

Integrating Combustion Kinetic Rate Equations by Selective Use of Stiff and Nonstiff Methods

Krishnan Radhakrishnan*
NASA Lewis Research Center, Cleveland, Ohio

The effect of switching between nonstiff and stiff methods on the efficiency of algorithms for integrating combustion kinetic rate equations is presented. Different integration methods are tested by application of the packaged code LSODE to four practical combustion kinetics problems. The problems describe adiabatic, homogeneous gas phase combustion reactions. It is shown that selective use of nonstiff and stiff methods in different regimes of a typical batch combustion problem is faster than the use of either method for the entire problem. The implications of this result to the future development of fast integration techniques for combustion kinetic rate equations are discussed.

Introduction

THE ordinary differential equations (ODE's) describing complex chemical reactions are characterized by widely different time constants. Although the differential equations are generally stable, standard numerical techniques such as the explicit Runge-Kutta and Adams methods are prohibitively expensive to use because of the severe steplength restriction imposed by the requirements for numerical stability. Such systems of differential equations are commonly referred to as "stiff" systems.¹⁻⁵

The problem of stiffness has been recognized for some time,⁶ and several techniques have been developed for stiff ODE's. At the present time, the packaged codes EPISODE⁷ and LSODE^{8,9} are the most extensively used routines for stiff ODE's. Among several codes examined in recent detailed studies,¹⁰⁻¹⁴ LSODE was found to be the fastest for solving chemical kinetic rate equations. However, it is recognized by combustion device modelers that LSODE is not fast enough for economical calculations of multidimensional reacting flow problems.¹⁵

The numerical solution of combustion kinetic rate equations is complicated by the existence of a narrow region ("heat release" zone) in which the species concentrations and temperature change rapidly, as illustrated in Fig. 1 for a typical batch reaction combustion problem. In the heat release regime, especially in the early part, many of the species and the temperature have positive time constants—an indication that the governing ODE's are unstable. Since small steplengths are required for solving unstable ODE's, the use of methods designed for stiff problems—designated herein as "stiff methods"—may be inefficient. During early heat release, explicit "nonstiff methods"—i.e., methods suitable for nonstiff problems—may be adequate. However, implicit methods are more accurate than explicit methods, which are therefore used only as predictors in predictor-corrector algorithms.¹⁵ The corrector equations are iterated until convergence is obtained. It is not clear which corrector

iteration technique is optimal in the nonstiff regime. Both simple (or functional)^{16,17} and Jacobi-Newton^{18,19} iteration techniques have been used because they avoid the expense associated with forming and inverting Jacobian matrices, which is required by Newton-Raphson iteration. However, much larger steplengths can be used with Newton-Raphson iteration. For unstable ODE's, this advantage may not be of much help, and it is therefore not apparent which technique is the most efficient.

During late heat release and equilibration, the governing ODE's are stable, so Newton-Raphson iteration is the optimal convergence method. In these regimes, especially during equilibration, the different species approach the equilibrium state at different rates and the ODE's are stiff—i.e., classical numerical techniques will require prohibitive amounts of computer time in these regimes. Here, stiff methods are better suited to solving the problem.

In developing an efficient algorithm to solve combustion kinetic rate equations, it is important to recognize and accommodate the widely different characteristics of the three regimes (induction, heat release, and equilibration) encountered in a typical combustion problem. Such a situation where the problem changes character occurs in other areas, and schemes have been proposed for automatic switching between stiff and nonstiff methods.¹⁶

In this paper, the nature of the ODE's arising in combustion chemistry is examined and the effect of switching between stiff and nonstiff methods on solution efficiency is investigated. The use of different corrector iteration techniques with nonstiff methods is also studied.

Governing Ordinary Differential Equations

The first-order ODE's describing the time rate of change of species i ($i=1, NS$) can be written as

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

$$n_i(t=0) = \text{given} \quad T(t=0) = \text{given} \quad (1)$$

where n_i is the mole number of species i , NS is the total number of distinct species in the gas mixture, T is the temperature, and f_i is the net rate of the formation of species i due to all forward and reverse reactions in which species i participates. A more detailed description of the governing ODE's is given in Ref. 12.

Presented as Paper 85-0237 at the AIAA 23rd Aerospace Sciences Meeting, Reno, NV, Jan. 14-17, 1985; received Feb. 4, 1985; revision received Feb. 23, 1987. Copyright © 1987 American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

*NRC-NASA Research Associate. Member AIAA.

The initial value problem is to solve the system of Eqs. (1) for the chemical composition and temperature at the end of a prescribed time interval, given the initial conditions and the reaction mechanism. All problems considered in the present study involve only adiabatic, homogeneous, gas phase chemical reactions. The problems are, however, of two types—constant pressure and constant density. The following conservation equations serve as algebraic constraints on the species rate equations.

Constant pressure:

$$\sum_{i=1}^{NS} n_i h_i = H_0 = \text{constant} \quad (2a)$$

Constant density:

$$\sum_{i=1}^{NS} n_i (h_i - RT) = U_0 = \text{constant} \quad (2b)$$

where h_i is the molar-specific enthalpy of species i , R is the universal gas constant, and H_0 and U_0 are the mass-specific enthalpy and internal energy respectively of the ideal gas mixture. Time differentiation of Eqs. (2a) and (2b) provides the following ODE's for the temperature.

Constant pressure:

$$\frac{dT}{dt} = - \sum_{i=1}^{NS} f_i h_i / \sum_{i=1}^{NS} n_i c_{pi} \quad (3a)$$

Constant density:

$$\frac{dT}{dt} = - \sum_{i=1}^{NS} f_i (h_i - RT) / \sum_{i=1}^{NS} n_i (c_{pi} - R) \quad (3b)$$

where c_{pi} is the constant-pressure molar specific heat of species i .

Methods and Iteration Techniques Examined

The packaged code LSODE^{8,9} was used for the present investigation because it contains both stiff and nonstiff methods, and switching between the two methods is relatively straightforward. The methods included in this package are a variable-step, variable-order implicit Adams method, suitable for nonstiff problems, and a variable-step, variable-order backward differentiation formula (BDF),²⁰ suitable for stiff problems. These methods are among the most efficient currently available for nonstiff and stiff problems respectively.^{17,21} Both techniques employ a standard explicit predictor formula—a Taylor series expansion using the method devised by Nordsieck²²—to provide an initial estimate of the solution. To correct this estimate, a range of iteration formulas is included in LSODE. The methods and corrector techniques attempted in this study are examined briefly; details are available in Refs. 7, 23, and 24.

The ODE's presented in the previous section can be generalized as follows:

$$\dot{y}_i \equiv \frac{dy_i}{dt} = f_i(y_k) \quad i, k = 1, N \quad (4a)$$

or, using vector notation,

$$\dot{\mathbf{y}} \equiv \frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}) \quad (4b)$$

where

$$\begin{aligned} y_i &= n_i, & i &= 1, NS \\ y_N &= T & N &= NS + 1 \end{aligned} \quad (5)$$

Bold face type represents a vector quantity.

The techniques used in this study are step-by-step methods. They compute the approximations \mathbf{y}_n ($\equiv \mathbf{y}_{i,n}$; $i = 1, N$) to the exact solution $\mathbf{y}(t_n)$ [$\equiv \mathbf{y}_i(t_n)$; $i = 1, N$] at discrete points in time t_n ($n = 1, 2, \dots$). Assuming that solutions \mathbf{y}_{n-1} , \mathbf{y}_{n-2}, \dots have been obtained at times t_{n-1} , t_{n-2}, \dots , the methods used in LSODE to advance the solution ($= \mathbf{y}_{i,n}$) to time t_n involve linear multistep formulas of the type

$$y_{i,n} = \sum_{j=1}^{K_1} \alpha_j y_{i,n-j} + h_n \sum_{j=0}^{K_2} \beta_j \dot{y}_{i,n-j}, \quad i = 1, N \quad (6)$$

where $h_n (= t_n - t_{n-1})$ is the size of the steplength to be attempted, $\dot{y}_{i,n}$ [$\equiv f_i(y_{k,n})$] is the approximation to the exact derivative $\dot{y}_i(t_n)$ [$\equiv f_i(y_k(t_n))$] and K_1 , K_2 , α_j , and β_j are associated with the formula selected to solve the problem over this time step. For the implicit Adams method of order q , $K_1 = 1$, $K_2 = q - 1$, and Eq. (6) becomes

$$y_{i,n} = y_{i,n-1} + h_n \sum_{j=0}^{q-1} \beta_j \dot{y}_{i,n-j}, \quad i = 1, N \quad (7)$$

For a BDF of order q , $K_1 = q$, $K_2 = 0$, and Eq. (6) becomes

$$y_{i,n} = \sum_{j=1}^q \alpha_j y_{i,n-j} + h_n \beta_0 \dot{y}_{i,n}, \quad i = 1, N \quad (8)$$

Equations (7) and (8) can be written in the general form

$$y_{i,n} = \Psi_{i,n} + h_n \beta_0 \dot{y}_{i,n} = \Psi_{i,n} + h_n \beta_0 f_i(y_{k,n}), \quad i = 1, N \quad (9)$$

where $\Psi_{i,n}$ contains previously computed information. In vector notation, Eq. (9) becomes

$$\mathbf{y}_n = \Psi_n + h_n \beta_0 \mathbf{f}(\mathbf{y}_n) \quad (10)$$

All of the different corrector iteration techniques used in the present study to solve Eq. (10) can be generalized by the recursive relation

$$(I - h_n \beta_0 J) [\mathbf{y}_n^{(m+1)} - \mathbf{y}_n^{(m)}] = \Psi_n + h_n \beta_0 \mathbf{f}[\mathbf{y}_n^{(m)}] - \mathbf{y}_n^{(m)} \quad (11)$$

where I is the identity matrix, the matrix J depends on the iteration method selected, and (m) and $(m+1)$ denote the iteration numbers. $\mathbf{y}_n^{(0)}$ is the result obtained by the predictor step.

The choice $J=0$, called successive substitution,⁴ simple or functional iteration, and Jacobi iteration,^{16,17} results in

$$\mathbf{y}_n^{(m+1)} = \Psi_n + h_n \beta_0 \mathbf{f}[\mathbf{y}_n^{(m)}] \quad (12)$$

Equation (12) is very simple to use, but this method converges only linearly.⁴ In addition, for successful convergence, the steplength h_n may be restricted to very small values.¹

Newton-Raphson (NR) iteration, on the other hand, converges quadratically and can use much larger steplengths than functional iteration.^{4,16} For this method, J is the Jacobian matrix and the elements J_{ik} ($i, k = 1, N$) are given by

$$J_{ik} = \frac{\partial f_i}{\partial y_k} \quad i, k = 1, N \quad (13)$$

The formation of this matrix and the solution of the system of Eqs. (11) can require much computing time. To reduce this computational work, the iteration matrix $(I - h_n \beta_0 J)$ is not updated at every iteration. For additional savings, it is updated only when the solution to Eq. (11) does not converge. Hence the iteration matrix is only accurate enough for the iterations to converge, and the same matrix may be used over several steps.

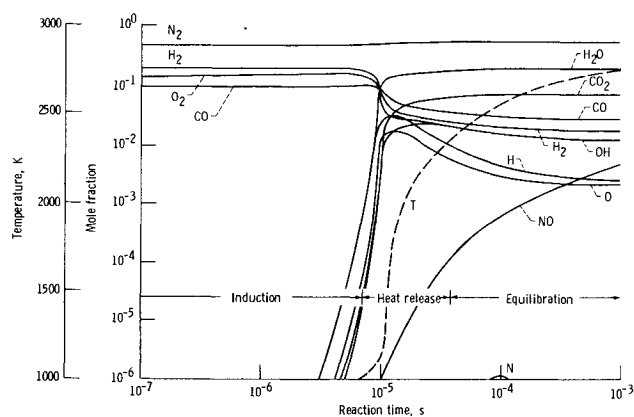


Fig. 1 Variation with reaction time (s) of the chemical species mole fractions and temperature (K) for test problem 1 (initial temperature = 1000 K, pressure = 10 atm).

Table 1 Summary of methods and corrector iteration techniques

Method flag, <i>MF</i>	Basic method	Iteration technique
10	—	Simple or functional
11	Variable-step, variable-order implicit Adams	Newton-Raphson with analytical Jacobian
13		Jacobi-Newton with finite difference generated Jacobian
21	Variable-step, variable-order, backward differentiation formula	Newton-Raphson with analytical Jacobian

The Jacobi-Newton (JN) iteration technique^{12,19} can be obtained from the NR iteration method by neglecting all off-diagonal elements of the Jacobian matrix. Hence for this technique

$$J_{ik} = 0; \quad k \neq i \quad = \frac{\partial f_i}{\partial y_k}; \quad k = i \quad (14)$$

This technique is as simple as functional iteration in the sense that no matrix inversion is involved. It also converges faster than functional iteration—better than linear but not quite quadratic.¹⁹

In summary, functional and JN iteration techniques require much less work per step than NR iteration but have to use smaller steplengths and converge at slower rates. For stable problems where the Jacobian changes slowly, NR iteration is clearly the optimal method. For unstable regimes, however, where rapidly changing solutions may require frequent updating of the Jacobian for successful convergence, simple or JN iteration may be more efficient; this may also be the case when very accurate numerical solutions are required. Because any change in the steplength alters the iteration matrix, it is not economical to consider small changes in the steplength with NR iteration. On the other hand, simple and JN iteration techniques can take advantage of even modest increases in the steplength. JN iteration requires a little more work per step than simple iteration, but it converges faster. It can also use steplengths at least as large as those used by simple iteration.¹⁸ The optimal corrector iteration technique therefore depends on the nature of the problem, the basic method used, and the accuracy required of the numerical solution.

In LSODE, both the basic method and the corrector iteration technique are selected via a method flag, *MF*. If NR

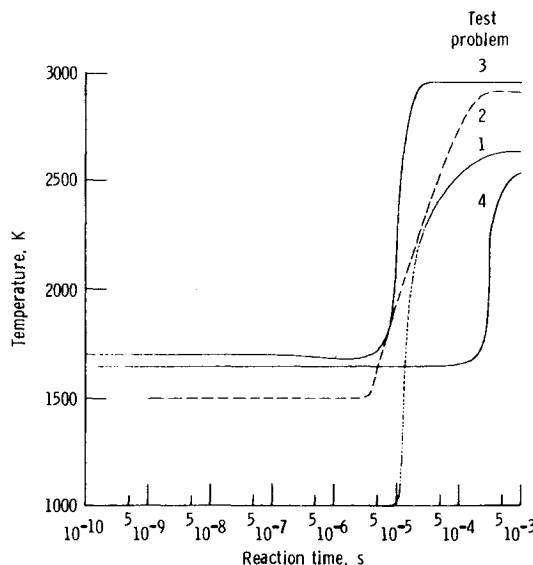


Fig. 2 Variation of the temperature (K) with reaction time (s) for test problems 1-4.

iteration is employed, either the user can provide analytical expressions for the elements of the Jacobian matrix or the code will estimate these elements by finite-difference approximations. For JN iteration, however, this option is not available, and the code uses internally-generated finite-difference approximations for the diagonal elements of the Jacobian matrix. For all results obtained with NR iteration, analytical Jacobians were used. The basic methods and iteration techniques employed in the present study are summarized in Table 1, together with the relevant values for *MF*.

Test Problems

Four practical combustion kinetics problems were used in the present study. All four cases described adiabatic, homogeneous, gas-phase, transient, batch combustion reactions. Test problem 1 described the ignition and subsequent combustion of a mixture of 33% carbon monoxide and 67% hydrogen with 100% theoretical air at an initial temperature of 1000 K and a pressure of 10 atm. It involved 12 reactions among 11 species. Test problem 2, involving 30 reactions among 15 species, described the ignition and subsequent combustion of a stoichiometric hydrogen-air mixture at 2 atm and 1500 K initial temperature. Both test cases 1 and 2 were at constant pressure and are discussed in more detail in Ref. 12. Test problem 3, taken from Burcat and Radhakrishnan,²⁵ described the ignition and subsequent combustion of a stoichiometric propene-oxygen-argon mixture at an initial temperature and pressure of 1700 K and 4 atm respectively. This constant density test case consisted of 113 reactions among 31 species. The reaction mechanism and rate constants were taken from Westbrook and Pitz.²⁶ Test case 4, taken from Bittker and Scullin,²⁷ was a lean methane-air ignition and combustion problem at a constant pressure of 1 atm and an initial temperature of 1645 K. This test problem involved 58 reactions among 24 species.

Figure 1 presents the variation with time of the chemical species mole fractions and temperature for test problem 1. The variation of the temperature with the reaction time for all four test cases is shown in Fig. 2. All four test problems were solved over a time period of 1 ms. This reaction period encompassed all three combustion regimes (induction, heat release, and equilibration) for test problems 1-3. Test case 4, however, included the first two regimes (induction and heat release) but only the beginning of equilibration.

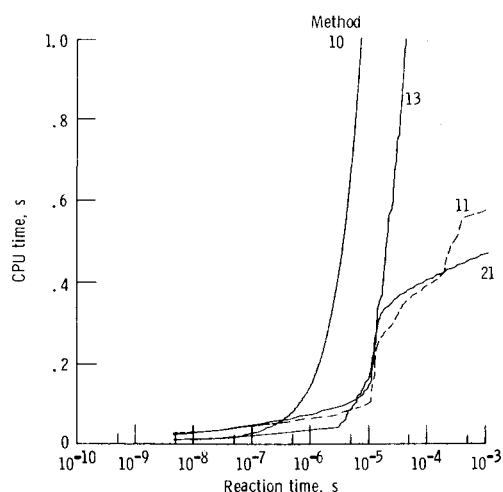


Fig. 3 Variation of the CPU time (s) with reaction time (s) for methods 10, 11, 13, and 21 (test problem 1, local relative error tolerance, $EPS = 10^{-2}$).

Results

In this section we present the effects on the computational work of using stiff and nonstiff methods in different regimes of a typical combustion kinetics problem. All results were obtained on the NASA Lewis Research Center's IBM 370/3033 computer using single-precision accuracy.

As illustrated in Fig. 1 and discussed in the introductory section, a typical combustion kinetics problem consists of three distinctly different regimes: induction, heat release, and equilibration. During induction and early heat release when many of the ODE's have positive time constants, small step-lengths must be used to insure solution accuracy.^{12,13} In these regimes, nonstiff methods may be more efficient.¹⁵ During late heat release and equilibration when the ODE's are more stable, much larger steplengths can be used, and NR iteration is the optimal convergence method.^{5,15,19} In these later regimes, especially during equilibration, the ODE's are stiff, so stiff methods are appropriate.

To investigate if it is more efficient to use a nonstiff method during induction and early heat release, the variation of the computer time with the reaction time was examined for all values of the method flag MF [$=10, 11, 13$, and 21 (see Table 1)] used in this study. Pure relative error control is appropriate for the problems employed in this study.¹² However, it could not be used because many of the mole numbers had zero initial value. A mixed relative and absolute error control was therefore used. Sufficiently small values were used for the species local absolute error tolerances to make the error control substantially relative for mole fractions greater than 0.1 ppm. For temperature, pure relative error control was used. To ensure that a comparison of computational work was made among comparably accurate methods, the same values for the absolute error tolerances were used with all methods and corrector iteration techniques. For clarity in presentation, methods corresponding to method flag $MF=10, 11, 13$, and 21 will hereafter be designated as methods 10, 11, 13, and 21, respectively.

Figures 3 and 4 present the variation of the computer (i.e., CPU) time in seconds with the reaction time for test problem 1 using values for the local relative error tolerance (EPS) of 10^{-2} and 10^{-5} respectively. For method 10 (implicit Adams with functional iteration) and $EPS = 10^{-2}$, the CPU time required up to the onset of heat release (reaction time $\sim 9 \mu s$, see Figs. 1 and 2) exceeded that required by method 21 (BDF with NR iteration using an analytical Jacobian) to solve the complete problem (Fig. 3). For $EPS = 10^{-5}$, however, the

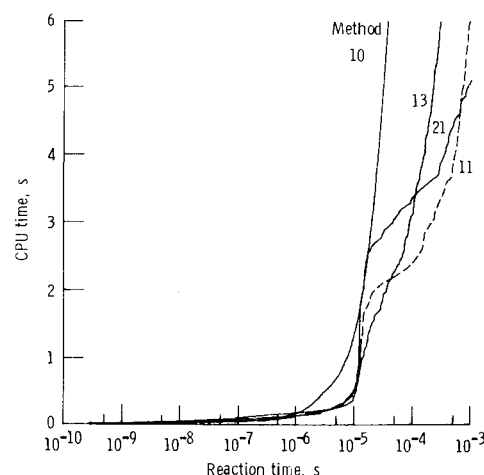


Fig. 4 Variation of the CPU time (s) with reaction time (s) for methods 10, 11, 13, and 21 (test problem 1, local relative error tolerance, $EPS = 10^{-5}$).

CPU times required during induction and early heat release were about the same for both methods (Fig. 4). For methods 11 (implicit Adams with NR iteration using an analytical Jacobian) and 13 (implicit Adams with JN iteration using internally generated approximations for the diagonal elements of the Jacobian matrix), the CPU times required during induction and early heat release compared favorably with, or were less than, those required by method 21. Note, however, that the CPU time required by method 21 for the complete problem was less than that required by all the nonstiff methods, indicating that the overall problem was stiff.

The results given in Figs. 3 and 4 show that JN and NR iteration techniques are more efficient than functional iteration in the nonstiff regime. These results also indicate that the use of a nonstiff method during induction and early heat release and a stiff method for the remainder of the problem would be more efficient than using either method for the complete problem. To examine the effects of such a switch, the following procedure was used. The code was run up to reaction time $t = t_{\text{switch}}$ with a nonstiff method. After every step successfully executed by the routine, the value of the time reached by the integrator was checked to ensure that it did not exceed t_{switch} . If the time exceeded t_{switch} , the method was switched to 21 and the problem was run to completion with the stiff method. Upon completion of the problem, the CPU time required to solve the problem was calculated. In addition, the following performance parameters that give an indication of the computational work required to solve the problem were noted: total number of steps ($NSTEP$), total number of functional (i.e., derivative) evaluations (NFE), and total number of Jacobian matrix evaluations (NJE).

Different values for t_{switch} were attempted, and the value resulting in the least CPU time to solve the problem was obtained by a trial and error process. Since the objective of the present investigation was solely to determine if switching methods resulted in efficiency increases and if so to identify the optimal iteration technique to be used in the nonstiff regime, no attempt was made to incorporate automatic method selection procedures.

Table 2 presents the minimal CPU times obtained for test problem 1 using the two-stage solution scheme previously outlined and different iteration techniques in the nonstiff regime. In this table, t_{switch} is the reaction time (in μs) up to which the program was run with the nonstiff method and the indicated iteration technique. For values of reaction time $t > t_{\text{switch}}$, the solution was obtained with the stiff method 21. Also given in Table 2 is the computational work required by method 21 to solve the complete problem. For method 10

Table 2 Summary of computational work required by two-region solution for test problem 1

Method	<i>EPS</i>	$t_{\text{switch}}, \mu\text{s}$	<i>NSTEP</i>	<i>NFE</i>	<i>NJE</i>	CPU, s
10/21	10^{-5}	17	1701	2801	51	3.87
11/21	10^{-2}	15	99	166	30	0.40
	10^{-3}	21	173	297	33	0.63
	10^{-4}	23	336	540	66	1.17
	10^{-5}	40	923	1675	139	3.33
13/21	10^{-2}	13	157	308	16, 45 ^a	0.49
	10^{-3}	14	244	459	14, 60	0.64
	10^{-4}	13	466	958	24, 135	1.29
	10^{-5}	15	956	1742	40, 146	2.52
21	10^{-2}	—	115	183	30	0.44
	10^{-3}	—	207	346	46	0.78
	10^{-4}	—	308	504	57	1.08
	10^{-5}	—	1263	2429	255	4.95

^aFor method 13, the first number is the total number of complete Jacobian matrix evaluations, and the second number is the total number of diagonal matrix approximations.

Table 3 Summary of computational work required by two-region solution for test problem 2

Method	<i>EPS</i>	$t_{\text{switch}}, \mu\text{s}$	<i>NSTEP</i>	<i>NFE</i>	<i>NJE</i>	CPU, s
11/21	10^{-2}	3	96	158	29	0.70
	10^{-3}	5	159	255	43	1.07
	10^{-4}	4.5	237	368	43	1.38
	10^{-5}	20	2846	4686	422	16.5
21	10^{-2}	—	100	163	29	0.71
	10^{-3}	—	157	243	36	1.03
	10^{-4}	—	295	471	63	1.87
	10^{-5}	—	3527	5705	579	20.5

and $EPS \geq 10^{-4}$, the CPU times required up to the onset of heat release exceeded those required by method 21 to solve the complete problem. Therefore no switching was attempted for these values of *EPS* with method 10. For $EPS = 10^{-5}$, however, the combination of methods 10 and 21 was about 20% faster than method 21 for the complete problem (Table 2). Note that fewer steps and functional evaluations were required by the stiff method, indicating that the average step length was smaller for method 10. However, the use of method 10 during induction and early heat release resulted in significantly fewer Jacobian evaluations. This was due to 1) not computing the Jacobian in the initial regime and 2) fewer Jacobian evaluations being required in the second regime because of the use of smaller step lengths.

The combinations of methods 11 and 21 and of methods 13 and 21 resulted in decreased CPU time (relative to method 21 for the complete problem) for most of the error tolerances (Table 2). Also, in all cases the combination of nonstiff and stiff methods was faster than using the nonstiff method for the complete problem. Note that the time at which methods had to be switched generally increased with decreasing *EPS*, i.e., increasing accuracy requirement. This implies that when *EPS* is decreased, accuracy requirements control the step size for a longer time. When accuracy requirements and not numerical solution stability requirements control the size of the step, the problem is not stiff.^{2,21} Hence, the time over which it was more efficient to use a nonstiff method increased with decreasing *EPS*.

The combination of methods 11 and 21 resulted in CPU time decreases ranging from negligibly small to over 30% for test problem 1 (Table 2). This switching process, i.e., the use of a nonstiff method during induction and early heat release and of a stiff method for the remainder of the problem, is not entirely satisfactory in that it does not always result in significant savings over the use of the stiff method 21 for the complete problem. Similar remarks apply to the use of

Table 4 Summary of computational work required by two-region solution for test problem 3

Method	<i>EPS</i>	$t_{\text{switch}}, \mu\text{s}$	<i>NSTEP</i>	<i>NFE</i>	<i>NJE</i>	CPU, s
11/21	10^{-2}	0.5	163	273	46	3.67
	10^{-3}	5.0	368	590	94	7.83
	10^{-4}	0.3	689	1133	212	15.8
	10^{-5}	0.03	1148	1794	200	20.9
21	10^{-2}	—	228	380	74	5.45
	10^{-3}	—	373	612	124	8.88
	10^{-4}	—	783	1355	273	19.5
	10^{-5}	—	1634	2706	449	35.9

method 13 in the initial regimes. Note that for method 13 *NJE* includes two types of Jacobian matrix evaluations—the first number if the total number of complete (i.e., analytical) Jacobian matrix evaluations, and the second number is the total number of diagonal matrix approximations (Table 2). One difficulty encountered with the use of method 13 was that it returned inaccurate solutions when relatively large values of *EPS* were used. This problem has been reported by others.^{28,29} It is not clear if this is caused by poor approximations for the diagonal elements or by an unreliable convergence test. Another difficulty encountered with this method was serious numerical instability for some test problems and values of *EPS*. Because of these problems with method 13, it was not attempted with the other three test cases.

For the other three test problems and most of the error tolerances used, the runs with method 10 required more CPU time until the onset of heat release than did method 21 for the complete problem, (e.g., Fig. 5). Hence, method 10 was also not attempted in the nonstiff regime for test problems 2–4.

Tables 3, 4, and 5 present the effects of switching between methods 11 and 21 for test problems 2, 3, and 4 respectively. For purposes of efficiency comparison, the computational work required by method 21 for the complete problem is also given in these tables. The results for test problem 2 (Table 3) were very similar to those obtained for test problem 1. The use of the two-region scheme resulted in efficiency increases for most of the error tolerances and, as *EPS* was decreased, the switching had to be performed at later times.

For test problem 3, however, no significant efficiency increases could be obtained by using the nonstiff method 11 during induction and early heat release and then switching to the stiff method 21. But significant efficiency increases could be obtained by switching before the onset of heat release (Table 4). Note that for $EPS = 10^{-5}$, switching from method 11 to method 21 at $t = 0.03 \mu\text{s}$ (for this problem heat release started at about $3 \mu\text{s}$, Fig. 2) resulted in a CPU time decrease of over 40%. For test problem 3, unlike test problems 1 and 2, the temperature dropped by a significant amount (~ 21 K) during induction. This decrease in temperature was diagnosed by the code as an indication of stiffness, especially when low values were used for *EPS*. Note the sharp increase in CPU time incurred by the nonstiff methods during induction (Fig. 5).

Test problem 4 was also quite different from test problems 1 and 2. Although the temperature drop during induction was not significant (less than 1 K), this problem was characterized by a fairly long ignition delay period (Fig. 2). In addition, when the temperature started to increase (at $t \sim 20 \mu\text{s}$), it did so gradually and not as rapidly as in the other problems. For example, at $t = 100 \mu\text{s}$, the temperature had risen by only about 10 K. Unlike the other three test problems, test problem 4 included only the beginning of the equilibration regime. A nonstiff method was therefore expected to be more efficient for most of the problem. However, the results given in Table 5 show that for increased

efficiency, switching had to be performed during induction. This indicates that for test problem 4 also it was more efficient to use a stiff method during induction, as illustrated in Fig. 6 for $EPS=10^{-4}$. Note the large increase in CPU time for method 11 between $t=1$ and $20 \mu\text{s}$. For method 21, the CPU time showed a large increase between approximately 300 and $350 \mu\text{s}$, corresponding to the rapid increase in the temperature between these times (Fig. 2). In this interval, method 11 was more efficient (Fig. 6) because accuracy requirements control the step size. The effect of using a nonstiff method in this interval was examined as follows for $EPS=10^{-4}$. The program was run with method 11 up to 2.5

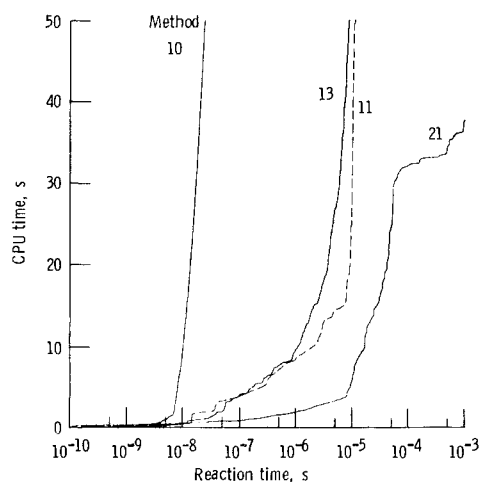


Fig. 5 Variation of the CPU time (s) with reaction time (s) for methods 10, 11, 13, and 21 (test problem 3, local relative error tolerance, $EPS=10^{-5}$).

Table 5 Summary of computational work required by two-region solution for test problem 4

Method	EPS	$t_{\text{switch}}, \mu\text{s}$	$NSTEP$	NFE	NJE	CPU, s
11/21	10^{-2}	25	157	264	45	2.24
	10^{-3}	12.5	302	521	80	4.14
	10^{-4}	2.5	673	1176	192	9.78
	10^{-5}	250	1530	2543	210	16.8
21	10^{-2}	—	198	326	64	2.93
	10^{-3}	—	490	860	172	7.57
	10^{-4}	—	723	1195	207	10.0
	10^{-5}	—	2000	3708	452	27.1

Table 6 Summary of computational work required by three-region solution for test problem 1

Method	EPS	$t_{\text{sw},1}, \mu\text{s}$	$t_{\text{sw},2}, \mu\text{s}$	$NSTEP$	NFE	NJE	CPU, s
21/10/21	10^{-2}	8.5	9.0	107	168	31	0.42
	10^{-3}	9.0	10.0	203	349	37	0.71
	10^{-4}	9.5	12.5	406	633	39	1.09
	10^{-5}	9.0	14.0	1076	1726	70	2.81
21/11/21	10^{-2}	8.5	15	104	157	30	0.38
	10^{-3}	8.5	22	167	263	34	0.58
	10^{-4}	9.0	55	278	435	43	0.90
	10^{-5}	9.0	65	866	1534	132	2.98
21/13/21	10^{-2}	8.0	13.0	136	231	23, 24 ^a	0.42
	10^{-3}	8.0	13.5	230	403	32, 41	0.70
	10^{-4}	9.0	13.5	297	499	32, 33	0.89
	10^{-5}	9.0	25	981	1691	58, 153	2.62

^aFor method 13, the first number is the total number of complete Jacobian matrix evaluations, and the second number is the total number of diagonal matrix approximations.

μs and between 300 and $350 \mu\text{s}$. At all other times method 21 was used. This resulted in a total CPU time requirement of 7.6 s—which was significantly faster than both the simple switch performed earlier (i.e., the two-stage solution scheme) and method 21 for the complete problem (Table 5).

The results previously discussed indicate that the induction regime is not necessarily nonstiff, so the use of a nonstiff method in this regime does not guarantee minimal computational work. In this regime, the use of either a stiff method or a combination of nonstiff and stiff methods may require the least computational work. To test this hypothesis, the following procedure was adopted. The program was run with the stiff method 21 until the onset of heat release and also during late heat release and equilibration. During early heat release, however, a nonstiff method was used.

Table 6 presents the minimal CPU time obtained for test problem 1 using the three-region solution scheme previously discussed; all iteration techniques were attempted during early heat release. In this table, $t_{\text{sw},1}$ and $t_{\text{sw},2}$ are the times at which the methods were switched from stiff to nonstiff and from nonstiff to stiff respectively. As EPS was increased, $t_{\text{sw},1}$ had to be decreased because heat release was predicted to start at an earlier time. As discussed previously, $t_{\text{sw},2}$ had to be increased with decreasing EPS . A comparison of Tables 2 and 6 shows that for almost all iteration techniques and error tolerances, the three-stage solution scheme was faster than both the two-stage solution scheme proposed earlier and the stiff method 21 for the complete problem. Note that the use of this combination of stiff and nonstiff methods has resulted in about a 50% reduction in the CPU time for $EPS=10^{-5}$ and method 13 during early heat release. Although the use of method 10 also resulted in efficiency increases, a very low value of EPS (10^{-5}) was required for significant reductions in the CPU time (Table 6). The use of such low values of EPS is wasteful, especially for multidimensional reacting flow calculations.³⁰ This indicates that either JN or NR iteration should be used during early heat release. For small values of EPS , JN iteration (method 13) is more efficient. But for large values of EPS , NR iteration (method 11) is superior (Table 6).

The results presented indicate that for the efficient solution of combustion kinetic rate equations, nonstiff methods should be used during early heat release. However, it is not clear if JN or NR iteration should be used in this regime. For large values of the local error tolerance, JN iteration resulted in significant errors. This could be due to the approximations for the Jacobian elements used in LSODE. No such problem was encountered with CREKID,^{12,14,19} which employs JN iteration but with an analytical Jacobian. This suggests that JN iteration with an analytical Jacobian should be attempted during early heat release. During late heat

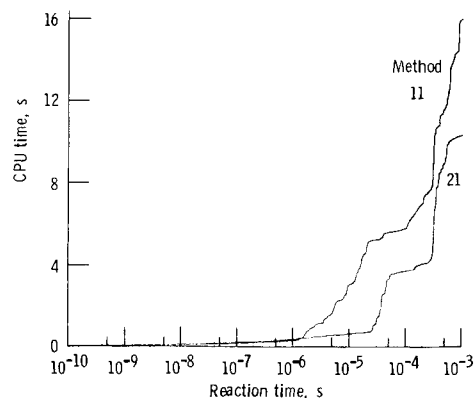


Fig. 6 Variation of the CPU time (s) with reaction time (s) for methods 11 and 21 (test problem 4, local relative error tolerance, $EPS=10^{-4}$).

release and equilibration, however, a stiff method should be used. During induction, either a stiff method or a combination of nonstiff and stiff methods appears to be the optimal choice.

These findings can be used to develop fast algorithms for solving combustion kinetic rate equations. However, the development of an automatic integration technique, i.e., one that uses the appropriate method in each regime, requires regime identification tests and reliable switching procedures. The optimal switching times are highly problem- and error tolerance-dependent, so generalized switching methods are not yet available. The trial and error procedure employed in the present work is useful when repetitive computations are made on the same problem or closely related problems, as, for example, calculations of combustion kinetics in homogeneous reacting flow. Until reliable switching procedures are developed, the use of stiff methods is recommended in all regimes of a typical batch combustion kinetics problem.

Conclusions

The nature of the ordinary differential equations arising in combustion chemistry was investigated by using combinations of stiff and nonstiff methods in different regimes of a typical batch reaction combustion problem. It was shown that the selective use of stiff and nonstiff methods in different regimes is faster than the use of either method for the complete problem. In particular, for optimal solution during induction, the use of either a stiff method or a combination of nonstiff and stiff methods is indicated. Stiff methods are also optimal during late heat release and equilibration. During early heat release, however, a nonstiff method should be employed. In this regime, both Newton-Raphson (NR) and Jacobi-Newton (JN) iteration techniques were superior to simple (or functional) iteration. However, it is not clear whether NR or JN is the optimal iteration technique. For large values of the local error tolerance, NR iteration was superior. The JN iteration technique included in the packaged code LSODE produced large errors and also resulted in solution instability when large values were specified for the local error tolerance. This may be the result of poor approximation for the Jacobian matrix. Further experimentation with an analytical Jacobian is necessary to determine the optimal iteration technique during early heat release.

Acknowledgments

This work was supported by the NASA Lewis Research Center, Cleveland, Ohio, through an NRC-NASA Research Associateship and in part through Grant NAG 3-294 and Contract NAS 3-24105.

References

- Lapidus, L. and Seinfeld, J. H., *Numerical Solution of Ordinary Differential Equations*, Academic Press, Inc., New York, 1971.
- Gelinas, R. J., "Stiff Systems of Kinetic Equations—A Practitioner's View," *Journal of Computational Physics*, Vol. 9, 1972, pp. 222-236.
- Lambert, J. D., *Computational Methods in Ordinary Differential Equations*, John Wiley and Sons Ltd., New York, 1973.
- Finlayson, B. A., *Nonlinear Analysis in Chemical Engineering*, McGraw Hill Inc., New York, 1980.
- Pratt, D. T. and Radhakrishnan, K., "Physical and Numerical Sources of Computational Inefficiency in the Integration of Chemical Kinetic Rate Equations: Etiology, Treatment and Prognosis," NASA TP-2590, 1986.
- Curtiss, C. F. and Hirschfelder, J. O., "Integration of Stiff Equations," *Proceedings of the National Academy of Science*, Vol. 38, 1952, pp. 235-243.
- 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.
- Hindmarsh, A. C., "LSODE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers," *SIGNUM Newsletter*, Vol. 15, 1980, pp. 10-11.
- Hindmarsh, A. C., "ODEPACK: A Systematized Collection of ODE Solvers," UCRL-88007, Lawrence Livermore Laboratory, 1982.
- Radhakrishnan, K., "A Comparison of the Efficiency of Numerical Methods for Integrating Chemical Kinetic Rate Equations," NASA TM-83590, 1984, *Computational Methods*, CPIA Publication 401, 1984, pp. 69-82.
- Radhakrishnan, K., "Fast Algorithms for Combustion Kinetics Calculations: A Comparison," NASA CP-2309, *Combustion Fundamentals Research*, 1984, pp. 257-267.
- Radhakrishnan, K., "Comparison of Numerical Techniques for Integration of Stiff Ordinary Differential Equations Arising in Combustion Chemistry," NASA TP-2372, 1984.
- Radhakrishnan, K., "New Integration Techniques for Chemical Kinetic Rate Equations. I. Efficiency Comparison," *Combustion Science and Technology*, Vol. 46, 1986, pp. 59-81.
- Radhakrishnan, K., "New Integration Techniques for Chemical Kinetic Rate Equations. II. Accuracy Comparison," NASA TM-86893, 1984, ASME Paper 85-GT-30, *Transactions of the ASME Journal of Engineering for Gas Turbines and Power*, Vol. 108, 1986, pp. 348-353.
- Pratt, D. T., "Exponentially-Fitted Methods for Integrating Stiff Systems of Ordinary Differential Equations: Applications to Homogeneous, Gas-Phase Chemical Kinetics," *Computational Methods*, CPIA Publication 401, 1984, pp. 53-67.
- Shampine, L. F., "Type-Insensitive ODE Codes Based on Implicit A-Stable Formulas," *Mathematics of Computation*, Vol. 36, 1981, pp. 499-510.
- Petzold, L., "Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations," *SIAM Journal of Scientific Statistical Computation*, Vol. 4, 1983, pp. 136-148.
- Shampine, L. F., "Type-Insensitive ODE Codes Based on Implicit A(α)-Stable Formulas," *Mathematics of Computation*, Vol. 39, 1982, pp. 109-123.
- Pratt, D. T. and Radhakrishnan, K., CREK1D: A Computer Code for Transient, Gas-Phase Combustion Kinetics," NASA TM-83806, 1984.
- Gear, C. W., "The Numerical Integration of Ordinary Differential Equations," *Mathematics of Computation*, Vol. 21, 1967, pp. 146-156.
- Shampine, L. F., "Stiffness and the Automatic Selection of ODE Codes," *Journal of Computational Physics*, Vol. 54, 1984, pp. 74-86.
- Nordsieck, A., "On the Numerical Integration of Ordinary Differential Equations," *Mathematics of Computation*, Vol. 16, 1962, pp. 22-49.
- Hindmarsh, A. C., "Linear Multistep Methods for Ordinary Differential Equations: Method Formulations, Stability, and the Methods of Nordsieck and Gear," UCRL-51186 Rev. 1, Lawrence Livermore Laboratory, 1972.
- 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.
- Burcat, A. and Radhakrishnan, K., "High Temperature Oxidation of Propene," *Combustion and Flame*, Vol. 60, 1985, pp. 157-169.
- Westbrook, C. W. and Pitz, W. J., "A Comprehensive Chemical Kinetic Reaction Mechanism for Oxidation and Pyrolysis of Propene and Propane," UCRL Preprint 89391, 1983.
- Bittker, D. A. and Scullin, V. J., "GCKP84—General Chemical Kinetics Code for Gas Phase Flow and Batch Processes Including Heat Transfer," NASA TP-2320, 1984.
- Byrne, G. D., Hindmarsh, A. C., Jackson, K. R., and Brown, H. G., "A Comparison of Two ODE Codes: GEAR and EPISODE," *Computational Chemical Engineering*, Vol. 1, 1977, pp. 133-147.
- Shampine, L. F., "Implementation of Implicit Formulas for the Solution of ODEs," *SIAM Journal of Scientific Statistical Computation*, Vol. 1, 1980, pp. 103-118.
- 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," *Journal of Physical Chemistry*, Vol. 81, 1977, pp. 2424-2427.