.823(−6)	1104.2
	1×10^{-3}	−7.681(1)	↓	1.008(−5)	1.032(−1)	6.002(2)	↓	9.507(−6)	7.431(−4)	4.038(2)	↓	9.707(−6)	1.815(−3)	3.464(1)	1.008(−5)	1151.8
	1×10^{-4}	−6.622(1)	↓	1.031(−5)	8.555(−2)	2.553(2)	↓	9.890(−6)	1.264(−3)	1.755(2)	↓	1.011(−5)	2.921(−3)	2.187(1)	1.031(−5)	1209.6
CHEMEQ-A	1×10^{-2}	−6.457(1)	O_2	1.000(−3)	9.143(−3)	−4.296(1)	O	1.000(−3)	7.951(−4)	−3.751(1)	NO	1.000(−3)	1.662(−3)	−1.873	9.919(−4)	2627.8
	1×10^{-3}	8.506	↓	4.195(−4)	1.001(−2)	1.355(1)	↓	1.719(−4)	1.470(−3)	8.917	↓	9.743(−4)	1.631(−3)	−3.554(−1)	7.417(−5)	2472.1
	1×10^{-4}	1.070	↓	5.053(−4)	9.723(−3)	1.789	↓	4.278(−4)	9.164(−4)	7.183(−1)	↓	9.986(−4)	1.660(−3)	−4.970(−2)	3.731(−4)	2614.1
CHEMEQ-B	1×10^{-2}	9.746(1)	H_2	9.815(−4)	6.300(−3)	1.054(2)	N	9.828(−4)	3.311(−7)	3.107(1)	↓	3.614(−4)	7.642(−4)	2.868	1.000(−3)	2625.7
	1×10^{-3}	1.495(1)	O_2	5.653(−4)	9.562(−3)	3.141(1)	↓	9.835(−4)	3.310(−7)	3.776(1)	↓	1.000(−3)	1.649(−3)	1.519	1.000(−3)	2625.7
	1×10^{-4}	2.123	↓	9.852(−4)	9.105(−3)	3.842	↓	9.646(−4)	3.322(−7)	3.070	↓	9.995(−4)	1.648(−3)	1.863(−1)	1.000(−3)	2625.7
CREK1D	1×10^{-2}	−4.446(1)	↓	2.390(−4)	1.137(−2)	−6.006(1)	O	2.390(−4)	1.203(−3)	3.531(1)	CO_2	1.234(−5)	4.571(−3)	6.204	1.248(−5)	1295.9
	1×10^{-3}	−1.152(1)	H_2	1.259(−5)	5.140(−2)	1.391(1)	OH	1.216(−5)	2.315(−3)	1.073(1)	↓	1.240(−5)	4.905(−3)	2.091	1.252(−5)	1312.5
	1×10^{-4}	−3.890	H_2	1.271(−5)	4.244(−2)	4.381	OH	1.216(−5)	2.334(−3)	3.393	↓	1.236(−5)	4.651(−3)	6.652(−1)	1.244(−5)	1284.3

[a]See text.

[b]Terminated at $t \cong 3.7\times10^{-5}$ s because of instability.

TABLE VI.—ROOT-MEAN-SQUARE ERRORS FOR TEST PROBLEM 2

[Numbers in parentheses designate powers of 10.]

Method	Local tolerance, EPS	Local absolute error tolerance for species mole numbers, ATOLSP	Initial step length, h_0, s	Maximum temperature change between updates of rate coefficients and thermodynamic properties, ΔT, K	RMS errors in species concentration						RMS error in species and temperature	
					Reactants		Intermediates		Products			
					Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s
EPISODE-A	5×10^{-4}	(a)	1×10^{-7}	(a)	4.993(2)	3.660(−3)	1.000(2)	(b)	1.000(2)	(b)	2.904(2)	3.660(−3)
	1×10^{-4}		1×10^{-7}		2.398(1)	4.449(−6)	1.843(2)	3.351(−6)	1.005(2)	2.701(−6)	1.218(2)	3.351(−6)
	1×10^{-5}		1×10^{-6}		1.528(1)	4.550(−6)	8.161(1)	3.288(−6)	4.331(1)	2.711(−6)	5.408(1)	3.288(−6)
EPISODE-B	5×10^{-4}		1×10^{-10}		2.691(2)	4.206(−5)	4.127(2)	4.385(−5)	8.660(1)	3.644(−5)	2.801(2)	4.385(−5)
	1×10^{-4}		1×10^{-11}		2.467(1)	4.400(−6)	1.994(2)	3.359(−6)	1.073(2)	2.681(−6)	1.316(2)	3.359(−6)
	1×10^{-5}		1×10^{-9}		1.893(1)	4.537(−6)	1.185(2)	3.295(−6)	6.284(1)	2.521(−6)	7.831(1)	3.295(−6)
LSODE-A	1×10^{-2}	1×10^{-8}	(a)		9.517	4.741(−6)	4.414(1)	3.441(−6)	2.375(1)	2.969(−6)	2.938(1)	3.441(−6)
	1×10^{-3}				9.408	4.677(−6)	4.114(1)	3.388(−6)	2.185(1)	2.976(−6)	2.737(1)	3.388(−6)
	1×10^{-4}				9.722	4.703(−6)	4.438(1)	5.983(−7)	2.289(1)	2.751(−6)	2.944(1)	5.983(−7)
LSODE-B	1×10^{-2}	1×10^{-9}			9.797	4.616(−6)	4.589(1)	3.420(−6)	2.461(1)	2.991(−6)	3.053(1)	3.420(−6)
	1×10^{-3}	1×10^{-8}			6.272(−1)	6.762(−4)	2.620	1.382(−7)	6.968	8.337(−6)	3.484	8.337(−6)
	1×10^{-4}	1×10^{-12}			9.922	4.656(−6)	4.541(1)	6.044(−7)	3.056(3)	1.280(−5)	1.528(3)	1.280(−5)
	1×10^{-4}				1.355(−1)	6.456(−4)	1.536(−1)	4.061(−8)	1.728(1)	7.700(−6)	8.641	7.700(−6)
GCKP84	1×10^{-2}	(a)	1×10^{-6}		3.024(1)	3.999(−6)	3.461(2)	2.950(−6)	1.776(2)	2.190(−6)	2.272(2)	2.950(−6)
	1×10^{-3}		1×10^{-6}		2.138(1)	4.117(−6)	1.493(2)	3.015(−6)	7.826(1)	2.526(−6)	9.865(1)	3.015(−6)
	1×10^{-4}		1×10^{-6}		8.134	4.321(−6)	3.272(1)	2.937(−6)	1.703(1)	3.072(−6)	2.176(1)	2.937(−6)
CHEMEQ-A	1×10^{-2}		(a)		2.622(1)	1.000(−3)	2.683(1)	1.000(−3)	3.117(1)	1.000(−3)	2.697(1)	1.000(−3)
	1×10^{-3}				2.025	9.957(−4)	2.230	9.780(−4)	2.370	9.954(−4)	2.133	9.956(−4)
	1×10^{-4}				2.022(−1)	1.000(−3)	2.059(−1)	9.960(−4)	2.420(−1)	9.973(−4)	2.074(−1)	9.973(−4)
CHEMEQ-B	1×10^{-2}				2.466(1)	5.508(−4)	2.820(1)	5.374(−4)	1.781(1)	6.790(−4)	2.380(1)	5.391(−4)
	1×10^{-3}				2.789	1.000(−3)	5.937	1.000(−3)	4.814	9.996(−4)	4.631	1.000(−3)
	1×10^{-4}				1.531	1.000(−3)	2.826	9.999(−4)	2.648	1.000(−3)	2.341	1.000(−3)
CREK1D	1×10^{-2}			1.5	1.539	1.000(−3)	4.880	1.717(−4)	5.036	1.157(−4)	3.959	1.717(−4)
	1×10^{-3}			2.0	6.131(−1)	1.000(−3)	2.514	1.620(−4)	2.942	1.453(−4)	2.148	1.620(−4)
	1×10^{-4}			2.0	3.080(−1)	9.847(−4)	7.408(−1)	2.061(−4)	7.964(−1)	1.698(−4)	6.122(−1)	2.061(−4)

[a]Not needed.

[b]See text.

TABLE VII.—ERRORS FOR TEST PROBLEM 2
[Numbers in parentheses designate powers of 10.]

Method	Local tolerance, EPS	Errors in species concentration												Error in temperature		
		Reactants				Intermediates				Products						
		Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Reaction time, s	Standard solution, K
EPISODE-A	5×10^{-4}	9.896(2)	O_2	3.660(−3)	1.408(−2)	−1.000(2)	(a)	(a)	(a)	−1.000(2)	(a)	(a)	(a)	−4.958(1)	3.660(−3)	2908.5
	1×10^{-4}	−4.120(1)	H_2	4.449(−6)	1.619(−1)	3.196(2)	H_2O_2	3.351(−6)	1.534(−7)	2.009(2)	H_2O	2.701(−6)	2.179(−3)	3.284	4.349(−6)	1569.9
	1×10^{-5}	−2.648(1)	H_2	4.550(−6)	1.446(−1)	1.392(2)	H_2O_2	3.288(−6)	1.233(−7)	8.662(1)	H_2O	2.711(−6)	2.235(−3)	1.873	4.371(−6)	1572.4
EPISODE-B	5×10^{-4}	4.679(2)	O_2	4.206(−5)	2.603(−2)	9.981(2)	HO_2	4.385(−5)	4.782(−6)	−1.000(2)	(a)	(a)	(a)	−3.890(1)	4.260(−5)	2457.7
	1×10^{-4}	−4.234(1)	H_2	4.517(−6)	1.517(−1)	3.396(2)	H_2O_2	3.359(−6)	1.534(−7)	2.147(2)	H_2O	2.681(−6)	2.035(−3)	3.434	4.284(−6)	1562.1
	1×10^{-5}	−3.273(1)	↓	4.537(−6)	1.482(−1)	2.057(2)	↓	3.295(−6)	1.228(−7)	1.257(2)	↓	2.521(−6)	1.373(−3)	2.430	4.361(−6)	1570.3
LSODE-A	1×10^{-2}	−1.663(1)		4.741(−6)	1.255(−1)	7.289(1)		3.441(−6)	1.637(−7)	4.751(1)		2.969(−6)	3.530(−3)	1.117	4.543(−6)	1583.7
	1×10^{-3}	−1.638(1)		4.677(−6)	1.356(−1)	7.100(1)	HO_2	6.059(−7)	1.317(−7)	4.369(1)		2.976(−6)	3.587(−3)	1.090	4.563(−6)	1585.9
	1×10^{-4}	−1.696(1)		4.703(−6)	1.314(−1)	7.929(1)	HO_2	5.983(−7)	1.274(−7)	4.578(1)		2.751(−6)	2.071(−3)	1.125	4.549(−6)	1584.4
LSODE-B	1×10^{-2}	−1.700(1)		4.616(−6)	1.457(−1)	7.613(1)	H_2O_2	3.420(−6)	1.524(−7)	4.921(1)		2.991(−6)	3.726(−3)	1.158	4.432(−6)	1571.1
	1×10^{-3}	1.401	CO_2	6.762(−4)	2.352(−4)	−5.935	OH	1.382(−7)	3.131(−7)	−1.394(1)	NO	8.337(−6)	1.363(−7)	1.054(−1)	1.965(−5)	2170.9
	[b]1×10^{-4}	−1.726(1)	H_2	4.656(−6)	1.389(−1)	8.084(1)	HO_2	6.044(−7)	1.308(−7)	6.112(3)		1.280(−5)	9.425(−7)	1.147	4.521(−6)	1581.1
	[c]1×10^{-4}	−3.023(−1)	CO_2	6.456(−4)	2.353(−4)	−3.183(−1)	H	4.061(−8)	1.243(−7)	−3.456(1)	↓	7.700(−6)	1.174(−7)	1.164(−2)	8.537(−5)	2679.5
GCKP84	1×10^{-2}	−5.147(1)	H_2	3.999(−6)	1.827(−1)	6.238(2)	H_2O_2	2.950(−6)	1.208(−7)	3.551(2)	H_2O	2.190(−6)	1.387(−3)	4.575	3.872(−6)	1554.8
	1×10^{-3}	−3.672(1)		4.117(−6)	1.624(−1)	2.589(2)		3.015(−6)	1.516(−7)	1.565(2)		2.526(−6)	3.190(−3)	2.824	3.946(−6)	1562.2
	1×10^{-4}	−1.418(1)	↓	4.321(−6)	1.284(−1)	5.441(1)	↓	2.937(−6)	1.156(−7)	3.407(1)	↓	3.072(−6)	1.189(−2)	9.296(−1)	4.151(−6)	1584.8
CHEMEQ-A	1×10^{-2}	−4.910(1)	O_2	1.000(−3)	1.412(−2)	−4.398(1)	HO_2	9.826(−4)	1.395(−5)	−4.719(1)	NO_2	1.000(−3)	1.967(−6)	−1.346	1.588(−4)	2829.8
	1×10^{-3}	4.226		9.957(−4)	1.413(−2)	4.255		9.780(−4)	1.395(−5)	3.677		9.954(−4)	1.966(−6)	−2.147(−1)	2.432(−4)	2888.6
	1×10^{-4}	3.888(−1)	↓	9.979(−4)	1.412(−2)	3.882(−1)	↓	9.973(−4)	1.395(−5)	3.882(−1)	↓	9.973(−4)	1.967(−6)	−9.072(−3)	2.577(−4)	2893.4
CHEMEQ-B	1×10^{-2}	3.936(1)	H_2	5.368(−4)	4.913(−2)	4.431(1)	N	5.375(−4)	3.574(−6)	−3.016(1)		6.790(−4)	1.772(−6)	1.625	7.911(−4)	2905.5
	1×10^{-3}	5.854	O_2	1.000(−3)	1.373(−2)	8.470		9.876(−4)	3.896(−6)	7.041		9.996(−4)	1.894(−6)	4.152(−1)	9.966(−4)	2903.2
	1×10^{-4}	3.049	↓	1.000(−3)	1.373(−2)	4.015	↓	1.000(−3)	3.897(−6)	3.937	↓	1.000(−3)	1.894(−6)	1.788(−1)	1.000(−3)	2903.2
CREK1D	1×10^{-2}	3.039		1.000(−3)	1.420(−2)	6.714	H_2O_2	1.717(−4)	8.967(−7)	−6.875	N_2O	1.717(−4)	4.788(−7)	−6.604(−1)	1.157(−4)	2760.4
	1×10^{-3}	1.258		1.000(−3)	1.420(−2)	3.302		1.825(−4)	8.624(−7)	−3.871		1.825(−4)	4.988(−7)	−3.304(−1)	1.453(−4)	2811.8
	1×10^{-4}	6.579(−1)	↓	9.847(−4)	1.420(−2)	1.028	↓	2.061(−4)	8.032(−7)	−1.165	↓	2.061(−4)	5.365(−7)	−9.044(−2)	2.061(−4)	2870.2

[a]See text.
[b]ATOLSP = 10^{-8}.
[c]ATOLSP = 10^{-12}.

ATOLSP, the runs with LSODE-A and LSODE-B were required to satisfy the accuracy criteria described in reference 2. Several of the runs with the other codes also satisfied these criteria. For LSODE-B and EPS = 10^{-4}, two ATOLSP values (10^{-8} and 10^{-12}) satisfied both the accuracy and execution time criteria for P2. Table VI includes results obtained with both values, to illustrate the effect of ATOLSP on the accuracy. However, the error plot given in figure 13 was generated with ATOLSP = 10^{-12}.

The maximum percent errors in each species type and temperature are given in tables V and VII for P1 and P2. For each species type the species incurring the maximum error, the reaction times at which the maximum errors occurred, and the standard solution values for species mole fractions and temperature at these times are listed.

For test problem 1 the runs with EPISODE-A and EPISODE-B and EPS $\geq 5\times10^{-6}$ predicted little or no change in the composition and temperature after an elapsed time of 1 ms. Hence, maximum errors of -100 percent were obtained for both intermediates and products (table V). Because the temperature and the more active reactants H_2, CO, and O_2 display monotonic behavior (fig. 2) the maximum errors in reactants and temperature occurred at t_{end}, the final time (≥ 1 ms) at which the solution was generated. Both EPISODE-A and EPISODE-B required only six steps to complete the problem, and, because a new step size is considered after every successful step, t_{end} was significantly greater than 1 ms. For intermediates and products maximum rms errors of 100 percent were obtained at several time values; hence, reaction times and standard solution values are not given in tables IV and V for these two species types. Also, no species name is listed in table V for either type because all intermediate and product species incurred the shown maximum errors.

The solution returned by EPISODE was also found to depend on the output stations specified by the user. For example, for some combinations of output times, EPISODE-B (with EPS = 1×10^{-6}) predicted no change in the composition and temperature after an elapsed time of 1 ms. However, by stipulating only one output station (at 1 ms), the correct solution was obtained. For this problem the run with GCKP84 and EPS = 1×10^{-2} exhibited serious instability and was therefore terminated. For reasons just discussed, no error plots for EPISODE-A, EPISODE-B, and the run with GCKP84 and EPS = 10^{-2} are presented.

Similar remarks apply to the results obtained for P2 with EPISODE-A and EPS $\geq 5\times10^{-4}$, and with EPISODE-B and EPS $\geq 5\times10^{-3}$. The run with EPISODE-A and EPS = 5×10^{-4} required only seven steps to complete the problem and t_{end} was therefore significantly greater than 1 ms. For this EPS, for exactly the same reasons given for P1, the following quantities are not shown in tables VI and VII: reaction times at which the maximum rms errors occurred in the intermediates and products, intermediate and product species incurring the maximum errors, the reaction times at which the maximum errors occurred, and the standard solution values.

For EPS = 5×10^{-4} and 10^{-3} EPISODE-B successfully completed P2 in that correct solutions were returned at $t = 1$ ms. However, during heat release they were significantly inaccurate. For example, the run with EPS = 5×10^{-4} predicted little change from the initial composition and temperature until $t \simeq 40$ μs when heat release began. In contrast, the standard solution shows that heat release is almost over by this time (fig. 3). As a result, maximum errors of -100 percent were observed for the products (table VII). This error was incurred by several product species at several time steps; hence, table VII does not list the product species name, reaction time, and standard solution value. Because of the difficulties experienced by EPISODE-A and EPISODE-B, error plots for EPS $\geq 5\times10^{-4}$ are not presented.

As discussed previously, all results with GCKP84 were obtained with $h_0 = 10^{-6}$ s, although its current default value is 5×10^{-8} s. The effects of this change in h_0 on the accuracy and execution time were studied by generating results with $h_0 = 5\times10^{-8}$ s. The maximum errors incurred by the solutions produced with both h_0 values are given in tables VIII to XI. For this study new standard solutions using $h_0 = 5\times10^{-8}$ s were established to bias the results in favor of the current default value for h_0. Despite this bias, tables VIII to XI show that in almost all cases $h_0 = 10^{-6}$ s produced more accurate solutions than $h_0 = 5\times10^{-8}$ s. (No results are given for P1 and EPS = 10^{-2} because the runs with both h_0 were terminated due to instability.) For P1 the results with $h_0 = 10^{-6}$ s were significantly more accurate for all values of EPS (tables VIII and IX). Surprisingly, for $h_0 = 5\times10^{-8}$ s the solution with EPS = 10^{-4} incurred substantially greater errors than those generated with the larger EPS. For P2 the differences in errors obtained with the two h_0 were small for EPS = 10^{-2} and 10^{-3}, but for EPS = 10^{-4}, $h_0 = 10^{-6}$ s produced significantly more accurate results than $h_0 = 5\times10^{-8}$ s (tables X and XI). Finally, the execution times required with the two h_0 are comparable for both problems and all EPS (tables VIII and X).

Examination of figures 4 to 17 shows sudden increases in the error plots for intermediate species and products. This behavior is caused by species reaching values of 0.1 ppm or greater (figs. 2 and 3) and introducing their contributions to the rms errors. For example, for P1 the intermediate species producing the sudden increases in the error plots are H (at $t \simeq 2$ μs), O (at $t \simeq 4$ μs), OH (at $t \simeq 4$ μs), and N (at $t \simeq 20$ μs). For products the pertinent species are H_2O (at $t \simeq 2$ μs), CO_2 (at $t \simeq 4$ μs), and NO (at $t \simeq 15$ μs).

EPISODE-A, EPISODE-B, LSODE-A, LSODE-B, GCKP84, and CREK1D all experienced difficulty tracking the standard solutions during induction and early heat release when the species and temperature change rapidly (figs. 2 and 3). The essentially isothermal induction period ends and heat release begins when the temperature starts to rapidly increase from its initial value. During induction, the reactants and

TABLE VIII.—EFFECTS OF INITIAL STEP LENGTH ON ROOT-MEAN-SQUARE ERRORS, MEAN INTEGRATED RMS ERROR, AND EXECUTION TIME FOR GCKP84 AND TEST PROBLEM 1

Local tolerance, EPS	Initial step length, h_0, s	RMS errors in species concentration						RMS error in species and temperature		Mean integrated rms error, ε_{rms}	CPU execution time, s
		Reactants		Intermediates		Products					
		Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s		
5×10^{-3}	5×10^{-8}	52.89	9.409×10^{-6}	1.976×10^{3}	3.942×10^{-6}	1.995×10^{3}	4.918×10^{-6}	1.515×10^{3}	4.918×10^{-6}	7.622×10^{-2}	0.87
	1×10^{-6}	44.16	9.612×10^{-6}	356.2	8.900×10^{-6}	282.0	8.900×10^{-6}	250.2	8.900×10^{-6}	1.938×10^{-2}	.86
1×10^{-3}	5×10^{-8}	44.87	9.929×10^{-6}	325.1	9.387×10^{-6}	297.7	3.554×10^{-6}	235.2	3.554×10^{-6}	2.375×10^{-2}	.97
	1×10^{-6}	38.61	9.894×10^{-6}	169.4	9.307×10^{-6}	132.4	9.307×10^{-6}	119.3	9.307×10^{-6}	1.381×10^{-2}	.98
1×10^{-4}	5×10^{-8}	51.71	9.361×10^{-6}	3.472×10^{3}	2.307×10^{-6}	3.323×10^{3}	2.674×10^{-6}	2.455×10^{3}	2.307×10^{-6}	1.084×10^{-1}	1.0
	1×10^{-6}	21.26	1.011×10^{-5}	50.73	2.900×10^{-6}	47.86	2.900×10^{-6}	37.82	2.900×10^{-6}	4.729×10^{-3}	1.1

TABLE IX.—EFFECTS OF INITIAL STEP LENGTH ON ERRORS INCURRED BY GCKP84 FOR TEST PROBLEM 1

Local tolerance, EPS	Initial step length, h_0, s	Errors in species concentration												Error in temperature		
		Reactants				Intermediates				Products						
		Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Reaction time, s	Standard solution, K
5×10^{-3}	5×10^{-8}	−84.77	H_2	8.858×10^{-6}	1.467×10^{-1}	3.098×10^{3}	OH	8.006×10^{-6}	1.521×10^{-4}	2.452×10^{3}	CO_2	4.918×10^{-6}	5.397×10^{-6}	60.89	9.409×10^{-6}	1065.6
	1×10^{-6}	−74.02		9.612×10^{-6}	1.027×10^{-1}	567.4		8.900×10^{-6}	6.320×10^{-4}	392.9		8.900×10^{-6}	1.248×10^{-3}	32.58	9.612×10^{-6}	1153.2
1×10^{-3}	5×10^{-8}	−75.63		9.929×10^{-6}	1.060×10^{-1}	520.9		9.465×10^{-6}	8.112×10^{-4}	364.6		3.554×10^{-6}	5.822×10^{-7}	32.35	9.929×10^{-6}	1142.9
	1×10^{-6}	−66.61		9.894×10^{-6}	8.070×10^{-2}	264.1		9.407×10^{-6}	1.250×10^{-3}	186.8		9.507×10^{-6}	2.517×10^{-3}	22.28	9.707×10^{-6}	1175.2
1×10^{-4}	5×10^{-8}	−83.68		9.106×10^{-6}	1.409×10^{-1}	4.918×10^{3}	O	2.307×10^{-6}	1.298×10^{-7}	4.912×10^{3}	H_2O	2.307×10^{-6}	1.146×10^{-6}	55.25	9.520×10^{-6}	1077.9
	1×10^{-6}	−38.08		1.011×10^{-5}	6.182×10^{-2}	68.07	OH	9.690×10^{-6}	1.963×10^{-3}	−58.62	CO_2	2.900×10^{-6}	4.348×10^{-7}	8.525	1.000×10^{-5}	1259.2

TABLE X.—EFFECTS OF INITIAL STEP LENGTH ON ROOT-MEAN-SQUARE ERRORS, MEAN INTEGRATED RMS ERROR, AND EXECUTION TIME FOR GCKP84 AND TEST PROBLEM 2

Local tolerance, EPS	Initial step length, h_0, s	RMS errors in species concentration						RMS error in species and temperature		Mean integrated rms error, $\mathcal{E}_{rms}$	CPU execution time, s
		Reactants		Intermediates		Products					
		Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s		
1×10^{-2}	5×10^{-8}	32.54	4.070×10^{-6}	431.1	2.995×10^{-6}	225.1	2.595×10^{-6}	282.9	2.995×10^{-6}	1.935×10^{-2}	1.7
	1×10^{-6}	31.44	3.999×10^{-6}	405.3	2.950×10^{-6}	203.9	2.407×10^{-6}	265.5	2.950×10^{-6}	2.115×10^{-2}	1.8
1×10^{-3}	5×10^{-8}	22.30	4.220×10^{-6}	159.3	3.095×10^{-6}	86.97	2.470×10^{-6}	105.5	3.095×10^{-6}	6.281×10^{-3}	2.0
	1×10^{-6}	23.14	4.117×10^{-6}	180.3	3.015×10^{-6}	93.41	2.526×10^{-6}	118.9	3.015×10^{-6}	7.891×10^{-3}	1.9
1×10^{-4}	5×10^{-8}	16.96	4.295×10^{-6}	98.25	2.962×10^{-6}	51.31	2.471×10^{-6}	65.03	2.962×10^{-6}	4.277×10^{-3}	2.3
	1×10^{-6}	10.76	4.321×10^{-6}	47.27	3.072×10^{-6}	24.61	2.937×10^{-6}	31.46	3.072×10^{-6}	2.396×10^{-3}	2.4

TABLE XI.—EFFECTS OF INITIAL STEP LENGTH ON ERRORS INCURRED BY GCKP84 FOR TEST PROBLEM 1

Local tolerance, EPS	Initial step length, h_0, s	Errors in species concentration												Error in temperature		
		Reactants				Intermediates				Products						
		Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Reaction time, s	Standard solution, K
1×10^{-2}	5×10^{-8}	−55.38	H_2	4.070×10^{-6}	1.783×10^{-1}	778.7	H_2O_2	2.995×10^{-6}	1.214×10^{-7}	450.1	H_2O	2.595×10^{-6}	3.382×10^{-3}	5.000	4.070×10^{-6}	1570.7
	1×10^{-6}	−53.41		3.999×10^{-6}	1.903×10^{-1}	742.2		2.950×10^{-6}	1.038×10^{-7}	407.8		2.407×10^{-6}	2.130×10^{-3}	4.861	3.872×10^{-6}	1550.5
1×10^{-3}	5×10^{-8}	−38.38		4.220×10^{-6}	1.524×10^{-1}	271.8		3.095×10^{-6}	1.714×10^{-7}	173.9		2.470×10^{-6}	2.486×10^{-3}	3.008	3.970×10^{-6}	1560.1
	1×10^{-6}	−39.63		4.117×10^{-6}	1.702×10^{-1}	317.4		3.015×10^{-6}	1.304×10^{-7}	186.8		2.526×10^{-6}	2.853×10^{-3}	3.132	4.117×10^{-6}	1575.8
1×10^{-4}	5×10^{-8}	−29.37		4.295×10^{-6}	1.399×10^{-1}	169.0		2.962×10^{-6}	1.082×10^{-7}	102.6		2.471×10^{-6}	2.490×10^{-3}	2.131	4.103×10^{-6}	1574.3
	1×10^{-6}	−18.70		4.321×10^{-6}	1.355×10^{-1}	78.12		3.072×10^{-6}	1.582×10^{-7}	49.23		2.937×10^{-6}	7.767×10^{-3}	1.259	4.151×10^{-6}	1579.6

TABLE XII.—EFFECTS OF ABSOLUTE ERROR TOLERANCE FOR SPECIES MOLE NUMBERS ON ROOT-MEAN-SQUARE ERRORS, MEAN INTEGRATED RMS ERROR, AND EXECUTION TIME FOR TEST PROBLEM 2 WITH LSODE-B AND EPS = 10^{-5}

Local absolute error tolerance for species mole numbers, ATOLSP	RMS errors in species concentration						RMS error in species and temperature		Mean integrated rms error, ε_{rms}	CPU execution time, s
	Reactants		Intermediates		Products					
	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s	Maximum rms error, percent	Reaction time, s		
10^{-7}	17.07	4.575×10^{-6}	134.8	2.956×10^{-5}	1.453×10^{5}	1.090×10^{-5}	7.263×10^{4}	1.090×10^{-5}	5.543	1.7
10^{-8}	9.853	4.694×10^{-6}	45.23	6.031×10^{-7}	2.224×10^{4}	8.241×10^{-6}	1.112×10^{4}	8.241×10^{-6}	2.792×10^{-1}	4.2
10^{-9}	5.601×10^{-1}	4.753×10^{-6}	1.828	3.368×10^{-6}	4.044×10^{3}	8.117×10^{-6}	2.022×10^{3}	8.117×10^{-6}	4.011×10^{-2}	3.2
10^{-10}	7.943×10^{-2}	4.758×10^{-6}	4.541×10^{-1}	6.498×10^{-8}	40.51	6.725×10^{-6}	20.25	6.725×10^{-6}	9.421×10^{-4}	2.5

TABLE XIII.—EFFECTS OF ABSOLUTE ERROR TOLERANCE FOR SPECIES MOLE NUMBERS ON ERRORS INCURRED FOR TEST PROBLEM 2 BY LSODE-B AND EPS = 10^{-5}

Local absolute error tolerance for species mole numbers, ATOLSP	Errors in species concentration												Error in temperature		
	Reactants				Intermediates				Products						
	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Species	Reaction time, s	Standard solution, mole fraction	Maximum error, percent	Reaction time, s	Standard solution, K
1×10^{-7}	−29.57	H_2	4.636×10^{-6}	1.422×10^{-1}	330.1	N	2.956×10^{-5}	1.087×10^{-7}	2.905×10^{5}	NO	1.090×10^{-5}	4.451×10^{-7}	2.141	4.453×10^{-6}	1573.4
1×10^{-8}	−17.17	↓	4.694×10^{-6}	1.328×10^{-1}	80.59	HO_2	6.031×10^{-7}	1.301×10^{-7}	4.448×10^{4}	↓	8.241×10^{-6}	1.109×10^{-7}	1.140	4.487×10^{-6}	1577.3
1×10^{-9}	-9.834×10^{-1}		4.753×10^{-6}	1.237×10^{-1}	−4.080	OH	1.431×10^{-7}	3.170×10^{-7}	8.087×10^{3}		8.117×10^{-6}	1.002×10^{-7}	6.266×10^{-2}	4.560×10^{-6}	1585.6
1×10^{-10}	-1.405×10^{-1}		4.758×10^{-6}	1.230×10^{-1}	-8.137×10^{-1}	OH	6.498×10^{-8}	2.386×10^{-7}	−81.01		6.725×10^{-6}	2.622×10^{-7}	1.049×10^{-2}	4.586×10^{-6}	1588.5

temperature remain fairly constant. EPISODE, LSODE, and CREK1D have virtually no errors in the reactants and temperature until heat release begins at ~ 9 μs (P1) and 3 μs (P2), when the errors in these quantities start to increase. For GCKP84, however, the error increases start at earlier times. The difference, more noticeable for P1, is due to the smaller reaction times obtained with this code for the onset of heat release. Although, for consistency, we used EPS = 10^{-5} to generate standard solutions with GCKP84, values of EPS as small as 10^{-9} (for P1) and 10^{-8} (for P2) were found to be necessary to achieve tolerance independence of the temperature-time trace at early times.

During induction, the intermediate species and the product H_2O increase rapidly from negligible initial concentrations. The errors in the intermediates and products in this regime are, therefore, relatively large. During early heat release ($t \lesssim 15$ μs for P1 and $t \lesssim 6$ μs for P2), these errors continue to remain large as more products are formed and the intermediate species continue to change quickly. In this regime, the reactants show a sharp decrease, and the temperature rises significantly. Many of the ODE's are unstable (ref. 2), and so errors introduced at any step will grow as the integration proceeds (refs. 1 and 24). For P1 the reactants and temperature vary rapidly between 9 and ~ 15 μs. For P2 the temperature rise is not as steep, but the reactants change sharply between 3 and ~ 6 μs. Between these times, the errors in the reactants and temperature are relatively large (figs. 4 to 7, 9 to 14, and 17). For P2 the errors incurred at early times are less for CREK1D than for LSODE because of the much smaller step lengths used by the former code in these regimes (refs. 2, 3, and 12). During late heat release and equilibration, however, EPISODE, LSODE, and GCKP84 incur much smaller errors. In these regimes the ODE's are stable (ref. 2), and so the errors decay as the integration proceeds, provided, of course, that the numerical method is stable.

The error plots for EPISODE, LSODE, and GCKP84 illustrate the dangers of assessing the accuracy of a technique (or of a run with a certain value for EPS) by comparing solutions at the final time ($\cong 1$ ms for both problems). Note that, although all these codes have negligible errors at the final times, the errors can be significant at early times. For example, with GCKP84 and EPS = 10^{-3} the maximum rms error in products is over 500 percent for P1. These plots also indicate that if the main objective of the calculations is to study postheat release phenomena (e.g., NO formation), the use of large error tolerances does not result in significant errors. The large errors incurred at early times, however, have important implications, especially in developing and validating reaction mechanisms. A procedure commonly used for this purpose is to compare ignition delay times (e.g., time required for the temperature to increase by a specified amount) predicted by the mechanism with those measured in a shock tube (e.g., refs. 25 and 26). The temperature error plots show that caution must be exercised in using some of the codes to develop reaction mechanisms by applying the above procedure. If, for example, we assume that the ignition delay time is the time required for a 25 K rise in the temperature, values of ~ 11 and 3.5 μs are obtained for P1 and P2, respectively. At these times, the error in temperature ranges from 10 to 25 K for EPISODE, 2 to 5 K for LSODE, 15 to 200 K for GCKP84, and 0 to 10 K for CREK1D.

In contrast to EPISODE, LSODE, GCKP84, and CREK1D, CHEMEQ incurs virtually no errors during induction and early heat release (figs. 7, 8, 15, and 16). Therefore, this code can be used to generate accurate ignition delay times. CHEMEQ is superior in these regimes because of the very small step lengths that it selects (refs. 2 to 5). However, as pointed out by Young and Boris (ref. 10), the continued use of the hybrid method used in CHEMEQ results in the global errors increasing with time. For example, with CHEMEQ-A and EPS = 10^{-2}, the rms error in reactants has risen to almost 50 percent for P1 (fig. 7) and 25 percent for P2 (fig. 15). The situation is worse with CHEMEQ-B (figs. 8 and 16). During equilibration, for CREK1D, also, the errors grow (figs. 9 and 17) because the formulation used by it in this regime is based on that used in CHEMEQ. However, CREK1D incurs smaller errors than CHEMEQ for most of the species types and for the temperature. For EPS $\leq 10^{-3}$ both CHEMEQ and CREK1D either are more accurate than or compare favorably with LSODE during late heat release and equilibration.

Figures 4 to 17 and tables IV to VII show the large variations in the maximum errors for the different techniques. EPISODE and GCKP84 experience the greatest difficulty tracking the solutions at early times—rms errors in excess of 100 percent are obtained with the two codes. In contrast, the errors incurred by LSODE, CHEMEQ, and CREK1D are significantly less. Comparisons of the runs with the largest EPS value show that LSODE is the most accurate code for P1, and CREK1D for P2. Comparing the errors in the different regimes shows that CHEMEQ is the most accurate code during induction and early heat release. During late heat release and equilibration, however, the other codes are more accurate.

Examination of figures 4 to 17 and tables IV to VII shows that all techniques are tolerance effective in the sense that a decrease in the local tolerance generally results in decreased global errors. We note, however, that with LSODE not all plots show an error decrease with EPS. On the contrary, for some runs the error increases with a reduction in EPS (figs. 4, 5, 12, and 13). This behavior can be explained by examining the nature of the error control performed in LSODE. As discussed in the section "Computational Procedure," the error control selected to be performed by LSODE is mixed relative and absolute for species mole numbers and pure relative for the temperature. For pure relative error control, the estimated local truncation error, d_i, in species i approximately satisfies the inequality

$$d_i \leq \text{EPS} \, |\sigma_i| \tag{28}$$

For pure absolute error control, d_i approximately satisfies

$$d_i \leq \text{ATOLSP} \tag{29}$$

These two inequalities are approximate because the code controls only the rms norm of the estimated local truncation errors in all variables and not the estimated local truncation error in each variable.

Equations (28) and (29) show that, since $\sigma_i \ll 1$, relative error control is more accurate for a given value of the local error tolerance. Hence, relative error control is appropriate for the two test problems. However, when $\sigma_i = 0$, relative error control cannot be used. This problem is resolved by using a mixed relative and absolute error control, and

$$d_i \leq \text{EPS } |\sigma_i| + \text{ATOLSP} \tag{30}$$

Equation (30) shows that for the error control to be relative, ATOLSP must satisfy the inequality

$$\text{ATOLSP} \ll \text{EPS } |\sigma_i| \tag{31}$$

If ATOLSP $\gg$ EPS $|\sigma_i|$ the error control is absolute. In this study, we have considered only species with mole fractions $(x_i) \geq 0.1$ ppm. This value of x_i corresponds to $\sigma_i \simeq 3\times10^{-9}$ and 4×10^{-9}, respectively, for P1 and P2. Hence, for the error control to be always relative, ATOLSP must be less than 3×10^{-9} EPS and 4×10^{-9} EPS, respectively, for P1 and P2. Only the runs with EPS = 10^{-2} for P1 satisfy these requirements. Hence, they are the most accurate at early times when the mole numbers of many intermediate and product species have very small values. Note that for P2 even the standard solutions do not satisfy the requirement on ATOLSP. To ensure their accuracy, the standard solutions generated with LSODE-A and B were checked, respectively, against the solutions obtained with LSODE-A and B using EPS = 10^{-5} and ATOLSP = 10^{-15}, which satisfy equation (31). These comparisons showed agreement to three significant figures for all species with mole fractions ≥ 0.1 ppm. For LSODE-A agreement to three significant figures was obtained for all species, even those with mole fractions significantly smaller than 0.1 ppm. But for LSODE-B the agreement for mole fractions < 0.1 ppm was not good for NO. For all other species, however, good agreement was obtained for mole fractions $> 10^{-11}$. This observation indicates that LSODE-A is more accurate than LSODE-B.

Equation (30) also shows that for given values of EPS and ATOLSP, σ_i must satisfy the following inequality

$$\sigma_i \gg \frac{\text{ATOLSP}}{\text{EPS}} \equiv \sigma_{\min} \tag{32}$$

to achieve relative error control. As σ_i increases from zero, the error control becomes less absolute and is equally relative and absolute at $\sigma_{\min}$. For $\sigma_i > \sigma_{\min}$, the error control becomes increasingly relative as σ_i increases. Hence, the quantity $\sigma_{\min}$ may be regarded as the value at which, for increasing σ_i, the error control starts to change character from being more absolute to becoming more relative. Because $x_i = \sigma_i/\sigma_m$, the value of x_i $(=x_{\min})$ corresponding to $\sigma_{\min}$ is given by

$$x_{\min} = \text{ATOLSP}/(\text{EPS } \sigma_m) \tag{33}$$

For P1, $x_{\min} \simeq 3\times10^{-7}$ and 3×10^{-6} for EPS = 10^{-3} and 10^{-4} and the ATOLSP given in table IV. These values are attained by most of the species at $t \simeq 5$ and 7 μs, respectively (fig. 2). Hence, until these times the solutions with EPS = 10^{-3} and 10^{-4} are expected to be worse than or, at best, as accurate as the run with EPS = 10^{-2}. Examination of figures 4 and 5 shows that the errors in intermediates and products for EPS = 10^{-3} and 10^{-4} are worse than those for EPS = 10^{-2} until $t \simeq 6$ and 5 μs, respectively, for LSODE-A, and until $t \simeq 7$ and 6 μs, respectively, for LSODE-B. In addition, all maximum rms and maximum errors, almost all of which occur at $t > 7\mu$s, exhibit reductions with decreasing EPS (tables IV and V).

For P2 and LSODE-A, $x_{\min}$ has values of 2.5×10^{-5}, 2.5×10^{-4}, and 2.5×10^{-3}, respectively, for EPS = 10^{-2}, 10^{-3}, and 10^{-4}. Some of the species never reach these values (fig. 3). Hence, the errors do not show much sensitivity to changes in EPS (fig. 12 and tables VI and VII). For LSODE-B, however, the values used for ATOLSP ensure comparable levels of relative error control for EPS = 10^{-2} and 10^{-3}; for EPS = 10^{-4}, the control is more relative in the sense that it has a smaller value of ATOLSP/EPS. The errors, therefore, display decreases with reductions in EPS (fig. 13 and tables VI and VII). The sudden increases in the product errors around $t = 10$ μs were caused by the species NO, which LSODE-B had difficulty tracking (table VII).

The above discussion should be regarded as strictly approximate because it applies only to the estimated local truncation errors, whereas figures 4 and 5 give the estimated global errors, which represent the cumulative effects of the local errors. The number of integration steps required up to the relevant reaction time should therefore also be considered. However, the global errors accumulate in a complicated manner from the local errors. Other factors that must be taken into account are that LSODE controls only the norm of the estimated local errors and that different species reach mole fraction values of 0.1 ppm at different times. Finally, although we have ignored species with $x_i < 0.1$ ppm, they do incur errors whose magnitudes are controlled by ATOLSP and which grow with reaction time in the initial combustion regimes, for reasons previously given. It is therefore difficult to draw definitive conclusions about the ATOLSP to EMAX ratios required for combustion kinetics problems. For example, for P1, EPS = 10^{-3} is expected to produce more accurate results than EPS = 10^{-4} for the intermediates and products at early times, especially between 5 and 7 μs (see eq. (33) and the discussion following it), but figures 4 and 5 show the opposite behavior for both LSODE-A and LSODE-B. One conclusion that can, however, be made is that care must be exercised in specifying ATOLSP.

The effect of ATOLSP on solution accuracy is further illustrated for P2 and LSODE-B by the results presented in tables VI and VII (EPS = 10^{-4}) and XII and XIII (EPS = 10^{-5}). Note the significant error reductions obtained by decreasing ATOLSP. Comparing the errors given for LSODE-B in tables VI and VII with those in tables XII and XIII, respectively, shows that for the same value ($=10^{-8}$ or 10^{-9}) of ATOLSP, EPS = 10^{-5} does not produce significantly more accurate solutions than the larger EPS.

Tables XII and XIII show that, although the use of large values of ATOLSP can result in significant errors for the intermediate species and products, the effect on the temperature is small. Therefore, if the user is interested only in temperature versus time traces at early times, as for example in developing reaction mechanisms from ignition delay times, fairly large ATOLSP values can be assigned.

The results obtained above indicate that the ATOLSP needed to achieve acceptable accuracy depends as much on the nature of the solution as on the value specified for EPS and the minimum mole fraction to be considered in the error analysis. The estimate for ATOLSP given by equation (31) may not be small enough, as for example, P1 and EPS = 10^{-2} (table IV). On the other hand, the estimate may be needlessly conservative. For example, although the intermediate species increase much more rapidly for P2 than for P1, larger ATOLSP produced results that satisfied the accuracy criteria. Because the value needed for ATOLSP is a function of the problem, it can be obtained only after the problem is solved. The major problem associated with using LSODE to solve chemical kinetic rate equations is therefore the trial-and-error procedure necessary to obtain the optimal value of ATOLSP, that is, the value that minimizes the CPU time while satisfying prescribed accuracy requirements. Note that for P2, although the runs with LSODE-B and ATOLSP = 10^{-8} and 10^{-12} (EPS = 10^{-4}) required the same CPU time, the latter is significantly more accurate (table VI). In contrast, the runs with ATOLSP = 10^{-9}, 10^{-10}, and 10^{-11} required about 2.7, 1.7, and 1.7 s of CPU time, respectively. The trial-and-error search for the optimal ATOLSP can be time consuming, especially for large systems of ODE's. The use of an extremely small ATOLSP to ensure solution reliability can result in excessive CPU times. For example, for P2 the run using LSODE-B with EPS = 10^{-5} and ATOLSP = 10^{-11} required about 3.4 s of CPU time; in contrast, the run with ATOLSP = 10^{-15} required almost 20 s, although the solution was not significantly different.

The error control used in EPISODE and GCKP84 is pure relative for species with initially nonzero mole numbers and for the temperature; it is, however, pure absolute for species with initially zero mole numbers. Since most of the species have zero initial mole numbers for both test problems, the error control is mostly absolute. Hence, for the same value of EPS, EPISODE and GCKP84 are not as accurate as LSODE for solving chemical kinetic rate equations. To achieve comparable accuracy, especially at early times when σ_i is very small, small values have to be used for EPS (ref. 2). The runs with EPISODE and GCKP84 were therefore more expensive than the ones with LSODE (refs. 2 and 3). Modifying EPISODE to employ the same error control as LSODE produced significant reductions in execution times. Preliminary results with the revised EPISODE indicate that it is as fast as LSODE. For example, for P1 the runs with the modified EPISODE-A (EPS = 10^{-2}, 10^{-3}, and 10^{-4} and the ATOLSP given in table IV) required, respectively, 0.35, 0.41, and 0.61 s. The execution times compare very favorably with those required by LSODE-A: 0.37, 0.46, and 0.63 s (refs. 3 and 4). The above observations indicate that the error control used in EPISODE and GCKP84 is inappropriate for combustion kinetics problems.

Examination of tables IV to VII shows that temperature calculation method A does not necessarily produce less accurate solutions than method B. On the contrary, for most of the runs method A is more accurate for all codes; this result is most apparent for CHEMEQ and P1.

To provide a more comprehensive measure for comparing the accuracy of the methods examined, we adopted the following procedure: For each run, a mean integrated rms error, $\mathcal{E}_{rms}$, was defined as

$$\mathcal{E}_{rms} = \frac{1}{t_{end}} \int_0^{t_{end}} e_{rms}(t)\, dt \tag{34}$$

where t_{end} (≥ 1 ms) is the end of the integration time interval and $e_{rms}(t)$ is given by equation (27).

Equation (34) provides a single quantity that is a measure of the average error incurred in solving the complete problem. The integral in this equation was evaluated numerically using Simpson's rule (e.g., ref. 23), modified for unequal step sizes. For runs requiring an odd number of integration steps, the trapezoidal rule (ref. 23) was used on the last two mesh points.

The effects of h_0 on $\mathcal{E}_{rms}$ for GCKP84 are given in tables VIII and X for P1 and P2. Except for the run with EPS = 10^{-3} for P2, $h_0 = 10^{-6}$ s incurred either comparable or significantly smaller $\mathcal{E}_{rms}$ than $h_0 = 5\times10^{-8}$ s. Note, further, for P1 and $h_0 = 5\times10^{-8}$ s the substantial increase in $\mathcal{E}_{rms}$ when EPS is decreased to 10^{-4} (table VIII).

The variation of $\mathcal{E}_{rms}$ with ATOLSP is given in table XII for P2 using LSODE-B and EPS = 10^{-5}. This table illustrates the increasing accuracy obtained by reducing ATOLSP. It also shows that ATOLSP must be chosen carefully, as discussed previously.

The variations of $\mathcal{E}_{rms}$ with the user-specified local tolerance, EPS, are shown in figures 18 and 19 for P1 and P2, respectively. We have included the run with EPISODE-B and EPS = 5×10^{-4} in figure 19 because it was successfully completed. The $\mathcal{E}_{rms}$ given in figure 19 for LSODE-B and EPS = 10^{-4} was that obtained with ATOLSP = 10^{-12}. These figures show that all methods are tolerance effective (i.e., decreasing EPS results in reduced $\mathcal{E}_{rms}$). For both test problems temperature calculation method A is as accurate as

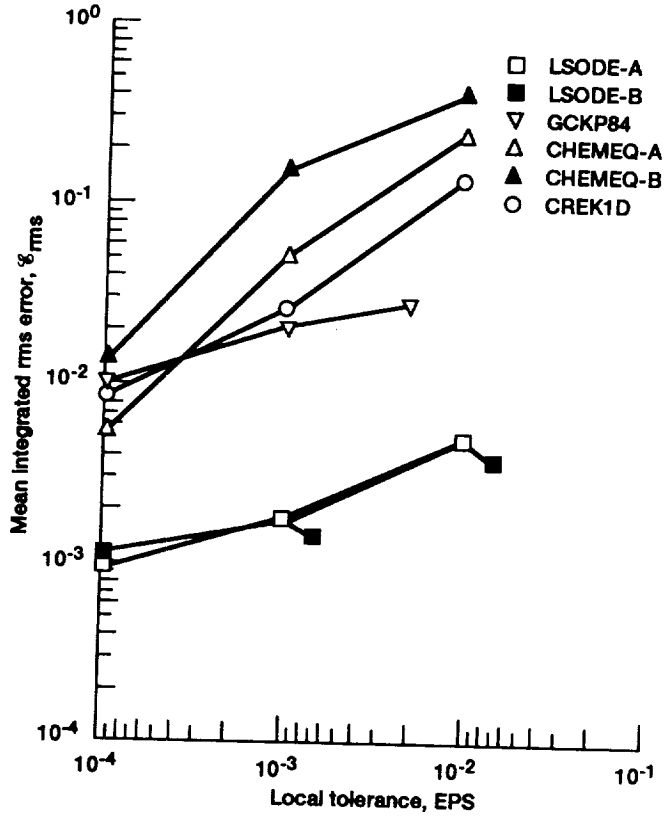

Figure 18.—Variation of the mean integrated rms error with the local tolerance for test problem 1.

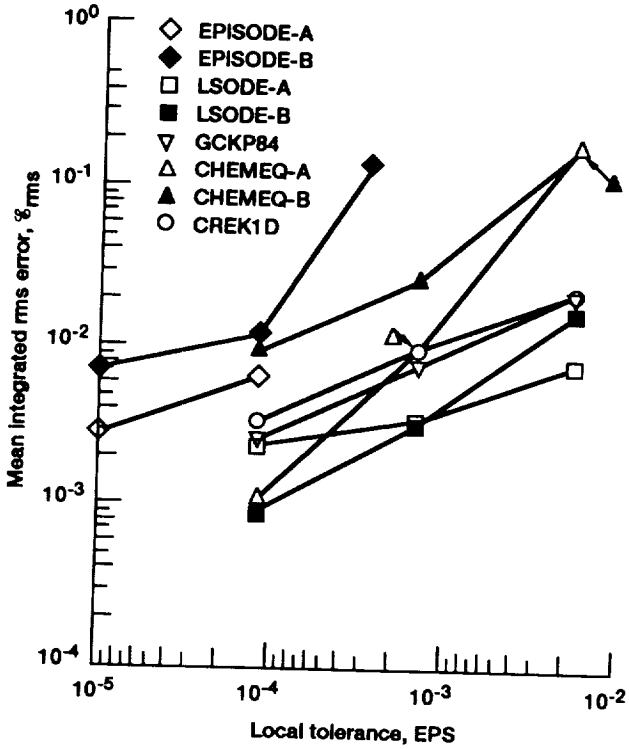

Figure 19.—Variation of the mean integrated rms error with the local tolerance for test problem 2.

method B. In many cases it is significantly more accurate, especially with CHEMEQ and EPISODE. For P2 and EPS = 10^{-4}, LSODE-B is more accurate than LSODE-A because it used a smaller ATOLSP (table VI).

For the same value of EPS, EPISODE and GCKP84 are significantly less accurate than LSODE (figs. 18 and 19) because the error control used in the two codes is inappropriate for chemical kinetics rate equations. For all techniques, note the significant discrepancies between the values specified for the user-supplied local tolerance and the errors actually incurred. With CHEMEQ-B, a value of EPS = 10^{-2} (1 percent) has resulted in an average error of almost 50 percent. The relatively large $\mathscr{E}_{rms}$ incurred by CREK1D and CHEMEQ is due to the difficulties which these codes experienced tracking the standard solutions during late heat release and equilibration. With LSODE, especially the runs with EPS = 10^{-2}, the correspondence between EPS and $\mathscr{E}_{rms}$ is better. These plots show that for a given value of EPS, LSODE is the most accurate code currently available for solving chemical kinetic rate equations. However, for P2, especially with the smallest EPS examined, GCKP84, CHEMEQ-A, and CREK1D compare favorably with LSODE (fig. 19).

Figures 20 and 21 present the variations of the computational work (expressed as the CPU time in seconds) with the mean integrated rms error for problems 1 and 2, respectively. Note the large differences in the CPU time required by the different codes to achieve comparable accuracy. For P1 and a ½ percent mean integrated global error, the CPU time varies from about 0.4 s for LSODE-A to over 40 s for CHEMEQ-A. In general, to produce an order of magnitude reduction in $\mathscr{E}_{rms}$ approximately doubles the computational cost. For both test problems LSODE is the most efficient code in the sense that it requires the least CPU time to attain a specified accuracy level.

Figures 20 and 21 show that the CPU times required by temperature calculation method A are less than, or compare

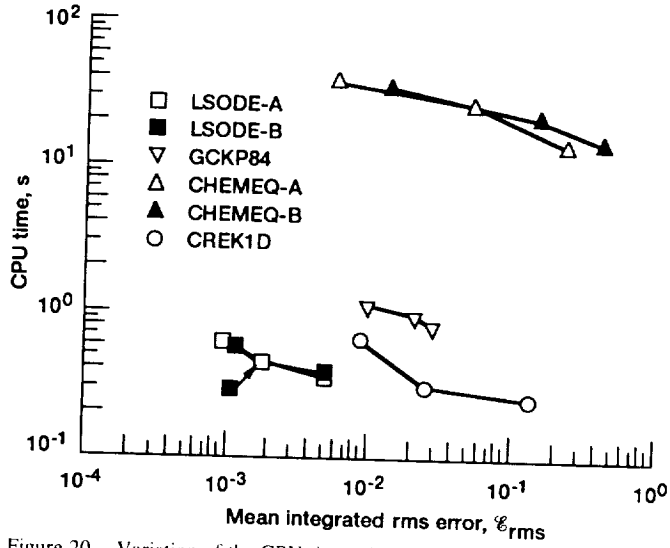

Figure 20.—Variation of the CPU time with the mean integrated rms error for test problem 1.

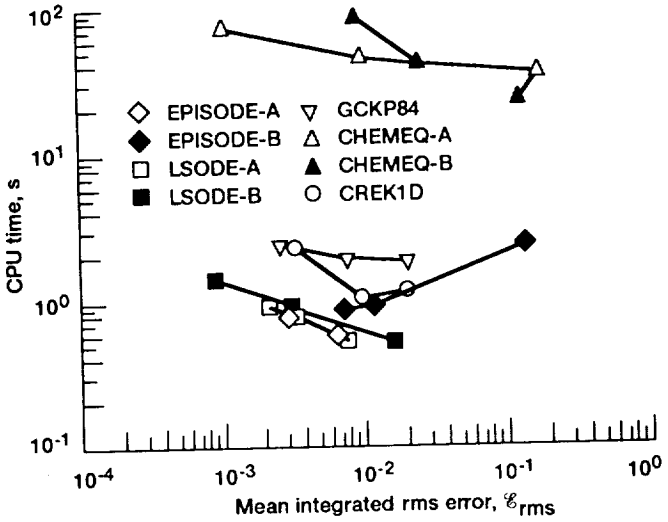

Figure 21.—Variation of the CPU time with the mean integrated rms error for test problem 2.

favorably with, those required by method B. This difference is most pronounced for CHEMEQ and EPISODE. For example, for P2 and a 1 percent $\mathscr{E}_{rms}$, CHEMEQ-A required only about half as much CPU time as CHEMEQ-B (fig. 21). Note that for EPISODE-B the computational work increases with increasing error.

EPISODE-A compares very favorably with LSODE for P2 (fig. 21). However, the solution generated by EPISODE can be strongly dependent on the value selected by the user for h_0 and a poor guess can result in incorrect and unstable solutions (refs. 2 to 5). It can also result in excessive CPU times. For example, the run using EPISODE-A with EPS = 10^{-4} and $h_0 = 10^{-8}$ s required about 129 s for P2; in contrast, the run with $h_0 = 10^{-7}$ s required only 0.59 s.

Conclusions

The accuracy of several codes (EPISODE, LSODE, GCKP84, CHEMEQ, and CREK1D) in solving combustion kinetic rate equations has been examined in detail. The accuracy studies were made by applying the codes to two practical combustion kinetics problems. Both problems described adiabatic, homogeneous, gas-phase chemical reactions and included all three combustion regimes: induction, heat release, and equilibration.

During induction and early heat release, when the species mole numbers and temperature change rapidly, EPISODE, LSODE, GCKP84, and CREK1D had difficulty tracking the solutions. The errors incurred by EPISODE and GCKP84 in these regimes were significantly larger than those incurred by LSODE and CREK1D. In contrast, the solutions generated with CHEMEQ displayed virtually no errors during induction and early heat release. However, during late heat release and equilibration, the errors obtained with CHEMEQ increased significantly. In these regimes, the other codes were more accurate.

Among the codes examined, LSODE was the most accurate for solving chemical kinetics problems. This study has also shown that LSODE is the most efficient code, that is, it required the least execution time to attain a specified accuracy. The major difficulty associated with its use is the trial-and-error procedure necessary to obtain optimal values for the local absolute error tolerances for the variables. A poor guess for the absolute error tolerance can result in excessive execution times or in seriously inaccurate solutions.

An important conclusion is that calculation of the temperature by solving the algebraic enthalpy conservation equation can be more accurate and efficient than integrating its differential equation.

Appendix A
Description of Codes Studied

The ordinary differential equations (ODE's) (2), (9) and (12) describing homogeneous gas phase chemical reactions can be generalized as follows:

$$\dot{y}_i \equiv \frac{dy_i}{dt} = f_i(\{y_k\}) \qquad i,k = 1,\ldots,N$$
$$y_i(t_0) = \text{given} \tag{A1}$$

where for temperature calculation method A (see the section "Evaluation of Temperature")

$$\left.\begin{aligned} y_i &= \sigma_i \qquad i = 1,\ldots,N_S \\ N &= N_S \end{aligned}\right\} \tag{A2}$$

for temperature calculation method B

$$\left.\begin{aligned} y_i &= \sigma_i \qquad i = 1,\ldots,N_S \\ y_{N_S+1} &= T \\ N &= N_S + 1 \end{aligned}\right\} \tag{A3}$$

and for the code GCKP84

$$\left.\begin{aligned} y_i &= \sigma_i \qquad i = 1,\ldots,N_S \\ y_{N_S+1} &= V \qquad y_{N_S+2} = \rho \qquad y_{N_S+3} = T \\ N &= N_S + 3 \end{aligned}\right\} \tag{A4}$$

In vector notation equation (A1) becomes

$$\dot{\underline{y}} \equiv \frac{d\underline{y}}{dt} = \underline{f}(\underline{y}) \qquad \underline{y}(t_0) = \text{given} \tag{A5}$$

where the underscore is used to denote a vector quantity. A matrix is denoted by a boldface letter. This notation is used throughout this appendix. In equation (A5) the N-dimensional column vectors $\underline{y}$ and $\underline{f}$ contain the dependent variables and their temporal derivatives, respectively.

The initial-value problem is to determine values for $\{y_i\}$ at one or more times in a prescribed integration interval, given $\{f_i\}$ and the values $\{y_i(t_0)\}$ at the initial time t_0. We now describe the codes studied in the present work and how they solve the above problem.

EPISODE AND LSODE

Both these codes use linear multistep methods of the form (refs. 6 to 9)

$$Y_{i,n} = \sum_{j=1}^{K_1} \alpha_{j,n} Y_{i,n-j} + h_n \sum_{j=0}^{K_2} \beta_{j,n} f_{n-j} \qquad i = 1,\ldots,N \tag{A6}$$

where $Y_{i,n}$ is an approximation to the exact solution $y_i(t_n)$, $f_{i,n}$ (equal to $f_i(\{Y_{k,n}\})$) is an approximation to the exact derivative $\dot{y}_i(t_n)$ (equal to $f_i(\{y_k(t_n)\})$), and the $\{\alpha_{j,n}\}$ and $\{\beta_{j,n}\}$ ($\beta_{0,n} > 0$) are associated with the particular formula selected by the user. The options include a variable-step, variable-order implicit Adams method (suitable for nonstiff problems) of orders 1 to 12, and a variable-step, variable-order backward differentiation formula (BDF) method (suitable for stiff problems) of orders 1 to 5. As discussed in the section "Methods and Codes Examined," the BDF method was more efficient for the problems examined in this study. Therefore, the discussion is restricted to this method. For a BDF method of order q, $K_1 = q$, $K_2 = 0$, and equation (A6) reduces to

$$Y_{i,n} = \sum_{j=1}^{q} \alpha_{j,n} Y_{i,n-j} + h_n \beta_{0,n} f_{i,n} \qquad i = 1,\ldots,N \tag{A7}$$

The step length h_n can vary from one step to the next in EPISODE but is held constant for $q + 1$ consecutive successful steps in LSODE. Hence, for EPISODE, $\{\alpha_{j,n}\}$ and $\{\beta_{j,n}\}$ can vary from one step to the next, but in LSODE they are predetermined constants corresponding to the order used.

Both codes use a predictor-corrector process to solve for $\underline{Y}_n$. An explicit method generates a predicted value, $\underline{Y}_n^{[0]}$, which is then corrected by iterating equation (A7) to convergence, that is, the improved estimates $\underline{Y}_n^{[m]}$ ($m = 1,\ldots,M$) are produced until the magnitude of the difference $(\underline{Y}_n^{[m]} - \underline{Y}_n^{[m-1]})$, in EPISODE, or $(h_n\dot{\underline{Y}}_n^{[m]} - h_n\dot{\underline{Y}}_n^{[m-1]})$, in LSODE, approaches zero within a specified accuracy. Here, $\underline{Y}_n^{[m]}$ and $\dot{\underline{Y}}_n^{[m]}$ are, respectively, the approximations generated for $\underline{Y}_n$ and $\underline{f}_n$ on the m^{th} iteration, the integer M is the number of iterations required for convergence, and $\underline{Y}_n^{[m]}$ is accepted as the numerical solution at t_n, provided it satisfies a prescribed local accuracy requirement. At each iteration m, $h_n\dot{\underline{Y}}_n^{[m]}$ is computed in LSODE from $\underline{Y}_n^{[m]}$ via the relation

$$\underline{Y}_n^{[m]} = \sum_{j=1}^{q} \alpha_{j,n} \underline{Y}_{n,j} + h_n \beta_{0,n} \dot{\underline{Y}}_n^{[m]} \tag{A8}$$

so that the pair $(\underline{Y}^{[m]}, h_n\dot{\underline{Y}}_n^{[m]})$ satisfies the BDF method (eq. (A7)) exactly. The predicted values of $\underline{Y}_n$ and $h_n\underline{f}_n$, denoted by $h_n\dot{\underline{Y}}_n^{[0]}$, also satisfy equation (A8).

The predicted quantities $\underline{Y}_n^{[0]}$ and $h_n\dot{\underline{Y}}_n^{[0]}$ are obtained by a q^{th}-order Taylor series expansion as follows: The history of the solution is maintained in the Nordsieck array (which is

a Taylor series array) $\mathbf{z}_n$ of size $N \times (q+1)$ (e.g., ref. 1). The i^{th} row ($i = 1,\ldots,N$) contains the $q+1$ elements $Y_{i,n}$, $h_n\dot{Y}_{i,n}$, $h_n^2/2!\ \ddot{Y}_{i,n},\ldots,h_n^q/q!\ Y_{i,n}^{(q)}$, where $Y_{i,n}^{(j)}$ is the approximation to $(d^jY_i/dt^j)_{t_n}$. The $(q+1)$ columns of $\mathbf{z}_n$ are numbered from 0 to q, and the j^{th} column ($j = 0,1,\ldots,q$), which will be denoted by the vector $z_n(j)$, contains the vector $h_nY_n^{(j)}/j!$ of the j^{th}-order scaled derivatives. If $\mathbf{z}_{n-1}$ has been obtained, the predicted history matrix, $\mathbf{z}_n^{[0]}$, at t_n is given by (ref. 1)

$$\mathbf{z}_n^{[0]} = \mathbf{z}_{n-1}\mathbf{A}(q) \tag{A9}$$

where $\mathbf{A}(q)$ is a $(q+1) \times (q+1)$ matrix, with element $A_{jk}(q)$ given by

$$A_{jk}(q) = \begin{pmatrix} 0 & j < k \\ \binom{j}{k} & j \geq k \end{pmatrix} \qquad j,k = 0,1,\ldots,q$$

The binomial coefficient, $\binom{j}{k}$, is defined as

$$\binom{j}{k} = \frac{j!}{k!\ (j-k)!}$$

Thus, the predicted array $\mathbf{z}_n^{[0]}$ is obtained by a simple q^{th}-order Taylor series expansion by using equation (A9). The matrix $\mathbf{z}_n^{[0]}$ contains predicted values of $\underline{Y}_n$ and its scaled derivatives up to order q, the current method order. Note, however, that because a q^{th}-order Taylor series expansion method is used, $z_n^{[0]}(q) = z_{n-1}(q)$.

The estimates $Y_n^{[m]}$ and, in LSODE, $h_n\dot{Y}_n^{[m]}$ ($m = 1,\ldots,M$) are generated, as described below, until the iteration converges. The local error test is then applied and, if passed, the Nordsieck history matrix $\mathbf{z}_n$ is constructed by using the relation

$$\mathbf{z}_n = \mathbf{z}_n^{[0]} + e_n l_n(q) \tag{A10}$$

where

$$e_n = Y_n^{[M]} - \underline{Y}_n^{[0]}$$

in EPISODE, and

$$e_n = h_n\dot{Y}_n^{[M]} - h_n\dot{Y}_n^{[0]}$$

in LSODE. The $(q+1)^{\text{th}}$-dimensional vector $\underline{l}_n(q)$

$$l_n(q) = (l_{0,n}(q),\ l_{1,n}(q),\ldots,l_{q,n}(q)) \tag{A11}$$

contains the method coefficients for the Nordsieck history formulation of the q^{th}-order BDF method. Because EPISODE and LSODE use different calculation procedures, the $\{l_{i,n}\}$ values are, in general, different in the two codes. For EPISODE, $l_n(q)$ depends on the method order and the step length history, satisfies $l_{0,n}(q) = 1$ and $l_{1,n} = 1/\beta_{0,n}$, and has to be recomputed at the start of each step. For LSODE, $l_n(q)$ is a function of only q, satisfies $l_{0,n}(q) = \beta_{0,n}$ and $l_{1,n}(q) = 1$, and has to be recomputed only when the method order is changed.

To correct the initial estimate $Y_n^{[0]}$ (i.e., to solve equations (A7)), both codes include a variety of iteration techniques. For combustion kinetics problems the most efficient is the Newton-Raphson iteration (ref. 2), which is given by the recursive relation

$$\mathbf{P}(Y_n^{[m+1]} - Y_n^{[m]}) = \sum_{j=1}^{q} \alpha_{j,n}Y_{n-j} + h_n\beta_{0,n}f(Y_n^{[m]}) - Y_n^{[m]} \tag{A12}$$

for $m = 0,1,\ldots,M-1$. The $N \times N$ iteration matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \mathbf{I} - h_n\beta_{0,n}\mathbf{J} \tag{A13}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{J}$ is the Jacobian matrix, with element J_{ij} given by

$$J_{ij} = \partial f_i/\partial y_j \qquad i,j = 1,\ldots,N$$

For this method, much computation time is required to form the Jacobian matrix and to perform the linear algebra necessary to solve equation (A12). To reduce this computational work, $\mathbf{P}$ is not updated at every iteration. For further savings, it is updated only when it has been determined to be absolutely necessary for convergence. Hence, the iteration matrix is only accurate enough for the iteration to converge, and the codes may use the same matrix over several steps of the integration. In any case, both EPISODE and LSODE update $\mathbf{P}$ at least every 20th step. The linear algebra required to solve equation (A12) is performed by the LU method (e.g., ref. 27), rather than by explicitly inverting the matrix, which requires prohibitive amounts of computer time (ref. 23).

Convergence of the estimates is ascertained as discussed below. EPISODE constructs a vector $\underline{Y}_{\max}$, which depends on the user-specified value for the local error control IERROR as follows:

IERROR = 1 (absolute error control):

$$Y_{\max_i} = 1 \qquad i = 1,\ldots,N$$

IERROR = 2 (pure relative error control):

$$Y_{\max_i} = |Y_{i,n-1}| \qquad i = 1,\ldots,N$$

IERROR = 3 (semirelative error control):

$$Y_{\max_i} = \max\{|Y_{i,n-1}|, |Y_{i,n-2}|\} \quad \text{for } i \text{ that satisfy } y_i(t_0) \neq 0$$
$$= \max\{1, |Y_{i,n-1}|\} \quad \text{for } i \text{ that satisfy } y_i(t_0) = 0 \tag{A14}$$

The test for iteration convergence is based on the successive differences ($\underline{Y}_n^{[m]} - \underline{Y}_n^{[m-1]}$), as compared with $\underline{Y}_{\max}$ and the user-supplied local error tolerance parameter EPS. Convergence is said to occur if

$$\delta_m \equiv \left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{Y_{i,n}^{[m]} - Y_{i,n}^{[m-1]}}{Y_{\max_i}} \right)^2 \right)^{1/2} \leq C_E \text{EPS} \tag{A15}$$

In LSODE an error weight vector $\underline{\text{EWT}}$ is constructed as follows:

$$\text{EWT}_i = \text{RTOL}_i \, |Y_{i,n-1}| + \text{ATOL}_i \qquad i = 1,\ldots,N$$

where RTOL_i and ATOL_i are, respectively, the user-supplied local relative and local absolute error tolerances for the i^{th} component. Both RTOL and ATOL can be specified either as a scalar or an array, as discussed in the section "Computational Procedure." The value of the user-supplied parameter ITOL indicates whether RTOL and ATOL are scalars or arrays. ITOL has four possible values which correspond to the types of RTOL and ATOL as follows:

ITOL = 1: scalar RTOL and scalar ATOL
ITOL = 2: scalar RTOL and array ATOL
ITOL = 3: array RTOL and scalar ATOL
ITOL = 4: array RTOL and array ATOL

The convergence test is based on the successive differences ($h_n \underline{\dot{Y}}_n^{[m]} - h_n \underline{\dot{Y}}_n^{[m-1]}$) as compared with $\underline{\text{EWT}}$, and is given by

$$\delta_m = \left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{h_n \dot{Y}_{i,n}^{[m]} - h_n \dot{Y}_{i,n}^{[m-1]}}{\text{EWT}_i} \right)^2 \right)^{1/2} \overset{?}{\leq} C_L \tag{A16}$$

The factors C_E and C_L in equations (A15) and (A16) are chosen to make the convergence tests consistent with the local truncation error tests. In particular, $C_E = 0.1 \; \mathfrak{I}_E(q)$ and $C_L = \mathfrak{I}_L(q)/2(q+2)$, where $\mathfrak{I}_E(q)$ and $\mathfrak{I}_L(q)$ are the test constants used, respectively, in EPISODE and LSODE for the local error test (eqs. (A27) and (A28)) and where the variable q indicates the method order.

If convergence is not achieved after the first iteration, the codes anticipate the magnitude of δ_m one iteration in advance by assuming that the estimates converge linearly. Thus, δ_{m+1}, which does not yet exist, is estimated by

$$\delta_{m+1} = \delta_m \frac{\delta_m}{\delta_{m-1}} = \delta_m C_m,$$

where C_m ($= \delta_m / \delta_{m-1}$) is the convergence rate. This assumption is used to modify the convergence tests (eqs. (A15) and (A16)) as follows:

for EPISODE

$$\delta_m' \overset{?}{\leq} C_E \text{ EPS}$$

where

$$\delta_m' = \delta_m \min(1, C_m')$$

$$C_m' = \max(0.1 \; C_{m-1}, C_m)$$

and for LSODE

$$\delta_m' \overset{?}{\leq} C_L$$

where

$$\delta_m' = \delta_m \min(1, 1.5 \; C_m')$$

$$C_m' = \max(0.2 C_{m-1}, C_m)$$

Now, at least two iterations are required to compute C_m. For the first iteration, C_m' is set equal to 1 in EPISODE and equal to the last value of C_m from the previous step in LSODE. For the first iteration of the first step and after every update of the Jacobian matrix, LSODE sets C_m' equal to 0.7.

If the corrector iteration fails to converge in three iterations, h_n is reduced by a factor of four if **P** is current and the step is retried; otherwise, **P** is updated at $\underline{y} = \underline{Y}_n^{[0]}$, and the step is retried. The same corrective actions are taken by LSODE if $C_m > 2$ after the second iteration. In the event of a singular iteration matrix, both codes reduce h_n by a factor of four and attempt the solution with the new step length. The integration is abandoned if either the step size is reduced below a minimum value (both codes) or 10 convergence failures have occurred (LSODE).

If the corrector converges after M (≤ 3) iterations, an estimate of the local truncation error is made, as described below. For a BDF method of any order k, the local truncation error, $\underline{d}_n(k)$ at t_n is given by

$$\underline{d}_n(k) = \mathfrak{C}_{k+1} h_n^{k+1} \, \underline{y}^{(k+1)}(t_n) \tag{A17}$$

where the variable k denotes the method order and the constant $\mathfrak{C}_{k+1}$ depends on the method formulation. For the variable-step method used in EPISODE (ref. 6),

$$\mathfrak{C}_{k+1} = \frac{\prod_{j=1}^{k} \xi_j}{(k+1)! \; l_{1,n}(k)} \tag{A18}$$

where

$$\xi_j = \frac{t_n - t_{n-j}}{h_n} \qquad j = 1,\ldots,k \tag{A19}$$

and $l_{1,n}(k)$ is the second element of the $(k+1)^{\text{th}}$-dimensional coefficient vector $l_n(k)$ for the k^{th}-order method (see eq. (A11)). For the formulation used in LSODE (ref. 28)

$$\mathcal{C}_{k+1} = \frac{1}{k+1} \tag{A20}$$

The error $d_n(q)$ in the q^{th}-order method (i.e., $k = q$) used on the n^{th} step (i.e., (t_{n-1},t_n)) is estimated as follows: As discussed previously, the last column of $\mathbf{z}_n$, $z_n(q)$ contains the vector $h_n^q Y_n^{(q)}/q!$ and that of $\mathbf{z}_n^{[0]}$, $z_n^{[0]}(q)$ contains the vector $h_n^q Y_{n-1}^{(q)}/q!$. The difference of $z_n(q)$ and $z_n^{[0]}(q)$ gives

$$z_n(q) - z_n^{[0]}(q) = \frac{h_n^{q+1}}{q!} Y_n^{(q+1)} + O(h_n^{q+2}) \tag{A21}$$

by using the mean value theorem for derivatives. From equation (A10) the above difference is seen to be equal to $l_{q,n}(q)\, e_n$, which, upon substitution into equation (A21), gives

$$h_n^{q+1} Y_n^{(q+1)} \simeq q!\; l_{q,n}(q) e_n \tag{A22}$$

if higher-order terms are neglected. This approximation is used in LSODE. EPISODE, however, takes into account errors in the past values and uses the following expression (ref. 6):

$$h_n^{q+1} Y_n^{(q+1)} \simeq \frac{1}{\gamma_n} e_n \tag{A23}$$

where

$$\gamma_n = \frac{\prod_{j=1}^{q} \xi_j}{(q+1)!} \left[1 + \prod_{j=2}^{q} \left(\frac{t_n - t_{n-j}}{t_{n-1} - t_{n-j}} \right) \right] \tag{A24}$$

and where ξ_j is given by equation (A19).

By substituting the above expressions for $h_n Y_n^{(q+1)}$ and the appropriate equation (A18) or (A20) into equation (A17) (with $k = q$) and simplifying the resulting expressions, we obtain the following estimates for $d_n(q)$:

For EPISODE

$$d_n(q) = \frac{1}{l_{1,n}(q)} \left[1 + \prod_{j=2}^{q} \left(\frac{t_n - t_{n-j}}{t_{n-1} - t_{n-j}} \right) \right]^{-1} e_n \tag{A25}$$

and for LSODE

$$d_n(q) = \frac{q! l_{q,n}(q)}{q+1} e_n = \frac{l_{0,n}(q)}{q+1} e_n \tag{A26}$$

because $q! l_{q,n}(q) = l_{0,n}(q)$ for the formulation used in LSODE (ref. 28).

The local error tests used in the two codes are as follows:

For EPISODE

$$\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{d_{i,n}}{Y_{\max_i}} \right)^2 \right)^{1/2} \overset{?}{\leq} \text{EPS}$$

which, upon using equation (A25) can be written as

$$D_q \equiv \frac{\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{e_{i,n}}{Y_{\max_i}} \right)^2 \right)^{1/2}}{\mathfrak{I}_E(q)\ \text{EPS}} \overset{?}{\leq} 1 \tag{A27}$$

where the test constant $\mathfrak{I}_E(q)$ is given by

$$\mathfrak{I}_E(q) = l_{1,n}(q) \left[1 + \prod_{j=2}^{q} \left(\frac{t_n - t_{n-j}}{t_n - t_{n-j}} \right) \right]$$

And for LSODE

$$\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{d_{i,n}}{\text{EWT}_i} \right)^2 \right)^{1/2} \overset{?}{\leq} 1$$

By using equation (A26), the above inequality can be expressed as

$$D_q \equiv \frac{\left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{e_{i,n}}{\text{EWT}_i} \right)^2 \right)^{1/2}}{\mathfrak{I}_L(q)} \overset{?}{\leq} 1 \tag{A28}$$

where the test constant $\mathfrak{I}_L(q)$ is given by

$$\mathfrak{I}_L(q) = \frac{q+1}{l_{0,n}(q)}$$

If the error test fails, the following corrective actions are taken. In EPISODE h_n is reduced so that equation (A27) is satisfied (see eqs. (A29) and (A30)), and the step is retried. If the results with the new step length do not pass the error test, h_n is reduced by a factor of five. After three and more error test failures EPISODE reduces the step length by a factor of 10 and reduces the method order by one if it is greater than one. If an error test failure occurs with $q = 1$, the Nordsieck history matrix $\mathbf{z}_{n-1}$ is reconstructed from $\underline{Y}_{n-1}$ and $\underline{f}(\underline{Y}_{n-1})$. After the first error test failure, LSODE reduces h_n and/or q by one and then retries the step. If the error test is again not satisfied, h_n is reduced by a factor of five. After three and more such failures, the method order is reduced to one if it is greater than one, the step size is reduced by a factor of 10, and the step is retried with a new Nordsieck history matrix $\mathbf{z}_{n-1}$, which is constructed from $\underline{Y}_{n-1}$ and $\underline{f}(\underline{Y}_{n-1})$. Both codes abandon the integration if h_n is reduced below a minimum value. The maximum number of error test failures allowed is seven in EPISODE and 10 in LSODE, after which an error exit is taken.

If the error test is passed, the step is accepted as successful, and the entire Nordsieck history array, $\mathbf{z}_n$, at t_n is updated by using equation (A10).

Periodically, both codes attempt to change the step length and/or the method order to minimize computational work while maintaining prescribed accuracy. After every step on which no convergence test or local error test failure occurs, EPISODE attempts to use a larger step length at the same method order. The new step size $h'(q)$, where the variable q denotes the order to be used on the next step, is chosen such that it exactly satisfies the local error bound (eq. (A27)) by assuming that the highest derivative remains constant. Then, because $\underline{d}_n$ varies as h_n^{q+1} (eq. (A17)),

$$r_{\text{same}} \equiv \frac{h'(q)}{h_n} = \left(\frac{1}{D_q}\right)^{\frac{1}{q+1}} \tag{A29}$$

where r is the ratio of the step length to be attempted on the next step to its current value and the subscript "same" indicates that the current order (q) is to be used on the next step. To allow for inaccuracies in the error estimate, certain safety factors are built into the calculation procedure for $h'(q)$ to produce a smaller value than that given by equation (A29). The formula used in EPISODE for r_{same} is

$$r_{\text{same}} = \frac{1}{(5D_q)^{\frac{1}{q+1}} + 10^{-6}} \tag{A30}$$

To increase the efficiency, both codes consider changing the method order to $q - 1$ or $q + 1$ at periodic intervals. After an unsuccessful step or when the current order equals the maximum order, $q_{\max}$, the choice $q + 1$ is not considered. Also, if $q = 1$, the choice $q - 1$ is rejected. For each method order q' the step size $h'(q')$ that will exactly satisfy the local error bound is obtained by using the procedure outlined above for $q' = q$ (eq. (A29)).

For the case $q' = q - 1$, $\underline{d}_n(q-1)$ varies as $h_n^q \underline{y}^{(q)}(t_n)$ (eq. (A17)), which is equal to $q!\underline{z}_n(q)$. The local error test for $q' = q - 1$ is as follows:

For EPISODE

$$D_{q-1} \equiv \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{z_{i,n}(q)}{Y_{\max_i}}\right)^2\right)^{1/2}}{\mathfrak{I}_E(q-1)\ \text{EPS}} \overset{?}{\leq} 1$$

where $z_{i,n}(q)$ is the i^{th} element of $\underline{z}_n(q)$ and (ref. 6)

$$\mathfrak{I}_E(q-1) = \frac{l_1(q-1)}{\prod_{j=1}^{q-1}\xi_j}$$

And for LSODE

$$D_{q-1} \equiv \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{z_{i,n}(q)}{\text{EWT}_i}\right)^2\right)^{1/2}}{\mathfrak{I}_L(q-1)} \overset{?}{\leq} 1$$

where (ref. 28)

$$\mathfrak{I}_L(q-1) = \frac{1}{(q-1)!}$$

The step length ratio, if the order is to be reduced to $q - 1$, is then given by

$$r_{\text{down}} \equiv \frac{h'(q-1)}{h_n} = \left(\frac{1}{D_{q-1}}\right)^{\frac{1}{q}} \tag{A31}$$

where the subscript "down" indicates that the order is to be decreased. If $q = 1$, r_{down} is set equal to zero because q cannot be decreased.

For the case $q' = q + 1$, $\underline{d}_n(q+1)$ varies as $h_n^{q+2}\underline{y}^{(q+2)}(t_n)$, which is estimated by differencing the quantity $h^{q+1}\underline{Y}^{(q+1)}$ over the last two steps and then using the mean value theorem for derivatives. For EPISODE equation (A23) gives

$$\frac{e_n}{\gamma_n} - \left(\frac{h_n}{h_{n-1}}\right)^{q+1} \frac{e_{n-1}}{\gamma_{n-1}} \cong h_n^{q+1} Y_n^{(q+1)} - h_n^{q+1} Y_n^{(q+1)}$$

$$\cong h_n^{q+2} Y_n^{(q+2)} + O(h_n^{q+3})$$

where γ_n is given by equation (A24). For LSODE, equations (A22) and (A26) show that the approximation for $h_n^{q+1} Y_n^{(q+1)}$ is given by $l_{0,n}(q)e_n$. Because the methods used in this code are based on a constant step size, the quantity $l_{0,n}(q)[e_n - e_{n-1}]$ gives

$$l_{0,n}(q)[e_n - e_{n-1}] \cong h_n^{q+2} Y_n^{(q+2)} + O(h_n^{q+3})$$

The local error test for $q' = q + 1$ is given by
For EPISODE

$$D_{q+1} \equiv \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left[\frac{e_{i,n} - \left(\frac{\gamma_n}{\gamma_{n-1}}\right)\left(\frac{h_n}{h_{n-1}}\right)^{q+1} e_{i,n-1}}{Y_{\max_i}}\right]^2\right)^{1/2}}{\mathfrak{I}_E(q+1)\ \mathrm{EPS}} \overset{?}{\leq} 1$$

where (ref. 6)

$$\mathfrak{I}_E(q+1) = \frac{(q+2)l_{1,n}(q+1)}{\xi_{q+1}}\left[1 + \prod_{j=2}^{q}\left(\frac{t_n - t_{n-j}}{t_{n-1} - t_{n-j}}\right)\right]$$

And for LSODE

$$D_{q+1} \equiv \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{e_{i,n} - e_{i,n-1}}{\mathrm{EWT}_i}\right)^2\right)^{1/2}}{\mathfrak{I}_L(q+1)} \overset{?}{\leq} 1$$

where (ref. 28)

$$\mathfrak{I}_L(q+1) = \frac{q+2}{l_{0,n}(q)}$$

The step length ratio, if the order is to be increased to $q + 1$, is then given by

$$r_{\mathrm{up}} \equiv \frac{h'(q+1)}{h_n} = \left(\frac{1}{D_{q+1}}\right)^{\frac{1}{q+2}} \qquad \text{(A32)}$$

where the subscript "up" indicates that the order is to be increased. If $q = q_{\max}$ and in LSODE after a failed step, r_{up} is set equal to zero to prevent an order increase.

For reasons given previously certain safety factors are built into the step length ratios (eqs. (A29), (A31), and (A32)). The formula used in EPISODE for r_{same} is given by equation (A30); the other two ratios are computed as follows:

$$r_{\mathrm{down}} = \frac{1}{(5D_{q-1})^{\frac{1}{q}} + 10^{-6}}$$

$$r_{\mathrm{up}} = \frac{1}{(10D_{q+1})^{\frac{1}{q+2}} + 10^{-6}}$$

The formulas used in LSODE to calculate the step length ratios are

$$r_{\mathrm{down}} = \frac{1}{1.3\left\{(D_{q-1})^{\frac{1}{q}} + 10^{-6}\right\}}$$

$$r_{\mathrm{same}} = \frac{1}{1.2\left\{(D_q)^{\frac{1}{q+1}} + 10^{-6}\right\}}$$

$$r_{\mathrm{up}} = \frac{1}{1.2\left\{(D_{q+1})^{\frac{1}{q+2}} + 10^{-6}\right\}}$$

The order corresponding to the maximum step length ratio $r = \max(r_{\mathrm{down}}, r_{\mathrm{same}}, r_{\mathrm{up}})$ and the step length ratio r are selected to be attempted on the next step if, after a successful step, $r \geq 1.3$ (EPISODE) or 1.1(LSODE); otherwise, both changes are rejected. After a failed step, the order is decreased in LSODE if $r_{\mathrm{down}} > r_{\mathrm{same}}$; however, r is set equal to one if it is greater than one. Several additional tests are performed on r before the step length to be attempted next is selected. These tests may be summarized as follows:
For EPISODE

$$r \leftarrow \min\left\{\frac{h_{\max}}{h_n},\ r_{\max},\ \max\left(r,\ \frac{h_{\min}}{h_n},\ r_{\min}\right)\right\}$$

where the arrow denotes the replacement operator. And for LSODE

$$r \leftarrow \min\left\{\frac{h_{\max}}{h_n},\ r_{\max},\ \max\left(r,\ \frac{h_{\min}}{h_n}\right)\right\}$$

In EPISODE, $h_{\min}$ and $h_{\max}$ are set equal to, respectively, h_0, the user-supplied value for the step length to be attempted on the first step, and $10(t_{\text{out}} - t_{\text{out,old}})$ where t_{out} is the current time at which the solution is required and $t_{\text{out,old}}$ is the previous value of t_{out}. On the first call to EPISODE t_{out} is set equal to t_0, the initial value of t. In LSODE, however, $h_{\min}$ (default value = 0) and $h_{\max}$ (default value = ∞) are user-supplied optional input parameters. The quantity $r_{\min}$ (used only in EPISODE) is set equal to 0.1, and $r_{\max}$ depends on the code. In EPISODE, $r_{\max}$ is set equal to 10 for the first 10 integration steps; thereafter, it is set equal to 1.5. In LSODE, $r_{\max}$ is normally set equal to 10; for the first step length increase following either a convergence or local error test failure, it is set equal to two. In both codes for the first step length increase for the problem $r_{\max}$ is set equal to 10^4 if no convergence or error test failure has occurred.

After the step length ratio r has been computed, the step length h' to be attempted on the next step is given by

$$h' = rh_n.$$

Changes in method order (and step length in LSODE) are attempted only after S successful steps with the same order (and step length in LSODE), where S is normally set equal to $q + 1$. However, if an unsuccessful step occurs, the step length and/or order may be reduced. Following a failed error test or a failed convergence test, if $\mathbf{P}$ is current, EPISODE resets S equal to 2 if it is less than two, but LSODE resets S equal to $q + 1$ irrespective of its current value. If three or more error test failures occur on any one step S is set equal to five in LSODE and either $q + 1$ (if $q > 1$) or 10 (if $q = 1$) in EPISODE. Following a step for which the method order is not changed EPISODE sets S equal to 2. If method order and step length changes are rejected because $r < 1.1$, LSODE sets S equal to 3.

After every $S - 1$ successful steps, if $q < q_{\max}$, EPISODE saves e and γ, and LSODE saves e, so that r_{up} can be computed. To minimize storage requirements, the vector e is saved as the $(q_{\max} + 1)^{\text{th}}$ column of $\mathbf{z}$.

If the step size and/or method order is changed on the n^{th} step, $\mathbf{z}_n$ has to be modified. For the case $q' = q$, $h' \neq h_n$, the j^{th} column ($j = 0,1,..,q$) is multiplied by $(h'/h_n)^j$. For the case $q' = q - 1$, $h' \neq h_n$, the last column of the old $\mathbf{z}_n$ is ignored because it is not needed on subsequent steps. In addition, EPISODE adjusts the first q columns to reflect the reduced set of data represented by $\mathbf{z}_n$ (ref. 6). In both codes the above scaling by powers of (h'/h_n) is performed on the first q columns. For the case $q' = q + 1$, $h' \neq h_n$, EPISODE adds a column of zeros, representing $z_n(q + 1)$, to $\mathbf{z}_n$. LSODE sets $z_n(q + 1)$ equal to $h_n^{q+1} Y_n^{(q+1)}/(q + 1)!$, which is equal to $l_{q,n}(q)e_n/(q + 1)$ (see eq. (A22)). Both codes then rescale all $q + 2$ columns of $\mathbf{z}_n$ by powers of h'/h_n to account for any change in the step size.

The solution values at prescribed output times $t_{\text{out},1}$, $t_{\text{out},2}$,... are obtained quite easily from the history array. For each output station t_{out}, the codes continue the integration until the first mesh point n for which $t_n \geq t_{\text{out}}$ and then compute the solution at t_{out} by a $q'_{n+1}{}^{\text{th}}$-order Taylor series expansion about t_n:

$$Y(t_{\text{out}}) = \sum_{j=0}^{q'_{n+1}} \frac{(t_{\text{out}} - t_n)^j}{j!} Y_n^{(j)} = \sum_{j=0}^{q'_{n+1}} \left(\frac{t_{\text{out}} - t_n}{h'_{n+1}}\right)^j z_n(j) \tag{A33}$$

where q'_{n+1} and h'_{n+1} are, respectively, the method order and step size to be attempted on the next step.

Both codes start the integration with a single-step, first-order method because information is available at only the initial point, t_0. The Nordsieck history matrix $\mathbf{z}_0$ at t_0 is constructed from the initial conditions $y(t_0)$ and the ODE's as follows:

$$z_0(0) \equiv Y_0 = y(t_0); \qquad z_0(1) \equiv h_0\dot{Y}_0 = h_0 f(Y_0)$$

where h_0, the step size to be attempted on the first step, has to be supplied by the user to EPISODE. In LSODE, however, h_0 is an optional input variable and is computed by the code, unless the user has specified a value for it.

GCKP84

GCKP84 is a general-purpose chemical kinetics code designed to solve a wide variety of problems (ref. 15). It uses the integration technique developed by Zeleznik and McBride (ref. 18). As implemented in GCKP84 the integration algorithm is an extensively modified version of the GEAR package (ref. 19), which is similar to LSODE. In particular, GEAR includes the two linear multistep methods discussed previously. The methods are based on a constant step length, and the method coefficients $\{l_{,n}\}$ (eq. (A11)) have the same values as in LSODE. Hence, $l_n(q)$ is a function of only the current method order q, satisfies $l_{0,n}(q) = \beta_{0,n}$ (see eq. (A7)) and $l_{1,n}(q) = 1$, and has to be recomputed only when the method order is changed. GCKP84 uses the same two linear multistep methods but the maximum method order is different: 11 for the implicit Adams method and 8 for the BDF method. The methods are also implemented differently as discussed below. For reasons given previously we restrict discussion to the BDF method (eq. (A7)).

As in EPISODE and LSODE, GCKP84 maintains the solution history in the form of the Nordsieck history array, $\mathbf{z}$. The array $\mathbf{z}_n$ at the current time t_n is obtained by using a

predictor-corrector process. The prediction step is performed in two stages. First, an initial estimate for $\mathbf{z}_n^{[0]}$ is computed via equation (A9); that is, the result of the prediction step used in LSODE and EPISODE (and GEAR) serves only as an initial estimate for $\mathbf{z}_n^{[0]}$ in GCKP84. Second, the above result is modified by means of an expression similar to equation (A10), as follows: The difference $(z_n(1) - z_n^{[0]}(1))$ (equal to e_n in GEAR because $l_1(q) = 1$, eq. (A10)), that is,

$$e_n = h_n\dot{Y}_n - h_n\dot{Y}_n^{[0]} \tag{A34}$$

may be regarded as the error in the q^{th}-order predictor relative to the converged array $\mathbf{z}_n$. Equation (A10) gives the history matrix $\mathbf{z}_n$ by adding the remainder term associated with using the q^{th}-order predictor. Of course, since e_n can be computed only after the converged solution is produced at t_n, the above procedure cannot be used. However, since e_{n-1} is available, it can be used to improve the initial estimate given by equation (A9). However, for additional accuracy improvement, GCKP84 uses the quantity E obtained by accumulating the errors $\{e_n\}$ (see eq. (A35)). The quantity E_n may be regarded as the estimated global error in $h_n\dot{Y}_n^{[0]}$. Since E_n is not known at the start of the step, GCKP84 uses E_{n-1} to improve the estimate given by equation (A9) as follows:

$$\mathbf{z}_n^{[0]} \leftarrow \mathbf{z}_n^{[0]} + \left(\frac{h_n}{h_E}\right)^{q+1} E_{n-1} l_n(q)$$

where h_E, which is normally equal to h_{n-1}, is the step size that E_{n-1} is based on and the term $(h_n/h_E)^{q+1}$ accounts for this fact: the exponent $q + 1$ arises because the current order is q and the local error varies as h^{q+1} (see eq. (A17)). On the first step, E_{n-1} (equal to E_0) is set equal to zero because $\dot{Y}_0$ (equal to $f(y(t_0))$) is known exactly.

After the prediction process is performed, the code checks the $\{Y_{i,n}^{[0]}\}$ for negative values. Because it is physically impossible for species concentrations, temperature, density, or velocity to be less than zero, the results of the predictor step are rejected if any $Y_{i,n}^{[0]} < 0$. Also, for each variable i for which the above condition is obtained, $E_{i,n-1}$ is reset to zero if it is less than zero. The step length is then reduced by a factor of two and a new $\mathbf{z}_n^{[0]}$ is generated. The above procedure is repeated until either all predicted solution components are nonnegative or the step length is reduced below a minimum value, $h_{\min}$, in which case an error exit is made.

To correct the initial estimate GCKP84 includes a variety of iteration techniques. For reasons given previously the discussion is restricted to the Newton-Raphson method. The procedure used to generate the improved estimates $Y_n^{[m]}$ $(m = 1,2,\ldots)$ is exactly the same as that described for LSODE: solve equation (A12). The iteration matrix $\mathbf{P}$ (eq. (A13)) is only accurate enough to achieve convergence, but the same $\mathbf{P}$ is used for a maximum number of 20 steps. At each iteration the approximation $h_n\dot{Y}_n^{[m]}$ to $h_n f_n$ is computed by using equation (A8). If any $Y_{i,n}^{[m]} < 0$, the iteration is abandoned. The step length is then reduced by a factor of two, and the step is retried.

Convergence of the estimates is said to occur if any of the following three tests, which are applied in the order they are given here, is satisfied. The first test involves the magnitude of the successive differences $(\dot{Y}_n^{[m]} - \dot{Y}_n^{[m-1]})$:

$$\delta_m \equiv \frac{1}{N}\left(\sum_{i=1}^{N}\left(\frac{\dot{Y}_{i,n}^{[m]} - \dot{Y}_{i,n}^{[m-1]}}{Y_{\max_i}}\right)^2\right)^{1/2} \overset{?}{\leq} 10^{-2}$$

where $Y_{\max,i}$ is given by the expression used in EPISODE for semirelative error control (eq. (A14). The second test is based on the size of the current estimate for e_n relative to the size of the current estimate for E_n (see eqs. (A34) and (A35)):

$$\left(\frac{1}{N}\sum_{i=1}^{N}\left[\frac{h_n\dot{Y}_{i,n}^{[m]} - h_n\dot{Y}_{i,n}^{[0]}}{h_n\dot{Y}_{i,n}^{[m]} - h_n\dot{Y}_{i,n}^{[0]} + \left(\frac{h_n}{h_E}\right)^{q+1} E_{i,n-1}}\right]^2\right)^{1/2} \overset{?}{\leq} 0.1$$

If for any i the denominator in the above summation is less than 10^{-50}, it is set equal to 10^{-2}. The third criterion is based on how rapidly the iteration is improving the solution and is given by

$$\left|\frac{\delta_m - \delta_{m-1}}{\delta_{m-1}}\right| \overset{?}{\leq} 10^{-3}$$

which can be applied only after two iterations. However, the third test is applied only after five iterations and that too only if $\delta_m \leq 5$.

If convergence is not achieved after four iterations, the iteration matrix $\mathbf{P}$ is updated at $y = Y_n^{[0]}$, and the correction process is retried. This procedure is repeated four times, after which, if the estimates have not converged, the step length is reduced by a factor of two and the step is retried. The same corrective actions are taken if on the fifth or subsequent iteration $\delta_m > \delta_{m-1}$. The above cycle of updating $\mathbf{P}$ every four iterations and then reducing h_n by a factor of two after four such updates is repeated until either convergence is obtained or the step length is reduced below $h_{\min}$, in which case an error exit is taken.

After corrector convergence the local error test is applied. This test is based on E_n, which is estimated by using

$$E_n = e_n + \left(\frac{h_n}{h_E}\right)^{q+1} E_{n-1} \tag{A35}$$

and can be written as

$$D_q \equiv \frac{\Delta_n}{\text{EPS } \mathfrak{I}_G(q)} \equiv \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{E_{i,n}}{Y_{\max_i}}\right)^2\right)^{1/2}}{\text{EPS } \mathfrak{I}_G(q)} \overset{?}{\leq} 5 \qquad \text{(A36)}$$

where EPS is the user-supplied local error tolerance and $\mathfrak{I}_G(q)$ (equal to $2/l_{0,n}(q)$) is the local error test coefficient for order q.

If the error test fails, the error vector E_{n-1} is updated by using equation (A35), and h_E is set equal to h_n because E_{n-1} is now based on h_n. The code GCKP84 then reduces the step size and/or the method order by one and retries the step. After three and more error test failures, the method order is reduced to one if it is greater than one, and the step length is set equal to $h_{\min}$. A new Nordsieck history matrix at t_{n-1} is constructed from Y_{n-1} and $f(Y_{n-1})$, E_{n-1} is set equal to zero, and the step is retried. After seven such failures or if h_n is reduced below $h_{\min}$, the integration is abandoned and an error exit is made.

If the error test passes, the step is accepted as successful, the Nordsieck history array $\mathbf{z}_n$ is updated by using equation (A10), E_n is computed by means of equation (A35), and h_E is set equal to h_n.

To increase the efficiency of the integration, the code periodically considers changing the method order to $q-1$ or $q+1$. Of course, if $q=1$, the choice $q-1$ is not considered. After an unsuccessful step or if either q is equal to the maximum method order, $q_{\max}$, or $D_q > 4\ t_G(q)$, the choice $q+1$ is rejected. For each method order q' the step size $h'(q')$ is computed from an estimate of the local error in a manner similar to the procedures used in EPISODE and LSODE (eqs. (A30) to (A32)). For each method order q' GCKP84 computes the step length ratio $r(q')$ as follows:

$$r(q') \equiv \frac{h'(q')}{h_n} = \frac{1}{\frac{4}{3}(D_{q'}D_z)^{\frac{1}{q'+1}} + 10^{-6}} \qquad \text{(A37)}$$

where

$$D_{q-1} = \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{z_{i,n}(q) - 0.5l_{q,n}(q)E_{i,n-1}}{Y_{\max_i}}\right)^2\right)^{1/2}}{\text{EPS } \mathfrak{I}_G(q-1)}$$

and

$$D_{q+1} = \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{E_{i,n} - \left(\frac{h_n}{h_E}\right)^{q+1}E_{i,n-1}}{Y_{\max_i}}\right)^2\right)^{1/2}}{\text{EPS } \mathfrak{I}_G(q+1)}$$

The local error test coefficients $\mathfrak{I}_G(q-1)$ and $\mathfrak{I}_G(q+1)$ for orders $q-1$ and $q+1$, respectively, are given by

$$\mathfrak{I}_G(q-1) = 2$$

and

$$\mathfrak{I}_G(q+1) = 2(q+2)/l_{0,n}(q)$$

The quantity D_z in equation (A37) is set equal to 10, unless $\Delta_{n-1} \geq 10^{-25}/\sqrt{N}$, in which case it is set equal to 0.1. If Δ_n is also greater than $10^{-25}/\sqrt{N}$, D_z is set equal to Δ_n/Δ_{n-1}. Finally, if D_z is less than $(0.25)^{q+1}$, it is set equal to this quantity.

The order corresponding to the maximum step length ratio $r = \max(r(q-1), r(q), r(q+1))$ and the step length ratio r are selected to be attempted on the next step if $r \geq 1.1$ after a successful step; otherwise, both changes are rejected. After a failed step, q is decreased if $r(q-1) > r(q)$; however, r is set equal to 1 if it is greater than 1. The following additional tests are performed on r before the step length h' (equal to rh_n) to be attempted next is selected:

$$r \leftarrow \min\left\{\frac{h_{\max}}{h_n},\ r_{\max},\ \max\left(r,\ \frac{h_{\min}}{h_n}\right)\right\}$$

The minimum, $h_{\min}$, and maximum, $h_{\max}$, step sizes are, respectively, set equal to h_0, the user-supplied value for the step length to be attempted on the first step, and $10(t_{\text{out}} - t_{\text{out,old}})$. On the first call t_{out} is set equal to t_0. The quantity $r_{\max}$ is set equal to 10. For the first step length increase following either a failed convergence test or a failed error test, it is set equal to two. However, after three or more error test failures, it is set equal to min $(10^4, h_n/h_{\min})$, thereby ensuring that the new step length equals $h_{\min}$. For the first step length increase for the problem, $r_{\max}$ is set equal to 10^4 if no convergence or error test failure has occurred.

Changes in method order and step length are attempted only after S successful steps with the same order and step length, where S is normally set equal to $q+2$. However, if an unsuccessful step occurs or if $D_q > 4\ \mathfrak{I}_G(q)$, the step length may be changed, and the method order may be reduced. Following a failed convergence or local error test, S is set equal to $q+2$. After three and more error test failures, S is set equal

to three. If method order and step size changes are rejected because $r < 1.1$, S is set equal to 10. Finally, the successful step counter is increased by one only if convergence is obtained in eight or fewer iterations.

If the step size and/or the method order are changed on the n^{th} step, $\mathbf{z}_n$ has to be modified. For the cases $q' = q - 1$ and $q' = q$, the modifications are made exactly as in LSODE (described previously). For the case $q' = q + 1$, $\mathbf{z}_n$ is first augmented by a column containing the vector $l_{q,n}(q)\, E_n/(q+1)$, which is approximately equal to $h_n^{q+1} Y_n^{(q+1)}/(q+1)!$, and then all $(q+2)$ columns are scaled by powers of h'/h_n to account for any change in the step size.

The solution values at the prescribed output times $t_{\text{out},1}$, $t_{\text{out},2}$, ... are obtained from the Nordsieck history array by using the Taylor series expansion method (eq. (A33)) described for EPISODE and LSODE. The same procedure used in these two codes to start the integration is used in GCKP84; the step size, h_0, to be attempted on the first step must be supplied by the user.

CHEMEQ

In this technique, developed by Young and Boris (ref. 10), the species rate equation (A1) is expressed as a difference between two positive-definite terms as follows:

$$\frac{dy_i}{dt} = f_i = \mathcal{P}_i - \mathcal{D}_i \qquad i = 1,\ldots,N_S \tag{A38}$$

where, for species i, the production rate $\mathcal{P}_i$ and the destruction rate $\mathcal{D}_i$ can be derived from equation (3):

$$\left.\begin{aligned} \mathcal{P}_i &= \rho^{-1} \sum_{j=1}^{N_R} (\nu_{ij}' R_{-j} + \nu_{ij}'' R_j) \\ \mathcal{D}_i &= \rho^{-1} \sum_{j=1}^{N_R} (\nu_{ij}' R_{-j} + \nu_{ij}'' R_{-j}) \end{aligned}\right\} \quad i = 1,\ldots,N_S \tag{A39}$$

When the temperature ODE (eq. (9)) is required (method B), it can be cast in a similar form by combining equations (9) and (A38)

$$\frac{dy_{N_S+1}}{dt} = \frac{dT}{dt} = \frac{-\sum_{k=1}^{N_S} f_k \hbar_k}{\sum_{k=1}^{N_S} y_k c_{p,k}} = \frac{-\sum_{k=1}^{N_S} (\mathcal{P}_k - \mathcal{D}_k)\hbar_k}{\sum_{k=1}^{N_S} y_k c_{p,k}}$$

$$= \mathcal{P}_{N_S+1} - \mathcal{D}_{N_S+1},$$

$$= \mathcal{P}_T - \mathcal{D}_T,$$

where

$$\mathcal{P}_T = \frac{\sum_{k=1}^{N_S} \mathcal{D}_k \hbar_k}{\sum_{k=1}^{N_S} y_k c_{p,k}}$$

and

$$\mathcal{D}_T = \frac{\sum_{k=1}^{N_S} \mathcal{P}_k \hbar_k}{\sum_{k=1}^{N_S} y_k c_{p,k}}$$

The objective of this decomposition is to enable factorization of y_i from $\mathcal{D}_i$

$$\mathcal{D}_i = \mathcal{L}_i y_i = y_i/\tau_i$$

where $\mathcal{L}_i$, the loss coefficient for species i, is obtained simply by dividing $\mathcal{D}_i$ by y_i (i.e., $\mathcal{L}_i = \mathcal{D}_i/y_i$). With this new notation, equation (A38) can be written as

$$\frac{dy_i}{dt} = \mathcal{P}_i - \mathcal{L}_i y_i = \mathcal{P}_i - y_i/\tau_i \tag{A40}$$

which, for constant $\mathcal{P}_i$ and $\mathcal{L}_i$, can be solved to give

$$y_i(t_n) = y_i(t_{n-1}+h_n) = \frac{\mathcal{P}_i}{\mathcal{L}_i} + y_i(t_{n-1}) - \frac{\mathcal{P}_i}{\mathcal{L}_i}\exp(-\mathcal{L}_i h_n)$$

Expressed in this way, it can be seen that $1/\mathcal{L}_i\ (= \tau_i)$ describes how quickly the variable y_i reaches its equilibrium value.

In advancing the solution from time t_{n-1} to time t_n, all of the equations are separated into two classes, stiff and nonstiff, according to the criterion

$$\frac{h_n}{\tau_{i,n-1}} \begin{cases} \geq 1 & \text{stiff} \\ < 1 & \text{nonstiff} \end{cases}$$

where $\tau_{i,n-1}$ denotes the value of τ_i at time t_{n-1}. The two types of equations are integrated by separate predictor-corrector schemes. For equations classified as nonstiff, the improved Euler method (with the Euler method as predictor and the modified Euler method (or trapezoidal rule) as corrector) is used. For equations classified as stiff, a simple, stable, asymptotic formula is used.

Nonstiff predictor:

$$Y_{i,n}^{[0]} = Y_{i,n-1} + h_n f_{i,n-1} \tag{A41a}$$

Stiff predictor:

$$Y_{i,n}^{[0]} = \frac{Y_{i,n-1}(2\tau_{i,n-1} - h_n) + 2h_n\,\tau_{i,n-1}\mathcal{P}_{i,n-1}}{2\tau_{i,n-1} + h_n} \tag{A41b}$$

Nonstiff corrector:

$$Y_{i,n}^{[m+1]} = Y_{i,n-1} + \frac{h_n}{2}\left[f_{i,n-1} + f_{i,n}^{[m]}\right] \tag{A42a}$$

Stiff corrector:

$$Y_{i,n}^{[m+1]} = \left\{\frac{h_n}{2}\left[\tau_{i,n}^{[m]} + \tau_{i,n-1}\right]\left[\mathcal{P}_{i,n}^{[m]} + \mathcal{P}_{i,n-1}\right] + Y_{i,n-1}\left[\tau_{i,n}^{[m]} + \tau_{i,n-1} - h_n\right]\right\} \Big/ \left[\tau_{i,n}^{[m]} + \tau_{i,n-1} + h_n\right] \tag{A42b}$$

In equations (A41) and (A42), $m + 1$ is the current iteration number. The zeroth estimate is the result of the predictor step. Also, $f_{i,n}^{[m]} = f_i(\{Y_{k,n}^{[m]}\})$. Convergence is ascertained by comparing $Y_{i,n}^{[m+1]}$ with $Y_{i,n}^{[m]}$ for all N components using the relative error criterion

$$\sigma \equiv \max_i \left[\frac{|Y_{i,n}^{[m+1]} - Y_{i,n}^{[m]}|}{\min|Y_{i,n}^{[m]},\ Y_{i,n}^{[m+1]}|}\right] \overset{?}{\leq} \text{EPS} \tag{A43}$$

To avoid numerical difficulties with the use of equation (A43), each estimate is constrained by a minimum value. In the present work, a variable that is less than 10^{-20} is set equal to 10^{-20}. Thus, for a species with decaying concentration, convergence is obtained trivially once $Y_i \leq 10^{-20}$, and its equation is decoupled from the equation set.

If convergence is not achieved after ITMAX iterations, the step length is halved and the step repeated. In this study, a value of ITMAX = 5 was used because it minimized the execution time for both test problems (ref. 2). If the corrector converges after M iterations ($M \leq$ ITMAX), the step is accepted as successful, and the solution is updated

$$Y_{i,n} = Y_{i,n}^{[M]} \qquad i = 1,\ldots,N.$$

No attempt is made either to estimate or control the local truncation error.

After each step n the step size h'_{n+1} to be attempted on the next step is computed from the converged integration cycle as follows:

$$h'_{n+1} = h_n\left[\frac{1}{(\sigma/\text{EPS})^{1/2}} + 0.005\right]$$

The step size, h_0, to be attempted on the first step is determined such that none of the variables will change by more than a prescribed amount. The formula used for h_0 is

$$h_0 = \text{EPS}\min_i\left[\frac{y_i(t_0)}{f_i(t_0)},\ \text{or}\ \frac{1}{\mathfrak{L}_{i,0}} \quad \text{if } y_i(t_0) \leq 10^{-20}\right]$$

The solution at each output station t_{out} was computed by linear interpolation between the computed approximations at t_{n-1} and t_n, where t_n is the first mesh point that is $\geq t_{\text{out}}$:

$$Y(t_{\text{out}}) = Y_{n-1} + \frac{t_{\text{out}} - t_{n-1}}{h_n}(Y_n - Y_{n-1})$$

CREK1D

In CREK1D, attention is paid to the distinguishing physical and computational characteristics of the induction, heat release, and equilibration regimes (refs. 11 to 14). This code consists of two algorithms developed for the two distinctly different regimes: (a) induction and early heat release, when the ODE's are dominated by positive time constants and (b) late heat release and equilibration, when the ODE's are more stable (ref. 2). Both algorithms are based on an exponentially fitted trapezoidal rule, but they use different iterative methods for convergence.

The code CREK1D solves a mixed differential-algebraic system of equations: ODE's for the species mole numbers and the algebraic enthalpy conservation equation (8) for the temperature. The ODE's and algebraic equation are solved simultaneously; however, in the following discussion the variables y and Y refer only to the species mole numbers.

The solution method used for the species ODE's is a generalized, tunable, single-step, implicit procedure:

$$Y_{i,n} = Y_{i,n-1} + h_n[U_{i,n}f_{i,n} + (1 - U_{i,n})f_{i,n-1}] \qquad i = 1,\ldots,N_S \tag{A44}$$

where $U_{i,n}$ is a degree-of-implicitness factor. This parameter is obtained by "exponentially fitting" it to a locally exact solution of equation (A1) as follows: The species rate expression f_i is expressed in a locally linearized form such that

$$f_i = f_{i,n-1} + \theta_{i,n}(Y_i - Y_{i,n-1}) \qquad i = 1,\ldots,N_S \tag{A45}$$

where the choice of $\theta_{i,n}$, a suitable linearization constant, is discussed shortly. Equation (A45) assumes that in the interval $[t_{n-1}, t_n]$ (i.e., locally) each species mole number varies exponentially. Integration of this equation gives

$$Y_{i,n} = Y_{i,n-1} + h_n f_{i,n-1} \frac{\exp(h_n\theta_{i,n}) - 1}{h_n\theta_{i,n}} \qquad \text{(A46)}$$

To exponentially fit $U_{i,n}$ we first replace $f_{i,n}$ in equation (A44) with the expression obtained from equation (A45):

$$f_{i,n} = f_{i,n-1} + \theta_{i,n}(Y_{i,n} - Y_{i,n-1}) \qquad \text{(A47)}$$

and then equate the resulting expression for $Y_{i,n}$ with equation (A46). These operations give

$$U_{i,n} = \frac{1}{h_n\theta_{i,n}} + \frac{1}{1 - \exp(h_n\theta_{i,n})} \qquad i = 1,\ldots,N_S \qquad \text{(A48)}$$

In order to maintain absolute A-stability of equation (A48) (i.e., to keep errors introduced into the numerical solution at any one step bounded as h_n is increased indefinitely), $U_{i,n}$ must be restricted to the interval (0.5,1.0). For values of $\theta_{i,n} > 0$, equation (A48) gives $U_{i,n} < 0.5$. CREK1D resolves this problem by setting $\theta_{i,n} = 0$ whenever it is greater than zero. This value of $\theta_{i,n}$ gives $U_{i,n} = 0.5$, so that equation (A44) defaults to the second-order-accurate trapezoidal rule. However, for $\theta_{i,n} < 0$, equations (A44) and (A48) together are equivalent to the locally exact or exponential solution, which has an equivalent polynomial accuracy of order six to eight (ref. 11). Thus, equations (A44) and (A48), with the constraint $(0.5 \le U_{i,n} < 1)$, constitute an exponentially fitted trapezoidal rule, a method which is A-stable and has a polynomial-order accuracy of at least two and as great as six to eight.

The linearization constants $\{\theta_{i,n}\}$ are obtained in one of two ways. In the first, called functional linearization (see refs. 11 to 14), equation (A47) is solved explicitly for $\theta_{i,n}$ to give

$$\theta_{i,n} = \frac{f_{i,n} - f_{i,n-1}}{Y_{i,n} - Y_{i,n-1}} \equiv Z_{i,n} \qquad \text{(A49)}$$

In the second approach, called formal linearization (refs. 11 to 14), the net formation rate of each species is expressed as a difference between two positive-definite terms, as described in the previous section (see eqs. (A38) and (A40)). Comparing equations (A47) and (A40) gives

$$\theta_{i,n} = -\mathfrak{L}_{i,n-1} \qquad \text{(A50)}$$

for this procedure.

At each integration step, equation (A44) must be solved for $Y_{i,n}$. The solution is accomplished by Newton-Raphson (NR) iteration in regime b and Jacobi-Newton (JN) iteration (ref. 29) in regime a.

A Newton-Raphson functional $F_{i,n}^{[m]}$ $(i = 1,\ldots,N_S)$ for each species mole number is defined from equation (A44) by

$$F_{i,n}^{[m]} = \frac{Y_{i,n}^{[m]} - Y_{i,n-1}}{h_n U_{i,n}} - \left(\frac{1 - U_{i,n}}{U_{i,n}}\right) f_{i,n-1} - f_{i,n}^{[m]} \qquad \text{(A51)}$$

for $i = 1,\ldots N_S$. For temperature the functional $F_{T,n}^{[m]}$ is defined from the enthalpy conservation equation (8) as

$$F_{T,n}^{[m]} = \sum_{k=1}^{N_S} Y_{k,n}^{[m]}\, \hbar_k(T_n^{[m]}) - H_0(T_0) \qquad \text{(A52)}$$

where m is the iteration number, $T_n^{[m]}$ is the m^{th}-approximation to the exact value $T(t_n)$, $\hbar_k(T_n^{[m]})$ is the molar-specific enthalpy of species k at temperature $T_n^{[m]}$, and $H_0(T_0)$ is the initial mixture mass-specific enthalpy at the initial temperature T_0.

Newton-Raphson corrector equations with log variable corrections (for self-scaling of the widely varying mole numbers) are given by

$$\sum_{k=1}^{N_S} \frac{\partial F_{i,n}^{[m]}}{\partial \ln Y_{k,n}^{[m]}} \Delta\ln Y_{k,n}^{[m+1]} + \frac{\partial F_{i,n}^{[m]}}{\partial \ln T_n^{[m]}} \Delta\ln T_n^{[m+1]} = -F_{i,n}^{[m]} \qquad \text{(A53)}$$

for i = $1,\ldots,N_S$.

$$\sum_{k=1}^{N_S} \frac{\partial F_{T,n}^{[m]}}{\partial \ln Y_{k,n}^{[m]}} \Delta\ln Y_{k,n}^{[m+1]} + \frac{\partial F_{T,n}^{[m]}}{\partial \ln T_n^{[m]}} \Delta\ln T_n^{[m+1]} = -F_{T,n}^{[m]} \qquad \text{(A54)}$$

where

$$\Delta\ln Y_{i,n}^{[m+1]} = \ln Y_{i,n}^{[m+1]} - \ln Y_{i,n}^{[m]} \qquad i = 1,\ldots,N_S$$

and

$$\Delta\ln T_n^{[m+1]} = \ln T_n^{[m+1]} - \ln T_n^{[m]}$$

The partial derivatives in equation (A53) are obtained from equation (A51) and are as follows (with the step and iteration numbers suppressed for clarity):

$$\frac{\partial F_i}{\partial \ln Y_k} = Y_k\left(\frac{\delta_{ik}}{hU_i} - \frac{\partial f_i}{\partial Y_k}\right) \qquad \text{(A55)}$$

$$\frac{\partial F_i}{\partial \ln T} = -T\frac{\partial f_i}{\partial T}$$

where δ_{ik}, the Kronecker symbol, is

$$\delta_{ik} \begin{cases} = 0 & i \neq k \\ = 1 & i = k \end{cases}$$

and $\partial f_i/\partial Y_k$ and $\partial f_i/\partial T$ can be derived from equations (3) to (8), and (10). In evaluating $\partial f_i/\partial Y_k$, the partial derivatives with respect to σ_m are assumed to be negligible compared with the other terms. The required partial derivatives are then given by

$$\frac{\partial f_i}{\partial Y_k} = -(\rho Y_k)^{-1} \sum_{j=1}^{N_R} (\nu'_{ij} - \nu''_{ij})(\nu'_{kj} R_j - \nu''_{kj} R_{-j}) \qquad \text{(A56)}$$

$$\frac{\partial f_i}{\partial T} = \frac{f_i}{T} - \frac{1}{\rho T} \sum_{j=1}^{N_R} (\nu'_{ij} - \nu''_{ij}) \left[R_j \left(N_j + \frac{T_j}{T} - n'_j \right) - R_{-j} \left(N_{-j} + \frac{T_{-j}}{T} - n''_j \right) \right]$$

where

$$n'_j = \sum_{i=1}^{N_S} \nu'_{ij} \qquad n''_j = \sum_{i=1}^{N_S} \nu''_{ij}$$

The partial derivatives of F_T are obtained by differentiating equation (A52) and are as follows:

$$\frac{\partial F_T}{\partial \ln Y_k} = Y_k h_k,$$

$$\frac{\partial F_T}{\partial \ln T} = \sum_{i=1}^{N_S} Y_k c_{p,k} T, \qquad \text{(A57)}$$

where, again, the step and iteration numbers have been suppressed. The $N_S + 1$ equations (A53) and (A54) are solved simultaneously by LU decomposition and back-substitution (ref. 27). The resulting log variable corrections are used to update the current estimates $\{Y_{i,n}^{[m+1]}\}$ and $T_n^{[m+1]}$ by the approximate equations

$$\left.\begin{aligned} Y_{i,n}^{[m+1]} &= Y_{i,n}^{[m]} \left(1 + \Delta \ln Y_{i,n}^{[m+1]}\right) \qquad i = 1,\ldots,N_S \\ T_n^{[m+1]} &= T_n^{[m]} \left(1 + \Delta \ln T_n^{[m+1]}\right) \end{aligned}\right\} \qquad \text{(A58)}$$

The solution procedure does not use a predictor; instead, the converged results $\{Y_{i,n-1}\}$ and T_{n-1} from the previous step are used to start the iteration.

The JN iteration technique can be derived from the NR iteration procedure by neglecting the off-diagonal elements of the Jacobian matrix for the mixed differential-algebraic system of equations. With this simplification, equations (A53) to (A55) reduce to

$$\frac{\partial F_{i,n}^{[m]}}{\partial \ln Y_{i,n}^{[m]}} \Delta \ln Y_{i,n}^{[m+1]} = -F_{i,n}^{[m]} \qquad i = 1,\ldots,N_S \qquad \text{(A59)}$$

$$\left.\begin{aligned} \frac{\partial F_{T,n}^{[m]}}{\partial \ln T_n^{[m]}} \Delta \ln T_n^{[m+1]} &= -F_{T,n}^{[m]} \\ \frac{\partial F_{i,n}^{[m]}}{\partial \ln Y_{i,n}^{[m]}} &= \frac{Y_{i,n}^{[m]}}{h_n U_{i,n}} - Y_{i,n}^{[m]} \frac{\partial f_{i,n}^{[m]}}{\partial Y_{i,n}^{[m]}} \end{aligned}\right\} \qquad \text{(A60)}$$

The iteration procedure is further simplified by approximating $\partial f_i/\partial Y_i$, (eq. (A56)) as follows:

$$\frac{\partial f_i}{\partial Y_i} = -(\rho Y_i)^{-1} \sum_{j=1}^{N_R} (\nu'_{ij} R_j + \nu''_{ij} R_{-j})$$

which, when combined with equation (A38), gives

$$\frac{\partial f_i}{\partial Y_i} = -\frac{\mathfrak{D}_i}{Y_i}$$

With these simplifications equation (A59) can be solved explicitly for the iterative corrections

$$\Delta \ln Y_{i,n}^{[m+1]} = -\frac{F_{i,n}^{[m]}}{Y_{i,n}^{[m]}/h_n U_{i,n} + \mathfrak{D}_{i,n}^{[m]}} \qquad i = 1,\ldots,N_S \qquad \text{(A61)}$$

The temperature correction is obtained by substituting equation (A57) into equation (A60):

$$\Delta \ln T_n^{[m+1]} = -\frac{F_{T,n}^{[m]}}{T_n^{[m]} \sum_{k=1}^{N_S} Y_{k,n}^{[m]} c_{p,k}(T_n^{[m]})} \qquad \text{(A62)}$$

where $c_{p,k}(T_n^{[m]})$ is the constant-pressure molar specific heat of species k at temperature $T_n^{[m]}$. The current estimates are updated by using equation (A58).

To start this iteration process, the predicted values for the species mole numbers are given by equation (A46):

$$Y_{i,n}^{[0]} = Y_{i,n-1} + h_n f_{i,n-1} \frac{\exp(h_n \theta_{i,n}) - 1}{h_n \theta_{i,n}} \qquad i = 1,\ldots,N_S \qquad \text{(A46a)}$$

The predicted temperature is obtained by a single NR iteration of the enthalpy conservation equation (8)

$$T_n^{[0]} = T_{n-1} + \frac{H_0(T_0) - \sum_{k=1}^{N_S} Y_{k,n}^{[0]}\, h_k(T_{n-1})}{\sum_{k=1}^{N_S} Y_{k,n}^{[0]}\, c_{p,k}(T_{n-1})} \qquad \text{(A63)}$$

For both NR and JN iteration schemes the test for convergence of the estimates $\{Y_{i,n}^{[m]}\}$ is based on the magnitudes of the log variable corrections, and is given by

$$\delta_m \equiv \left(\frac{\sum_{i=1}^{N_S} (\Delta \ln Y_{i,n}^{[m]})^2}{N_S} \right)^{1/2} \overset{?}{\leq} \text{EPS} \qquad \text{(A64)}$$

This test is used only with variables whose magnitudes are greater than 10^{-20}; that is, the summation does not include species with mole numbers $\leq 10^{-20}$. At each iteration the estimated convergence rate, C_m, defined as

$$C_m \equiv \frac{\delta_m}{\delta_{m-1}}$$

is also computed. If convergence is not obtained after ITMAX iterations, where ITMAX is the user-supplied maximum number of corrector iterations to be attempted, or if $C_m > 0.8$, the step length is decreased. The new step length is calculated as follows:

$$h_n \leftarrow h_n \min\{0.5,\ \max(0.1,\ 0.5/C_m)\}$$

and the step is retried with the new step size. At least two iterations are required to define C_m; on the first iteration δ_{m-1} is set equal to 10. A value of ITMAX = 10 was used for both problems examined in this study.

If convergence is achieved in M iterations ($M \leq$ ITMAX), the step is accepted as successful, and the solution is updated:

$$Y_{i,n} = Y_{i,n}^{[M]} \qquad i = 1,\ldots,N_S$$

$$T_n = T_n^{[M]}$$

After corrector convergence, the step length, h_{iter}, that would produce a convergence rate in the range (0.4,0.5) is estimated as follows:

$$h_{\text{iter}} = h_n(0.4/C_m)^{1/2} \qquad C_m < 0.4$$

$$= h_n \qquad 0.4 \leq C_m \leq 0.5$$

$$= h_n(0.5/C_m)^{1/2} \qquad C_m > 0.5$$

If convergence occurs on the first iteration, C_m is set equal to 0.1.

At each step an average weighted local truncation error estimate, d_n, is computed by using the approximations

$$d_n = \frac{1}{6}\left(\frac{1}{N_S} \sum_{i=1}^{N_S} \left\{ \frac{Y_{i,n} - Y_{i,n}^{[0]}}{\max\,(Y_{i,n-1}, Y_{i,n})} \right\}^2 \right)^{1/2}$$

for the JN iteration, and

$$d_n = \frac{1}{3}\left(\frac{1}{N_S} \sum_{i=1}^{N_S} \left\{ \frac{Y_{i,n} - Y_{i,n}^{[1]}}{\max\,(Y_{i,n-1}, Y_{i,n})} \right\}^2 \right)^{1/2}$$

for the NR iteration. The above summations include only species whose mole numbers are greater than 10^{-20}. For both iteration techniques the step length, h_{accy}, that would exactly satisfy the user-specified local relative error tolerance, EPS, is calculated from

$$h_{\text{accy}} = h_n(EPS/d_n)^{1/3}$$

The step length h'_{n+1} to be attempted on the next step is taken to be

$$h'_{n+1} = \min(h_{\text{iter}},\ h_{\text{accy}},\ 10h_n) \qquad \text{(A65)}$$

However, if convergence difficulties forced a reduction in the step length on the current step, h'_{n+1} is restricted to

$$h'_{n+1} \leftarrow \min(h_n,\ h'_{n+1}) \qquad \text{(A66)}$$

to prevent a recurrence of the problem.

CREK1D automatically selects the linearization method and the iteration scheme to be used for solving equation (A44). During induction and heat release, when small step lengths are required for solution accuracy, the JN iteration is used to minimize computational work. During late heat release and equilibration, when the ODE's are more stable and larger step lengths can be used, the NR iteration is preferred since it has better convergence properties than the JN iteration. The regime identification test exploits the fact that during equilibration many reactions achieve a condition in which the forward and reverse rates are large but with vanishingly small differences (refs. 13 and 30). The actual test used at the beginning of each time step is

$$|f_i| \overset{?}{\leq} 10^{-3}\,(\mathcal{P}_i + \mathcal{D}_i) \qquad \text{(A67)}$$

where $\mathcal{P}_i$ and $\mathcal{D}_i$ are given by equation (A39). If any two species satisfy equation (A67), regime b is obtained, and the NR iteration is used for the step. If fewer than two species

satisfy equation (A67), regime a is obtained, and the JN iteration is used for the step. Once the NR iteration is selected for any one step, the above test is no longer applied, and the NR iteration is used for the rest of the problem.

Whenever the reaction rate for any species satisfies equation (A67), that species is considered to be in "quasi-steady state" and the "L-formulated" equation (A50) is used. For all other species the "Z-formulated" equation (A49) is used. To minimize computational work, the $\{Z_{i,n}\}$ are evaluated only once per step: at the beginning of the time step, using equation (A49). However, since $Y_{i,n}$ and $f_{i,n}$ are not known at the start of the step, the $\{Z_{i,n}\}$ are approximated using values from the previous step:

$$Z_{i,n} \simeq Z_{i,n-1} = \frac{f_{i,n-1} - f_{i,n-2}}{Y_{i,n-1} - Y_{i,n-2}}$$

CREK1D also includes an algorithm for filtering the initial conditions that may be ill posed. These ill-posed conditions may arise, for example, in multidimensional modeling because of the averaging of mole numbers over adjacent grid nodes. CREK1D therefore "filters" the initial conditions to provide physically meaningful initial mole numbers and net species production rates. For purposes of this filtering CREK1D uses the decomposition performed in CHEMEQ (eqs. (A38) and (A40)). On the first call to CREK1D it uses this formulation over one time step of length h_1, given by

$$h_1 = \frac{1}{\max_i \mathfrak{L}_{i,0}} \tag{A68}$$

The predictor-corrector algorithm uses equation (A46) (with $\theta_{i,1} = -\mathfrak{L}_{i,0}$) as the predictor

$$Y_{i,1}^{[0]} = Y_{i,0} + h_1 f_{i,0} \left\{ \frac{1 - \exp(-h_1 \mathfrak{L}_{i,0})}{h_1 \mathfrak{L}_{i,0}} \right\} \qquad i = 1,\ldots,N_S$$

An implicit Euler corrector is then iterated to convergence

$$Y_{i,1}^{[m+1]} = Y_{i,0} + h_1 f_{i,1}^{[m+1]}$$

In the above two equations, the subscript 1 indicates that this is the first step. Using equations (A38), (A40), and (A58), together with the approximations $\mathcal{P}_{i,1}^{[m+1]} = \mathcal{P}_{i,1}^{[m]}$ and $\mathfrak{L}_{i,1}^{[m+1]} = \mathfrak{L}_{i,1}^{[m]}$, the preceding corrector equation can be rewritten to provide the following expression for the log variable corrections $\{\Delta \ln Y_{i,1}^{[m+1]}\}$:

$$\Delta \ln Y_{i,1}^{[m+1]} = \frac{Y_{i,0} - Y_{i,1}^{[m]} + h_1 f_{i,1}^{[m]}}{Y_{i,1}^{[m]} + h_1 \mathfrak{D}_{i,1}^{[m]}} \qquad i = 1,\ldots,N_S \tag{A69}$$

Equation (A69) is iterated until converged; that is, the criterion given by equation (A64) is satisfied. If convergence is not obtained after 10 iterations or $C_m > 0.8$, the step length is halved, and the step retried. If convergence is obtained after M iterations ($M \leq 10$), the step is accepted as successful, the solution for the mole numbers is updated

$$Y_{i,1} = Y_{i,1}^{[M]} \qquad i = 1,\ldots,N_S$$

and the temperature T_1 is obtained by a single Newton-Raphson iteration

$$T_1 = T_0 + \frac{H_0(T_0) - \sum_{k=1}^{N_S} Y_{k,1} h_k(T_0)}{\sum_{k=1}^{N_S} Y_{k,1} c_{p,k}(T_0)}$$

The step size, h_2', to be attempted on the next step is determined from the maximum loss coefficient at t_1 by using an expression similar to equation (A68). For this step, the JN iteration (eqs. (A46a) and (A61) to (A63)) is used, with all $\theta_{i,n}$ set equal to zero, so that all $U_{i,n} = 0.5$ (see eq. (A48)). The predictor step (eq. (A46a)), therefore, reduces to the explicit Euler method, and the corrector (eq. (A44)), to the trapezoidal rule. For the next and subsequent steps the step size is adjusted according to equation (A65) or (A66), and the iteration procedure and linearization constants are selected as described previously. If NR iteration is used, the Jacobian matrix for the mixed differential-algebraic system of equations is updated at $y = Y_{n-1}$, $T = T_{n-1}$.

The solution values at the prescribed output times $t_{\text{out},1}$, $t_{\text{out},2}$,... are obtained by adjusting the step length so that the internal mesh points coincide with these times. Thus, the step size h_{n+1}' is given by

$$h_{n+1}' \leftarrow \min(h_{n+1}', t_{\text{out}} - t_n),$$

where t_{out} is the current value of the output time, and the results at t_{out} are generated by solving the governing equations. To continue the integration past each output time, the procedures described above for the second and subsequent steps are used.

To reduce the computational cost, the use of exponential functions is minimized by replacing them with rational function approximations. For example, the term $(e^x - 1)/x$ in equation (A46a) is evaluated by means of a (2,2) diagonal Padé approximation, $e_{(2,2)}^x$, for $\exp x$:

$$e_{(2,2)}^x = \frac{1 + \frac{x}{2} + \frac{x^2}{12}}{1 - \frac{x}{2} + \frac{x^2}{12}} \qquad x < 0$$

which gives

$$\frac{e^x - 1}{x} \cong \frac{1}{1 - x(1/2 - x/12)} \qquad x < 0$$

Similarly, the tuning factor $U_{i,n}$ (eq. (A48)) is evaluated by using the approximation

$$U_{i,n} \cong 1 - \frac{1}{2} \exp\left(\frac{x - |x|}{12}\right)$$

This equation requires six operations to evaluate and does not exhibit the singularity at $x = 0$ of the exact expression (eq. (A48)).

Although log-variable corrections are used in the code, evaluation of logarithms of the variables is not required. Also, the use of the approximations given by equations (A58) avoids the cost of computing the exponentials of the log-variable corrections to obtain the new estimates.

Another technique used in the code to reduce computational work is to locally linearize the expressions for the thermodynamic properties of the species and the rate coefficients. In particular, during the course of iterative convergence of the equations, the thermodynamic properties and rate coefficients are not reevaluated while the current temperature is within a local window $(T, T + \Delta T)$, where ΔT is specified by the user. Use of this strategy has been shown to reduce the computational work (refs. 2, 3, and 5).

Appendix B
Description of Temperature Calculation Method A

In this method, the temperature T_n at each discrete time t_n is computed from the solution for the species mole numbers returned by the integrator by using the algebraic enthalpy conservation equation

$$\sum_{i=1}^{N_S} \sigma_{i,n} h_i(T_n) = H_0 \tag{8}$$

Equation (8) is solved for the temperature by using the Newton-Raphson (NR) iteration technique (e.g., ref. 23). This equation is rewritten as

$$F(T_n) = \sum_{i=1}^{N_S} \sigma_{i,n} h_i(T_n) - H_0 \tag{B1}$$

so that solving equation (8) is equivalent to finding the zero of F. The quantity $F(T_n^{[m]})$ is the amount by which the mixture mass-specific enthalpy at the m^{th} approximation for T_n, $T_n^{[m]}$ ($m = 1,2,\ldots$), fails to satisfy equation (8). A new approximation, $T_n^{[m+1]}$, for the temperature is obtained from equation (B1) by locally linearizing F at $T_n^{[m]}$:

$$T_n^{[m+1]} = T_n^{[m]} - \frac{F(T_n^{[m]})}{(\partial F/\partial T)_{T=T_n^{[m]}}} = T_n^{[m]} - \frac{F(T_n^{[m]})}{\sum_{i=1}^{N_S} \sigma_{i,n} c_{p,i}(T_n^{[m]})}$$

The test for convergence of the estimates is based on the magnitude of the corrections $\delta T_n^{[m+1]}$ (equal to $T_n^{[m+1]} - T_n^{[m]}$) and is given by

$$\left| \frac{\delta T_n^{[m+1]}}{T_n^{[m]}} \right| \overset{?}{\leq} \text{ERMAX}$$

where the vertical bars denote absolute value and ERMAX is the local relative error tolerance. If convergence is not obtained after MAXITS iterations, where MAXITS is the user-supplied maximum number of corrector iterations to be attempted, an error exit is taken. If convergence is achieved in M iterations ($M \leq$ MAXITS), the solution $T_n^{[M]}$ is accepted as the temperature at t_n:

$$T_n = T_n^{[M]}$$

The NR iteration will converge if the initial guess (i.e., $T_n^{[0]}$), is sufficiently accurate (ref. 23). The present work did not utilize a predictor to generate $T_n^{[0]}$; instead, the most recently computed temperature was used to start the iteration. Now, the temperature was evaluated at the end of each integration step and whenever the species derivatives and Jacobian matrix were computed. Hence, the converged value obtained either at the end of the previous step or from the previous estimates for the mole numbers was used as the initial guess for the current temperature. For the very first temperature computation for the problem, the initial temperature, T_0, served as the predicted value. The above procedure was found to be satisfactory in that the iteration converged for all integration methods and EPS values used in this study. In addition, the converged temperature was not significantly different from that obtained by integrating the temperature differential equation (ref. 2).

References

1. Gear, C.W.: Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall, Englewood Cliffs, NJ, 1971.
2. Radhakrishnan, K.: Comparison of Numerical Techniques for Integration of Stiff Ordinary Differential Equations Arising in Combustion Chemistry. NASA TP-2372, 1984.
3. Radhakrishnan, K.: New Integration Techniques for Chemical Kinetic Rate Equations. I. Efficiency Comparison, Combust. Sci. Technol., vol. 46, no. 1-2, 1986, pp. 59-81.
4. Radhakrishnan, K.: A Comparison of the Efficiency of Numerical Methods for Integrating Chemical Kinetic Rate Equations. Computational Methods, K.L. Strange, ed., Chemical Propulsion Information Agency, CPIA-PUBL-401, 1984, pp. 69-82 (Also, NASA TM-83590).
5. Radhakrishnan, K.: Fast Algorithms for Combustion Kinetics Calculations: A Comparison. Combustion Fandamentals Research, NASA CP-2309, 1984, pp. 257-267.
6. Byrne, G.D.; and Hindmarsh, A.C.: A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations. ACM Trans. Math. Software, vol. 1, no. 1, 1975, pp. 71-96.
7. Hindmarsh, A.C.; and Byrne, G.D.: EPISODE: An Effective Package for the Integration of Systems of Ordinary Differential Equations. UCID-30112, Rev. 1, Lawrence Livermore Laboratory, 1977.
8. Hindmarsh, A.C.: LSODE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers. ACM SIGNUM Newsletter, vol. 15, no. 4, 1980, pp. 10-11.
9. Hindmarsh, A.C.: ODEPACK: A Systematized Collection of ODE Solvers. Scientific Computing, R.S. Stepleman, ed., North Holland Publishing Co., Amsterdam, 1983, pp. 55-64.
10. 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. Phys. Chem., vol. 81, no. 25, 1977, pp. 2424-2427.
11. Pratt, D.T.: CREK1D: A Computer Code for Transient, Gas-Phase Combustion Kinetics. Paper WSCI 83-21, Combustion Institute, Apr. 11-12, 1983.
12. Pratt, D.T.; and Radhakrishnan, K.: CREK1D: A Computer Code for Transient, Gas-Phase Combustion Kinetics. NASA TM-83806, 1984.
13. Pratt, D.T.: Fast Algorithms for Combustion Kinetics Calculations. Stiff Computation, R.C. Aiken, ed., Oxford University Press, New York, 1985, pp. 217-229.
14. Pratt, D.T.; and Radhakrishnan, K.: A Fast Algorithm for Calculation of Combustion Chemical Kinetics, Paper 8-A1, Combustion Institute, May 5-6, 1986.
15. Bittker, D.A.; and Scullin, V.J.: GCKP84—General Chemical Kinetics Code for Gas Phase Flow and Batch Processes Including Hat Transfer Effects. NASA TP-2320, 1984.
16. Pratt, G.L.: Gas Kinetics. Wiley, London, 1969.
17. Gear, C.W.: The Automatic Integration of Ordinary Differential Equations. Commun. ACM, vol. 14, no. 3, 1971, pp. 176-179.
18. Zeleznik, F.J.; and McBride, B.J.: Modeling the Internal Combustion Engine, NASA RP-1094, 1984.
19. Hindmarsh, A.C.: GEAR: Ordinary Differential Equation System Solver. UCID-30001, Rev. 3, Lawrence Livermore Laboratory, 1974.
20. Liniger, W.; and Willoughby, R.A.: Efficient Integration Methods for Stiff Systems of Ordinary Differential Equations. SIAM J. Numer. Anal., vol. 7, no. 1, 1970, pp. 47-66.
21. Brandon, D.M., Jr.: A New Single-Step Implicit Integration Algorithm with A-Stability and Improved Accuracy. Simulation, vol. 23, no. 1, 1974, pp. 17-29.
22. Babcock, P.D., Stutzman, L.F. and Brandon, D.M., Jr.: Improvements in a Single-Step Integration Algorithm. Simulation, vol. 33, no. 1, 1979, pp. 1-10
23. Forsythe, G.E.; Malcolm, M.A.; and Moler, C.B.: Computer Methods for Mathematical Computations. Prentice-Hall, Englewood Cliffs, NJ, 1977.
24. Shampine, L.F.; and Gordon, M.K.: Computer Solution of Ordinary Differential Equations. The Initial Value Problem, W.H. Freeman and Co., San Francisco, 1975.
25. Burcat, A.; and Radhakrishnan, K.: High Temperature Oxidation of Propene. Combust. Flame, vol. 60, no. 2, 1985, pp. 157-169.
26. Radhakrishnan, K.; and Burcat, A.: Kinetics of the Ignition of Fuels in Artificial Air Mixtures. II: Oxidation of Propyne. Combust. Sci. Technol., vol. 54, no. 1-6, 1987, pp. 85-101.
27. Forsythe, G.E.; and Moler, C.B.: Computer Solution of Linear Algebraic Systems. Prentice-Hall, Englewood Cliffs, NJ, 1967.
28. Hindmarsh, A.C.: Construction of Mathematical Software. Part III: The Control of Error in the GEAR Package for Ordinary Differential Equations. UCID-30050, Part 3, Lawrence Livermore Laboratory, 1972.
29. Ortega, J.M.; and Rheinbolt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
30. Pratt, D.T.; and Radhakrishnan, K.: Fast Algorithms for Combustion Kinetic Calculations. Paper WSCI 81-46, Combustion Institute, Oct. 20, 1981.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1993	Technical Paper

4. TITLE AND SUBTITLE

A Critical Analysis of the Accuracy of Several Numerical Techniques for Combustion Kinetic Rate Equations

5. FUNDING NUMBERS

WU-505-31-42

6. AUTHOR(S)

Krishnan Radhadrishnan

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

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E-5861

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

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TP-3315

11. SUPPLEMENTARY NOTES

Krishnan Radhakrishnan, National Research Council—NASA Research Associate at Lewis Research Center; presently at Sverdrup Technology, Inc., Lewis Research Center Group, 2001 Aerospace Parkway, Brook Park, Ohio 44142.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 07

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

A detailed analysis of the accuracy of several techniques recently developed for integrating stiff ordinary differential equations is presented. The techniques include two general-purpose codes EPISODE and LSODE developed for an arbitrary system of ordinary differential equations, and three specialized codes CHEMEQ, CREK1D and GCKP84 developed specifically to solve chemical kinetic rate equations. The accuracy study is made by application of these codes to two practical combustion kinetics problems. Both problems describe adiabatic, homogeneous, gas-phase chemical reactions at constant pressure, and include all three combustion regimes: induction, heat release and equilibration. To illustrate the error variation in the different combustion regimes the species are divided into three types, reactants, intermediates and products, and error versus time plots are presented for each species type and the temperature. These plots show that CHEMEQ is the most accurate code during induction and early heat release. During late heat release and equilibration, however, the other codes are more accurate. A single global quantity, a mean integrated root-mean-square error, that measures the average error incurred in solving the complete problem is used to compare the accuracy of the codes. Among the codes examined, LSODE is the most accurate for solving chemical kinetics problems. It is also the most efficient code, in the sense that it requires the least computational work to attain a specified accuracy level. An important finding is that use of the algebraic enthalpy conservation equation to compute the temperature can be more accurate and efficient than integrating the temperature differential equation.

14. SUBJECT TERMS

Stiffness; Reaction kinetics; Combustion chemistry; Numerical method; Accuracy; Efficiency

15. NUMBER OF PAGES

60

16. PRICE CODE

A04

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

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

NASA-Langley, 1993