Journal of Computational Physics **164,** 407–428 (2000)
doi:10.1006/jcph.2000.6605, available online at http://www.idealibrary.com on IDEAL®

A Quasi-Steady-State Solver for the Stiff Ordinary Differential Equations of Reaction Kinetics

David R. Mott,*,[1] Elaine S. Oran,* and Bram van Leer†

*Laboratory for Computational Physics and Fluid Dynamics, Naval Research Laboratory, 4555 Overlook Avenue SW, Washington, District of Columbia 20375; and †Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan 48109-2140
E-mail: mott@lcp.nrl.navy.mil

Received January 5, 2000; revised July 31, 2000

A quasi-steady-state method is presented that integrates stiff differential equations arising from reaction kinetics. This predictor–corrector method is A-stable for linear equations and second-order accurate. The method is used for all species regardless of the time scales of the individual equations, and it works well for problems typical of hydrocarbon combustion. Start-up costs are low, making the method ideal for use in process-split reacting-flow simulations which require the solution of an initial-value problem in every computational cell for every global time step. The algorithm is described, and error analysis and linear stability analysis are included. The algorithm is also applied to several test problems, and the results are compared to those of the stiff integrator CHEMEQ. The method, which we call α-QSS, is more stable, more accurate, and less costly than CHEMEQ.

Key Words: quasi-steady-state; reacting flow; stiff ODE; operator-split methods.

1. INTRODUCTION

Many problems in science and engineering may be described mathematically as coupled sets of ordinary differential equations (ODEs). These may be written generically in terms of the rates of change, $\{g_i\}$, of the dependent variables, $\{y_i\}$, as

$$\frac{dy_i}{dt} = g_i, \quad 1 \le i \le n. \tag{1}$$

When g_i depends on variables other than y_i (that is, the other y_j in the system), these equations are nonlinear.

[1] National Research Council-Naval Research Laboratory Postdoctoral Research Associate.

0021-9991/00 $35.00

Our primary application of Eq. (1) is to sets of coupled, nonlinear ODEs that represent chemical reaction sets. In this case, the dependent variables $\{y_i\}$ are concentrations or densities of reacting chemical species. Sometimes this equation is supplemented by another equation for the change in temperature or energy release that results from the species' interactions. The source term g_i, which is then a function of the concentrations and the thermodynamic state, may be written as the difference of the production rate q_i and the loss rate $p_i y_i$:

$$\frac{dy_i}{dt} = q_i - p_i y_i, \quad 1 \le i \le n\ . \tag{2}$$

The time scales $\tau_i \equiv 1/p_i$ for the various species often differ by many orders of magnitude, and there may be strong coupling between species (i.e., the Jacobian matrix has significant off-diagonal elements). Under these circumstances, the set of equations represented by Eq. (2) is considered *stiff* and does not lend itself readily to numerical solution by classical methods such as the low-order Euler methods or higher-order Adams–Moulton methods [1–3]. Such a system then requires special techniques to solve.

The coupled reaction set represented by Eq. (2) is often a part of a larger model that solves these equations coupled to the partial differential equations describing fluid dynamics. In such cases, the chemical reactions are only one of several processes that might, for example, include advection, diffusion, or radiation transport. Techniques based on *process splitting* (or *operator splitting*) are used to solve such chemically reacting flows [3]. The basic idea in operator splitting is to calculate the effects of individual physical processes separately for a chosen global time step Δt_{g} and then combine the results in some way. Each process in turn can change different system variables during Δt_{g}. Then, when it is time to integrate the ODEs representing the chemical changes within Δt_{g}, the integrator is faced with a new initial value problem in each computational cell. The integrator must therefore solve

$$\frac{dy_i}{dt} = q_i - p_i y_i, \quad y_i(t^0) = y_i^0 \quad 1 \le i \le n, \tag{3}$$

to $t = t^0 + \Delta t_{\mathrm{g}}$. The ODE integration may subdivide Δt_{g} into smaller steps, Δt, to obtain an accurate, stable solution. Here, the time step Δt is called the *chemical time step* because it is the time step that the ODE integrator uses to advance the chemical reactions. The size of Δt generally varies in the course of the calculation.

Given that fluid dynamic calculations are seldom accurate to better than a few percent, any requirement for the chemical integrator to calculate the species concentrations more accurately than a few tenths of a percent is usually excessive. Therefore, the chemical integrator may be relatively low order. Also, since the integrator must solve multiple initial value problems "from scratch" at every global time step, it would be easiest to use a single-point method, which uses only information from the current time level to calculate the concentrations at Δt. This is in contrast to multi-point methods that must store concentration or source-term values from several successive time steps in order to advance the solution. Multi-point methods have a start-up penalty until a sufficient number of steps have been taken to build the history required for the calculation, and they often require interpolation procedures if Δt changes during the integration. By comparison, a single-point method has minimal start-up penalty at the beginning of an integration step.

CHEMEQ [4, 5] is a second-order single-step ODE integrator that has been used successfully as a part of a number of different types of reacting-flow codes. These have included

applications to combustion [6–12] and solar physics [13–15]. CHEMEQ is a *hybrid method*, which means it chooses between a stiff method and a non-stiff method for integrating each ODE within the system depending upon the time scale of that equation. CHEMEQ has been shown to outperform standard stiff ODE solvers by a factor of 50–100 in speed in validation studies on chemical integrations alone (i.e., not coupled to fluid dynamics) when only moderate accuracy was required [4]. More recently, an integrator based heavily on CHEMEQ outperformed a first-order quasi-steady-state method and the implicit preconditioning method CHEMSODE [16] on a photochemical smog problem [17]. Despite its strengths, CHEMEQ exhibits instability under some situations and is limited in the accuracy it can achieve [20].

This report describes a quasi-steady-state method which we call α-QSS. This method is A-stable for linear problems and second-order accurate. It is more stable, more accurate, and less expensive than CHEMEQ, and it successfully integrates some systems for which CHEMEQ fails [20]. CHEMEQ2, a subroutine that employs α-QSS, has now been used successfully in hydrogen–air flame studies in microgravity [18], on studies of pulse-detonation engines [19], on thermonuclear mechanisms used in supernova simulations, and on test cases used to validate CHEMEQ [20]. In addition to describing the new algorithm, we present error and linear stability analyses. Results obtained using CHEMEQ2 are compared to those obtained using CHEMEQ.

2. INTRODUCTION TO QSS METHODS

Consider a simplified form of Eq. (2), in which the subscript i is dropped for convenience, $t^0 = 0$, and $y(t^0) = y^0$,

$$\frac{dy}{dt} = q - py \quad y(0) = y^0. \tag{4}$$

If p and q are constant, then Eq. (4) has an exact solution given by

$$y(t) = y^0 e^{-pt} + \frac{q}{p}(1 - e^{-pt}). \tag{5}$$

Quasi-steady-state (QSS) methods are based on the solution given in Eq. (5) [22–25]. If q and p are slowly varying, evaluating Eq. (5) at $t = \Delta t$ using $q(t^0)$ and $p(t^0)$ provides a good approximation for $y(\Delta t)$. This approach gives a first-order method which is the simplest QSS algorithm. More sophisticated QSS algorithms incorporate the time dependence of p and q and may place Eq. (5) into an alternative algebraic form. The common thread between the QSS methods is their basis in Eq. (5), which requires the methods to return the exact solution if q and p are constant. There are many QSS methods documented in the literature, and the α-QSS method is compared to several of them in Section 3.2.

QSS methods are often compared with standard stiff solvers such as LSODE [26, 27], which is a variable-order method based on Gear's backward differentiation formulae (BDF) [28]. However, such comparisons have been largely limited to the integration of a single problem from one set of initial conditions, not reacting-flow simulations in which start-up overhead and storage requirements play key roles in the overall efficiency of the integrator. Verwer and Simpson describe one such test from atmospheric chemistry, in which a simple two-step BDF method outperforms a first-order implicit QSS method and a two-stage

explicit QSS method. The test involved the calculation of emissions and was not coupled to fluid dynamics [23]. Jay *et al.* introduced two QSS methods and examined their performance on a set of atmospheric tests involving 32 species [24]. These two QSS methods outperformed both a standard, first-order QSS method and CHEMEQ, but the methods were slower than multi-point BDF methods. Variable-order, multi-point BDF methods generally outperform QSS methods when the chemistry integration stands alone. However, the demands of a reacting-flow application are very different than those of a stand-alone integration, and the conclusions of these studies cannot be extrapolated to reacting-flow problems.

Detailed studies of various BDF and QSS methods, as well as other competing integrators, would be required to reconcile these conflicting results and establish the best methods for reacting-flow calculations. In this article, we do not attempt to settle this debate. Instead, we choose to introduce a new QSS method that is well suited to these problems.

3. THE α-QSS ALGORITHM

3.1. *Algorithm Development*

Given the demands of a reacting-flow application, we chose a predictor–corrector implementation for the integrator. Evaluating Eq. (5) at Δt using initial values serves as the predictor step, and a correction based on the initial and the predicted values then follows. The corrector step can be repeated using the previous corrector result as the new predicted value.

First, a convenient algebraic form for Eq. (5) was chosen. Equation (5) can be evaluated at $t = \Delta t$, yielding

$$y(\Delta t) = y^0 + \frac{\Delta t(q - py^0)}{1 + \alpha p \Delta t}, \tag{6}$$

for α defined by

$$\alpha(p\Delta t) \equiv \frac{1 - (1 - e^{-p\Delta t})/(p\Delta t)}{1 - e^{-p\Delta t}}. \tag{7}$$

The parameter α is a function of $p\Delta t$, as shown in Fig. 1. Note that $\alpha \to 0$ as $p\Delta t \to -\infty$; $\alpha \to 1$ as $p\Delta t \to \infty$; and $\alpha = 1/2$ for $p\Delta t = 0$. The meanings of these limits are clarified by recalling that $p\Delta t = \Delta t/\tau$. The $\alpha \to 1$ limit corresponds to an infinitely fast ODE relative to Δt, and $\alpha = 1/2$ corresponds to an infinitely slow ODE. Equation (6) is exact for any value of p (provided q and p are constant). However, we split g such that py is a non-negative loss rate, so only values of $p\Delta t \geq 0$ need be considered.

A predictor–corrector method based on the solution in Eq. (6) takes the form

$$y^p = y^0 + \frac{\Delta t\left(q^0 - p^0 y^0\right)}{1 + \alpha^0 \Delta t p^0} \quad \text{Predictor,} \tag{8}$$

$$y^c = y^0 + \frac{\Delta t\left(q^* - p^* y^*\right)}{1 + \alpha^* \Delta t p^*} \quad \text{Corrector.} \tag{9}$$

Superscript 0 indicates initial values, and superscripts p and c indicate predicted and corrected values, respectively. The predictor uses the initial values of q, p, and y, but the

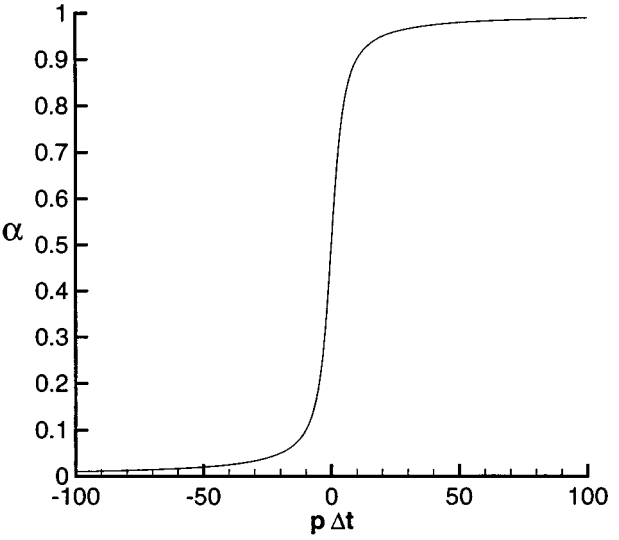

FIG. 1. The parameter α as a function of $p\Delta t$.

"starred" variables (q^*, p^*, y^*, and α^*) can be based on both the initial values and the predicted values.

If we assume linear profiles in time for q and p between the initial and predicted values, we can find an exact series solution for Eq. (2). (This solution is illustrated in conjunction with the error analysis in Section 3.3.) Unfortunately, the series solution does not readily provide an efficient integration technique, nor does it indicate appropriate averages for the starred variables in the corrector. However, solutions do exist under slightly simpler conditions that can be reproduced with appropriate choices of the starred variables.

For instance, if p is constant and q is linear in time, the exact solution to Eq. (2) can be written as

$$y(\Delta t) = y^0 + \frac{\Delta t(\tilde{q} - py^0)}{1 + \alpha \Delta t p}, \tag{10}$$

for $\alpha = \alpha(p\Delta t)$ from Eq. (7) and

$$\tilde{q} = \alpha q(\Delta t) + (1 - \alpha)q^0. \tag{11}$$

Alternatively, if $q = 0$ and p is linear in time, the exact solution of Eq. (2) is

$$y(\Delta t) = y^0 + \frac{\Delta t(-\bar{p}y^0)}{1 + \bar{\alpha}\Delta t\,\bar{p}}, \tag{12}$$

in which

$$\bar{p} = \frac{1}{2}(p(\Delta t) + p^0), \tag{13}$$

and $\bar{\alpha} = \alpha(\bar{p}\Delta t)$ from Eq. (7).

These exact solutions can be reproduced by our predictor–corrector scheme if we use Eq. (14) as the corrector step:

$$y^c = y^0 + \frac{\Delta t\left(\tilde{q} - \bar{p}y^0\right)}{1 + \bar{\alpha}\Delta t\,\bar{p}}. \tag{14}$$

To calculate $\tilde{q}$ and $\bar{p}$ from Eqs. (11) and (13), we replace $q(\Delta t)$ and $p(\Delta t)$ with q^p and p^p. When q and p are known functions of t, the exact values at Δt are used in these expressions. We refer to the new method, which uses Eq. (8) as the predictor and Eq. (14) as the corrector, as α-QSS. This name emphasizes the dual role that α plays in returning the exact solution for constant q and p and in providing a weighted average of q when q is not constant. This method can be applied to problems with general q and p and not just to the simplified cases for which α-QSS returns the exact solution. As shown in Sections 3.3 and 3.4, α-QSS is A-stable for linear problems and second-order accurate for general q and p.

3.2. *Comparison to Previous Methods*

In addition to the algebraic form chosen for Eqs. (8) and (14), α-QSS differs from previous QSS methods in its choice of averaging and its implementation as a predictor–corrector method. Previous methods that calculate average values for p and q use the same averaging method for both terms. For example, the two-stage explicit method introduced by Verwer and Van Loon [22] and tested by Verwer and Simpson [23] uses a simple algebraic average for both q and p calculated from initial and predicted values. CREK1D [25] uses an implicit exponential Euler formulation in which $\alpha(p\Delta t)$ gives a weighted average of the *composite* source terms:

$$y(\Delta t) = y^0 + \Delta t(\alpha g(\Delta t) + (1-\alpha)g^0). \tag{15}$$

In contrast, the α-QSS algorithm uses a simple algebraic average for p and an α-weighted average for q in order to match the exact solutions described earlier.

Other QSS methods combine the results of first-order calculations in a way that improves accuracy. Jay *et al.* [24] describe two such methods. Their "extrapolated QSS" method finds the solution at $t^0 + \Delta t$, first with a single step and then with two steps of $\Delta t/2$ each. A simple extrapolation then estimates the solution that would result if an infinitely small time step were used. Their second method, "symmetric QSS," is a two-step method requiring three evaluations of the source terms. Each of these steps acts as if q and p were constant, and the values for q and p are taken at the same time level based on the previous calculation. No averaging of q or p occurs between time levels in these methods.

The algebraic form of Eqs. (8) and (14), which was introduced in Eq. (6), is based on the asymptotic update employed by CHEMEQ when the time scale for an equation is smaller than some user-specified value [4, 5]. However, CHEMEQ effectively replaces $\bar{\alpha}$ in Eq. (14) with the constant $1/2$, which is equivalent to choosing the Padé approximation [2]

$$\exp(x) \approx \frac{2+x}{2-x} \tag{16}$$

either in the definition of α or in Eq. (5). When the time scale for an equation is larger than some user-specified value, that equation is integrated using the modified Euler method. The hybrid method studied by Lorenzini and Passoni [17] uses CHEMEQ's update equations

but different criteria for determining the time step and for choosing between the asymptotic and standard modified Euler updates. CHEMEQ's asymptotic update also uses different averages in the corrector for p and q than those used in Eq. (14). These differences lead to instability in CHEMEQ that is illustrated in Section 5. The averages chosen by α-QSS eliminate this instability, and α-QSS automatically approaches the modified Euler method as $p\Delta t \to 0$.

3.3. *Error Analysis*

The method has a third-order error term for a single step, which makes it second-order accurate over the course of an integration. This can be shown by examining the exact series solution of Eq. (1). Writing the series for $y(t)$, $q(t)$, and $p(t)$ about initial values y_0, q_0, and p_0 at $t^0 = 0$ gives

$$y(t) = y_0 + y_1 t + y_2 t^2 + \cdots = \sum_{j=0}^{\infty} y_j t^j, \tag{17}$$

$$q(t) = \sum_{j=0}^{\infty} q_j t^j, \tag{18}$$

$$p(t) = \sum_{j=0}^{\infty} p_j t^j. \tag{19}$$

This development deals with a single species, y, so subscript j corresponds to the coefficient of the t^j terms in the expansions in Eqs. (17)–(19) and not the jth species in a multi-species system. Substitution into Eq. (4) provides the coefficients for $y(t)$ in Eq. (17),

$$y_j = \frac{1}{j}\left(q_{j-1} - \sum_{k=0}^{j-1} p_{j-1-k} y_k\right), \tag{20}$$

for $j > 0$.

In general, q and p are given as functions of y, not as functions of t. Therefore, the coefficients in Eqs. (18) and (19) are not known, and Eq. (4) is a nonlinear differential equation. We will first perform an error analysis for the linear version of Eq. (4), in which q and p are known functions of t, and then extend this analysis to the nonlinear case. For the linear case, the predicted values are simply $q^p = q(\Delta t)$ and $p^p = p(\Delta t)$. Subtracting the series expansion for Eq. (14) from the exact solution evaluated at $t = \Delta t$ yields

$$y(\Delta t) - y^c = \frac{\Delta t^3}{6}\left(-\frac{1}{2} p_1 q_0 - q_2 + p_2 y_0\right) + O(\Delta t^4) \quad \text{[linear case]}. \tag{21}$$

The leading error term is $O(\Delta t^3)$ per time step. Since the number of time steps required to reach a given time is proportional to $1/\Delta t$, the error for the method is second-order [1].

The method is second-order for nonlinear problems as well. To illustrate this, first note that the leading error term for the predicted values y^p is second-order:

$$y(\Delta t) - y^p = \frac{\Delta t^2}{2}(q_1 - p_1 y_0) + O(\Delta t^3). \tag{22}$$

Since q and p are polynomials in the species concentrations for the nonlinear systems representing reaction kinetics, the leading error terms for the predicted values q^p and p^p are also second-order. This error can be represented as

$$q(\Delta t) - q^p = \epsilon_q \Delta t^2 + O(\Delta t^3), \tag{23}$$

$$p(\Delta t) - p^p = \epsilon_p \Delta t^2 + O(\Delta t^3), \tag{24}$$

for some unknown coefficients ϵ_q and ϵ_p. Using these predicted values in Eq. (14) gives an error term of the form

$$y(\Delta t) - y^c = \Delta t^3 \left(-\frac{1}{12} p_1 q_0 - \frac{1}{6} q_2 + \frac{1}{6} p_2 y_0 + \frac{1}{2} \epsilon_q - \frac{1}{2} \epsilon_p y_0 \right) + O(\Delta t^4)$$

[nonlinear case]. (25)

As with the linear problem, the leading-order error term for the nonlinear problem is $O(\Delta t^3)$, and the method is still second-order over the course of an integration.

3.4. *Linear Stability Analysis*

For the single linear equation

$$\frac{dy}{dt} = \lambda y, \tag{26}$$

the coefficient λ can be a function of t but not a function of y. Using the average value $\bar{\lambda}$ given by

$$\bar{\lambda} = \frac{1}{2}(\lambda(t = 0) + \lambda(t = \Delta t)), \tag{27}$$

α-QSS has amplification factor G given by

$$G = 1 + \frac{\bar{\lambda} \Delta t}{1 - \bar{\alpha} \bar{\lambda} \Delta t}. \tag{28}$$

The signs in Eq. (28) reflect the fact that $\lambda = -p$, and note that $\bar{\alpha} = \alpha(-\bar{\lambda} \Delta t)$. Using Eq. (7), the expression for G simplifies to

$$G = \exp(\bar{\lambda} \Delta t). \tag{29}$$

For $\bar{\lambda} = a + b\sqrt{-1}$ with a, b both real, the magnitude of G is simply

$$\|G\| = \exp(a \Delta t). \tag{30}$$

Since $\|G\| \leq 1$ for $a \leq 0$ for any value of b, the method is A-stable. This does not prove that α-QSS is A-stable when applied to nonlinear systems of ODEs for which $\{p_i\}$ and $\{q_i\}$ depend on $\{y_i\}$. However, in testing to date, an accuracy-based time step criterion has worked well for the QSS update. This criterion, used originally in the subroutine CHEMEQ [5], was used in CHEMEQ2 [21] to generate the results presented in Section 5.

4. IMPLEMENTATION ISSUES

The α-QSS update is used on all equations in the system regardless of the time scale of the ODE, and iterations may be done on the corrector that improve the accuracy of the result. Again using superscript 0 to indicate values at the beginning of the chemical time step and subscript i to specify species i, the QSS update is given by

$$y_i^{\mathrm{p}} = y_i^0 + \frac{\Delta t g_i^0}{1 + \alpha_i^0 \Delta t p_i^0} \qquad \text{Predictor,} \tag{31}$$

$$y_i^{\mathrm{c}} = y_i^0 + \frac{\Delta t \left(\tilde{q}_i - \overline{p_i} y_i^0\right)}{1 + \overline{\alpha_i} \Delta t \overline{p_i}} \qquad \text{Corrector.} \tag{32}$$

The predictor uses all initial values, and $\alpha_i^0 = \alpha(p_i^0 \Delta t)$. After calculating the predicted concentrations $\{y_i^p\}$ for all of the species in the system, next obtain $\{q_i^{\mathrm{p}}\}$ and $\{p_i^{\mathrm{p}}\}$ from $\{y_i^{\mathrm{p}}\}$, and then calculate

$$\overline{p_i} = \frac{1}{2}\left(p_i^0 + p_i^{\mathrm{p}}\right), \tag{33}$$

$\overline{\alpha_i} = \alpha(\overline{p_i} \Delta t)$, and finally

$$\tilde{q}_i = \overline{\alpha_i} q_i^{\mathrm{p}} + (1 - \overline{\alpha_i}) q_i^0. \tag{34}$$

Equation (32) then gives the corrected concentrations $\{y_i^{\mathrm{c}}\}$. To iterate on the corrector, use the values $\{y_i^{\mathrm{c}}\}$ from one step as $\{y_i^{\mathrm{p}}\}$ for the next step.

Having an accurate approximation for $\alpha(p\Delta t)$ that does not require an evaluation of the exponential function makes the method given by Eqs. (31) and (32) more attractive. Recall that p is strictly non-negative based on the way the chemical source term is split, so this approximation need only hold for positive values of $p\Delta t$. Equation (7) indicates that as $p\Delta t \to \infty$, a reasonable approximation for $\alpha(p\Delta t)$ is

$$\alpha(p\Delta t) \approx 1 - \frac{1}{p\Delta t}. \tag{35}$$

Using this approximation for α eliminates the need to find an accurate approximation for e^{-x} as $x \to \infty$, as would be required if the solver were based on Eq. (5) rather than Eq. (6). Using the Padé approximation [2]

$$e^x \approx \frac{1680 + 840x + 180x^2 + 20x^3 + x^4}{1680 - 840x + 180x^2 - 20x^3 + x^4} \tag{36}$$

in the definition of $\alpha(p\Delta t)$ gives

$$\alpha(p\Delta t) \approx \frac{840r^3 + 140r^2 + 20r + 1}{1680r^3 + 40r}, \tag{37}$$

for $r \equiv 1/(p\Delta t)$. These two approximations are shown with the exact curve for α in Fig. 2. The approximation given by Eq. (37) is labeled Padé (a). Unlike Fig. 1, the x-coordinate in Fig. 2 is r. The linear approximation in Eq. (35) is closer to the exact value of α than the approximation in Eq. (37) for $r \leq 0.16762$; for $r > 0.16762$, Eq. (37) is more accurate.

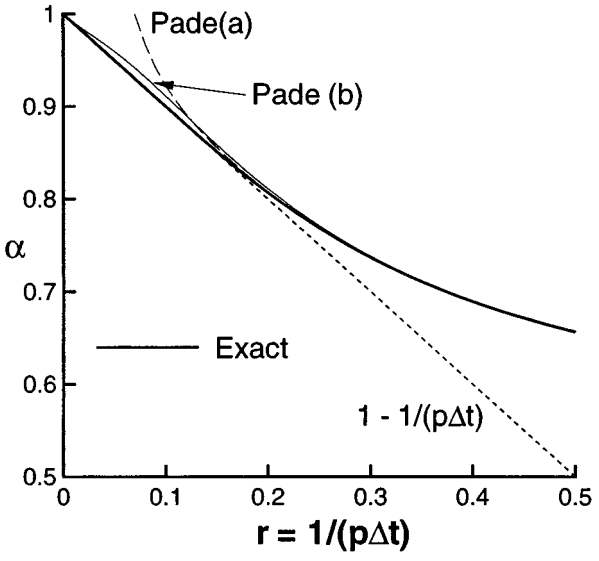

FIG. 2. Approximations for α as a function of $r = 1/(p\Delta t)$.

Therefore, the better approximation can be chosen based upon the value of r. This combined approximation differs from the exact value of α by at most 0.3%. This error occurs in a narrow region around the transition from the linear to the rational approximation.

A second approximation for α results from the Padé approximation

$$e^x \approx \frac{360 + 120x + 12x^2}{360 - 240x + 72x^2 - 12x^3 + x^4}. \tag{38}$$

This approximation gives

$$\alpha(p\Delta t) \approx \frac{180r^3 + 60r^2 + 11r + 1}{360r^3 + 60r^2 + 12r + 1}. \tag{39}$$

This second approximation for α is designated Padé (b) and is also shown in Fig. 2. Since this approximation recovers $\alpha = 1$ for $r = 0$ and $\alpha \to 1/2$ for $r \to \infty$, Eq. (39) can be used alone with only a slight accuracy penalty compared to the combined approximation of Eqs. (35) and (37). In testing to date, this accuracy penalty in α has not caused accuracy problems in the species solutions, and using this single equation eliminates the logic check required to determine which of Eqs. (35) or (37) to use for the combined method.

Accuracy is controlled by choosing Δt and the number of corrector iterations, N_c. A single corrector calculation is performed if $N_c = 1$, and as Section 5 illustrates, increasing N_c improves accuracy. The subroutine CHEMEQ2 used to test α-QSS chooses Δt using the same accuracy-based criterion as CHEMEQ. An introduction to the time step criterion is needed to understand the results comparison of Section 5, but the specifics of the implementation are not included here. The time step selection algorithm is fully documented elsewhere [20, 21]. The difference between the predicted and corrected concentration values is modeled as a single second-order term:

$$y_i^c - y_i^p = a_2(\Delta t)^2. \tag{40}$$

The user specifies a target magnitude for this correction term using a parameter ε:

$$\|a_2(\Delta t)^2\|_{\text{target}} = \varepsilon y_i^{\text{c}}. \tag{41}$$

The coefficient a_2 is calculated from Eq. (40) using values for y_i^{c}, y_i^{p}, and Δt for the current update. A new Δt is then calculated based on Eq. (41). In the numerical implementation, the Δt calculation is biased such that Δt is lowered quickly if the correction term is too large, and the species requiring the most restrictive Δt governs the calculation [5].

The effect of the thermodynamic state on the reaction rate constants has been ignored in the previous developments. The rate constants are often calculated once before the predictor step using the initial values and held constant during the corrector step. A new thermodynamic state is then found at the new time level and used for the following predictor and corrector. If the integration is particularly sensitive to the thermodynamic state, this state can be recalculated for the corrector using the predicted solution. If the system requires the integration of a thermodynamic variable (such as temperature) along with the species concentrations, then the source term for this extra variable is split just as with the concentrations. If there is no "loss" term for that variable that can be assumed proportional to the variable, then the entire source term is assigned to q, and the method reduces to the modified Euler method for that equation since $\alpha(0) = 1/2$.

Linear stability analysis indicates that α-QSS is A-stable, but this result holds no guarantees for the nonlinear systems of chemical kinetics. In a stand-alone integration, the accuracy-based time step criterion used by CHEMEQ2 can allow the time step to grow large enough as equilibrium is approached so that the method becomes unstable [20]. This phenomenon is illustrated in Section 5 and discussed further in Section 6. For such cases, a stability constraint such as that discussed in Section 6 may be used, or a ceiling on the time step may be imposed. In a reacting-flow code, the global time step acts as a ceiling on the chemical time step, so this instability problem is minimal.

5. NUMERICAL RESULTS

Two examples are described here in detail. The first is a system of equations involving cesium and cesium ions that was originally suggested by D. Edelson of Bell Laboratories. This test was used to compare the original CHEMEQ subroutine to other stiff solvers, including those of Gear and Kregel, as shown in [5]. The second set of tests involves a hydrogen–oxygen combustion mechanism and focuses on the effect of corrector iteration on the timing and accuracy of α-QSS. The α-QSS calculations were performed by the subroutine CHEMEQ2, which is the subroutine CHEMEQ with its hybrid method replaced with α-QSS. Two reacting-flow applications are then discussed briefly in Section 5.3.

5.1. *Cesium Tests*

The cesium mechanism, shown in Table I, involves seven species and seven one-way reactions. The rate constants k_i are fixed at the values shown. The inert collision partner, M in reaction 5, may be Cs, CsO_2, O_2, or N_2, so the concentration of M used to calculate the reaction rate is the sum of the concentrations of these four species. The initial conditions and the solution values at 1000 s used for the accuracy study are included in Table II [5]. These solution values, which we call the "accepted values" in the following error analysis,

TABLE I
Cesium Mechanism

	Reaction	k_i
1)	$O_2^- + Cs^+ \xrightarrow{k_1} Cs + O_2$	5×10^{-8} cm^3/s
2)	$Cs^+ + e^- \xrightarrow{k_2} Cs$	1×10^{-12} cm^3/s
3)	$Cs \xrightarrow{k_3} Cs^+ + e^-$	3.24×10^{-3} s^{-1}
4)	$O_2^- \xrightarrow{k_4} O_2 + e^-$	4×10^{-1} s^{-1}
5)	$O_2 + Cs + M \xrightarrow{k_5} CsO_2 + M$	1×10^{-31} cm^6/s
6)	$O_2 + e^- + O_2 \xrightarrow{k_6} O_2^- + O_2$	1.24×10^{-30} cm^6/s
7)	$O_2 + e^- + N_2 \xrightarrow{k_7} O_2^- + N_2$	1×10^{-31} cm^6/s

are the common result of running LSODE and CHEMEQ2 at excessively high accuracies. The species number densities as a function of time for this case are shown in Fig. 3.

Time step histories for CHEMEQ2 and CHEMEQ are shown in Fig. 4. As mentioned earlier, the asymptotic update used by CHEMEQ is unstable under some circumstances, and the details of this instability are documented elsewhere [20]. This cesium integration is an example of when CHEMEQ becomes unstable. CHEMEQ produces oscillations in Δt between 20 s and 1000 s. CHEMEQ2 does not produce these oscillations, although the accuracy-based time step constraint lowers the time step in this region.

A series of studies evaluated the accuracy of CHEMEQ2 compared to CHEMEQ. These solved the Cs test problem given above and used the reference solution at 1000 s as a benchmark. The tests varied the value of ϵ from 10^{-1} to 10^{-6}. Additional tests fixed ϵ and varied N_c from 1 to 10. Figure 5 summarizes the results of the tests by showing the rms error as a function of CPU time, which was scaled by the smallest increment the timing routine could resolve. The CHEMEQ2 results are shown as a series of overlapping profiles of the shape shown in the schematic in Fig. 6. Each profile is for a fixed value of ϵ, and the points on it correspond to different values of N_c.

The error computed for each computation (fixed ϵ and N_c) is based on the the accepted values at 1000 s. The relative error e_i for each species i is

$$e_i = \frac{y_{i,\text{accepted}} - y_{i,\text{calculated}}}{y_{i,\text{accepted}}}. \tag{42}$$

TABLE II
Initial and t = 1000 s Species Concentrations for the Cesium Mechanism Test Problem

Species	y_i(0 s) (cm^{-3})	y_i(1000 s) [cm^{-3}]
e^-	1×10^2	4.9657897283×10^4
O_2^-	5.2×10^2	2.5913949444×10^4
Cs^+	6.2×10^2	7.5571846728×10^4
Cs	1×10^{12}	1.5319405460×10^3
CsO_2	0	1.000×10^{12}
N_2	1.4×10^{15}	1.400×10^{15}
O_2	3.6×10^{14}	3.590×10^{14}

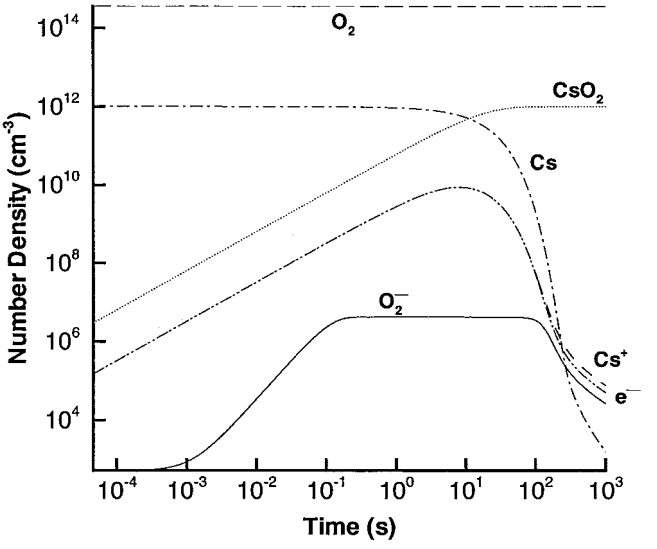

FIG. 3. Species number densities as a function of time for the cesium mechanism test problem.

A root-mean-square error for the six reacting species (excluding the inert N_2) is

$$e_{\text{rms}} = \sqrt{\frac{\sum_{i=1}^{6} e_i^2}{6}}. \tag{43}$$

There is only a single curve for CHEMEQ in Fig. 5. Each point on this curve corresponds to a different value of ε. The hybrid method, as implemented in CHEMEQ and used in this problem, becomes unstable and the solutions are corrupted if multiple corrector iterations are used. Lorenzini and Passoni, however, were able to use multiple corrector

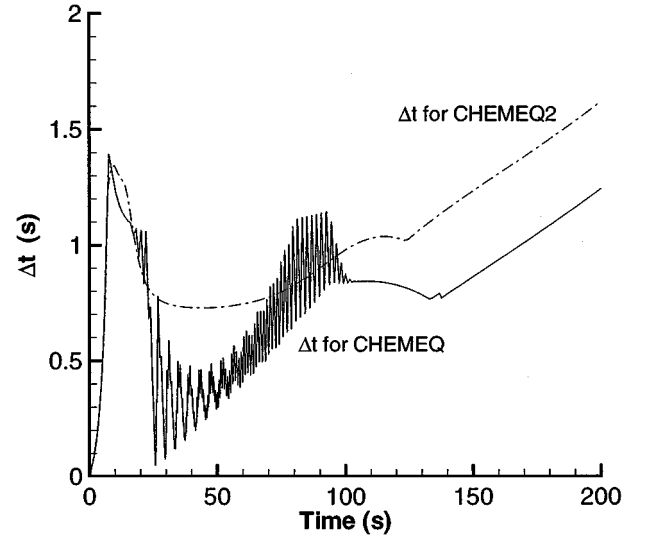

FIG. 4. Time step histories for the cesium integration using CHEMEQ and CHEMEQ2 for $\varepsilon = 0.01$.

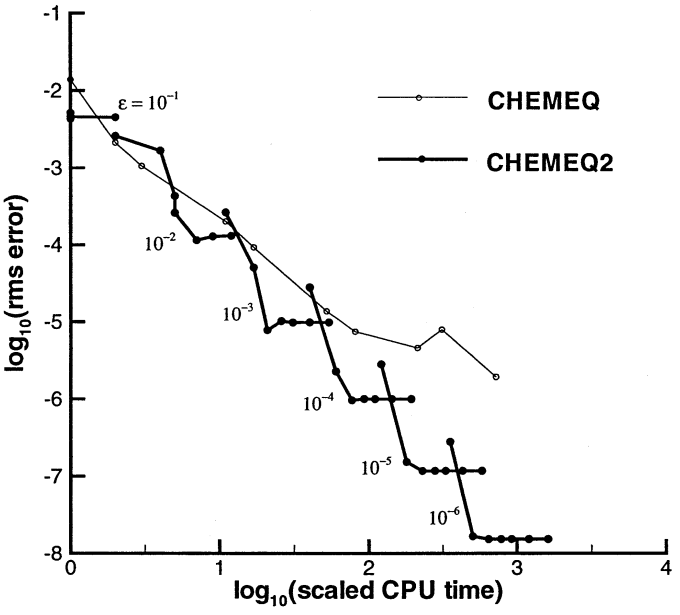

FIG. 5. Root mean square error at 1000 s versus scaled CPU time to reach 1000 s for CHEMEQ and CHEMEQ2 for a range of ε and N_c. A schematic of each CHEMEQ2 curve is shown in Fig. 6, and the CHEMEQ results correspond to $\varepsilon = 10^{-1}$ to 10^{-6}.

iterations successfully in other implementations of the hybrid method for other problems [17]. CHEMEQ2 does not have this instability problem.

For a single iteration and large enough ε, the CHEMEQ2 results lie roughly along the CHEMEQ curve. In this case, the CHEMEQ2 simulation takes less time, but gives a less

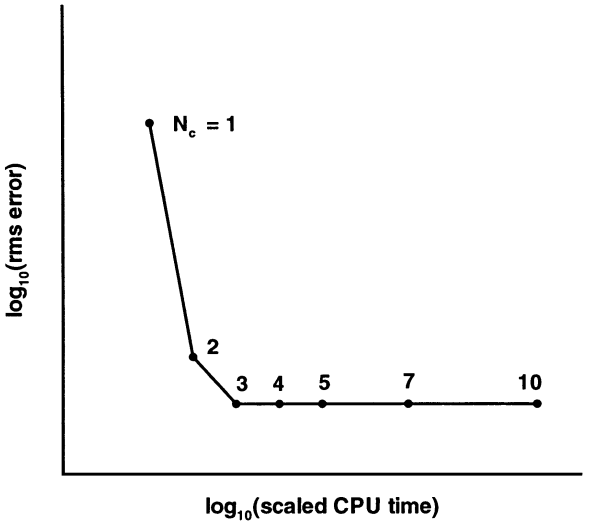

FIG. 6. Schematic of the types of profiles for fixed ε in Fig. 5. The numbers next to each symbol give the corresponding value of N_c. As described in the text, the solutions for the cesium test problem converge after about three iterations.

accurate solution. As ε is decreased, the CHEMEQ results do not give the same increase in accuracy for the increased computational costs.

The curves shown in Fig. 5 can be explained by comparing the CHEMEQ and the CHEMEQ2 algorithms. The CHEMEQ stiff predictor is identical to the CHEMEQ2 predictor in the limit as $\alpha^0 \to 1$, which corresponds to $p^0 \Delta t \to \infty$. The CHEMEQ stiff corrector, however, uses different average values for q and p than the CHEMEQ2 corrector and also effectively uses the $p\Delta t \to 0$ limit value of $\alpha = 1/2$. This inconsistency in the effective α between CHEMEQ's stiff predictor and corrector limits the growth of Δt for CHEMEQ. Therefore, CHEMEQ takes a smaller time step than CHEMEQ2 for the same ε, and, for moderate accuracy, this inconsistency in α does not affect the accuracy of the solution. The best accuracy achievable by CHEMEQ suffers from this inconsistency, so as ε becomes smaller, CHEMEQ2 gives more accurate answers than CHEMEQ.

The CHEMEQ2 curves for a fixed ε show dramatic increases in accuracy after just a few iterations. After about three iterations, the curves for a given ε flatten, which indicates that the method has converged to a final corrector value, and additional iterations will not improve the accuracy. The computational expense in adding iterations is less than that in reducing ε for similar improvements in accuracy. As ε is lowered, accuracy improves because the time step is decreased. As the number of iterations increases, accuracy improves because the corrector is able to refine the linear approximation for p and q used to calculate $\tilde{q}$ and $\bar{p}$ for the corrector equation, Eq. (32). Not all systems will converge for such low values of N_c, but, in general, iterating the corrector improves the accuracy.

For the CHEMEQ2 curve for $\varepsilon = 0.1$, the simulations took so little time that the precision of the timing routines was not sufficient to measure differences in timing between these runs. In addition, the calculations were performed on a computer that allows access to multiple users. These effects contribute to the error and uncertainty in the low-resolution data.

5.2. *Hydrogen–Air Tests*

The H_2–air combustion mechanism used consists of 25 reversible reactions involving nine species (including inert N_2) [32]. The reaction rates are calculated using the modified Arrhenius form

$$k_r = AT^B \exp(-C/T), \tag{44}$$

where T is the temperature. The rate k_r is either a forward or backward rate. The parameters A, B, and C for both the forward and backward rates for each reaction are given in Ref. [20]. Initially the mixture is at 1000 K and a pressure of 1 atmosphere and in the ratio $2:1:3.76$ for $H_2:O_2:N_2$. These conditions lead to initial number densities on the order of 10^{18} cm^{-3} for these three species. A minimum number density of 10^{-30} cm^{-3} is imposed on the other species to prevent numerical difficulties. Nitrogen is inert for the mechanism and thus acts as a diluent.

Selected species' number densities for this problem are presented as a function of time in Fig. 7. The figure shows that after an induction time of about 3.4×10^{-4} s, H_2 and O_2 are converted to H_2O in a relatively short time period. During this induction time radicals are formed that eventually initiate the rapid conversion of H_2 and O_2. Here, we focus on the H number density profile, which has a peak in the reaction zone that is difficult to predict accurately. A series of calculations examined the effect of ε and N_c on the location and the

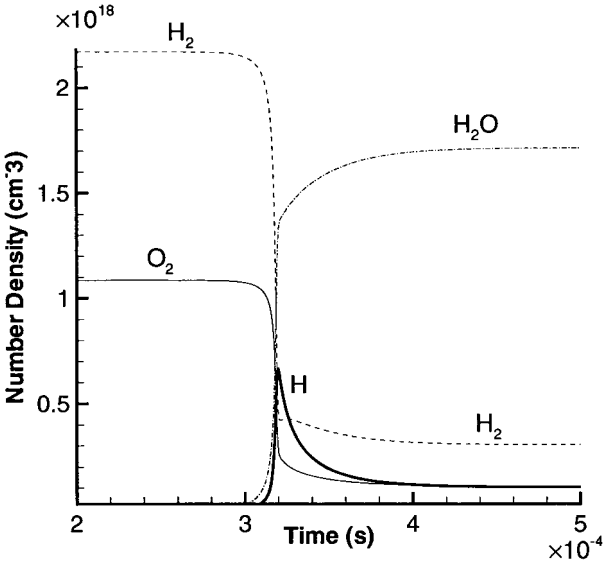

FIG. 7. Solution for the single-point hydrogen–air integration.

value of this peak. The errors in these parameters are calculated as

$$t_{\mathrm{p,error}} = \frac{\|t_{\mathrm{p}} - t_{\mathrm{p,reference}}\|}{t_{\mathrm{p,reference}}} \tag{45}$$

$$(n_{\mathrm{H}})_{\mathrm{p,error}} = \frac{\|(n_{\mathrm{H}})_{\mathrm{p}} - (n_{\mathrm{H}})_{\mathrm{p,reference}}\|}{(n_{\mathrm{H}})_{\mathrm{p,reference}}} \tag{46}$$

for peak number density value $(n_{\mathrm{H}})_{\mathrm{p}}$ at time t_{p}, compared to reference values. The reference values were obtained by integrating the equations with CHEMEQ2 for increasing N_{c} and decreasing ε values, until the solution ceased changing. The solution was then verified by comparison with a solution obtained by a simple modified Euler method using an exceptionally low error tolerance. Table III lists these errors and the CPU time required to reach 5×10^{-4} seconds for a variety of ε and N_{c} values. These calculations were performed on a DEC Alpha workstation, and the CPU times in Table III are scaled using the CPU time for the $\varepsilon = 10^{-3}$, $N_{\mathrm{c}} = 1$ simulation.

Figure 8 shows results of integrations for $\varepsilon = 10^{-4}$ and $N_{\mathrm{c}} = 1, 5$, and 10. This should be contrasted against the cesium calculations of the previous section that converged by $N_{\mathrm{c}} = 5$. In this case, the profiles are converging to the reference solution, but they have not completely converged by $N_{\mathrm{c}} = 10$. Table III suggests that reducing ε may be a more efficient way to improve the accuracy of the solution than increasing the iteration count. The errors are not of the same order as ε, however, and reducing ε by an order of magnitude does not result in a comparable reduction in the error. The errors in the time-to-peak and the peak value are not even comparable, with the $(n_{\mathrm{H}})_{\mathrm{p}}$ much more prone to error than t_{p}. This peak is very difficult for a low-order method to calculate. A higher-order method that employs information from several time steps would provide a much better result for this problem.

The question remains as to how accurate the integration can become if the number of iterations is increased dramatically. The results for $\varepsilon = 10^{-3}$ from Table III are repeated in

TABLE III
Results Obtained by Varying ε and N_c for the Hydrogen–Air Reaction Integration

	N_c		
ε	1	5	10
	(a) $t_{p,\,Error}$		
10^{-3}	6.66×10^{-2}	2.97×10^{-2}	1.84×10^{-2}
10^{-4}	2.79×10^{-2}	1.10×10^{-2}	6.29×10^{-3}
10^{-5}	1.06×10^{-2}	3.67×10^{-3}	1.97×10^{-3}
	(b) $(n_H)_{p,\,Error}$		
10^{-3}	0.392	0.166	9.40×10^{-2}
10^{-4}	0.146	4.56×10^{-2}	2.35×10^{-2}
10^{-5}	4.48×10^{-2}	1.27×10^{-2}	6.49×10^{-3}
	(c) Scaled CPU Times to 5×10^{-4} s		
10^{-3}	1.00	2.92	5.33
10^{-4}	3.19	9.92	18.3
10^{-5}	11.8	36.7	67.5

Table IV, and additional results obtained using 1000 corrector iterations are also included. For the $N_c = 1000$, the error in the peak value is an order of magnitude less than ε, and the time-to-peak error is three orders of magnitude lower than ε. This suggests that the corrector equation, Eq. (32), provides an accurate representation once it is sufficiently converged.

CHEMEQ2 inherits a final characteristic from CHEMEQ. Figure 9 shows the H profile as the system reaches equilibrium. The CHEMEQ2 results indicate that the accuracy-based time step can be too large for the corrector iteration to remain stable despite the fact that the

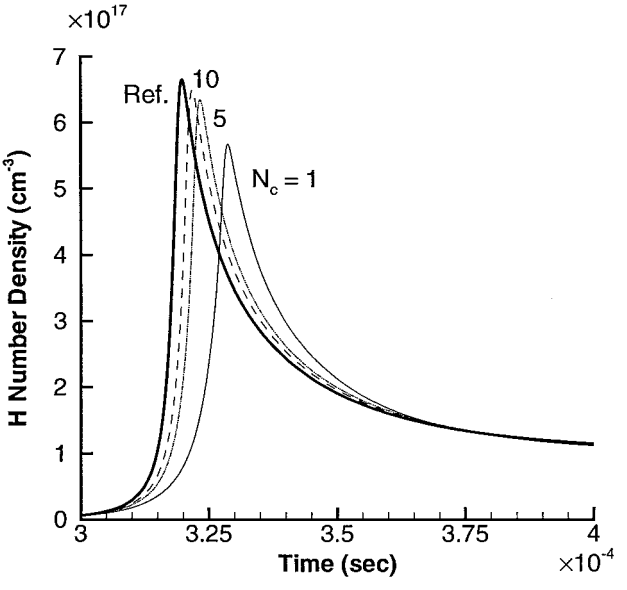

FIG. 8. Hydrogen number density for $N_c = 1$, 5, and 10, and $\varepsilon = 10^{-4}$. The dark, solid line is the reference solution, and the numbers next to the remaining curves indicate the value of N_c for each profile.

TABLE IV

Errors in t_p and $(n_H)_p$ for $\varepsilon = 10^{-3}$ and N_c = 1, 5, 10, and 1000, and the Scaled CPU Time Required for Each Simulation to Reach $t = 5 \times 10^{-4}$ Seconds

Iterations	t_p error	$(n_H)_P$ error	CPU time
1	6.66×10^{-2}	0.392	1.00
5	2.97×10^{-2}	0.166	2.92
10	1.84×10^{-2}	9.40×10^{-2}	5.33
1000	2.71×10^{-6}	1.77×10^{-4}	489

stability analysis indicated that α-QSS is A-stable for linear problems. Note that the scale in Fig. 9 is exaggerated; the range covered by the number density axis spans approximately 1% of the equilibrium value. This is not a problem in reacting-flow applications, as the global time step limits how large the chemical time step becomes. In this single-point integration, however, the instability is seen.

To overcome this instability, the convergence of the corrector iteration can be monitored. Let $y_i^{c(l)}$ denote the corrected value of $y_i(\Delta t)$ after l iterations. The change from one iteration to the next,

$$\Delta y_i^{c(l)} = y_i^{c(l)} - y_i^{c(l-1)}, \tag{47}$$

should decrease in size as l grows if the iteration is stable. The profile given as open circles

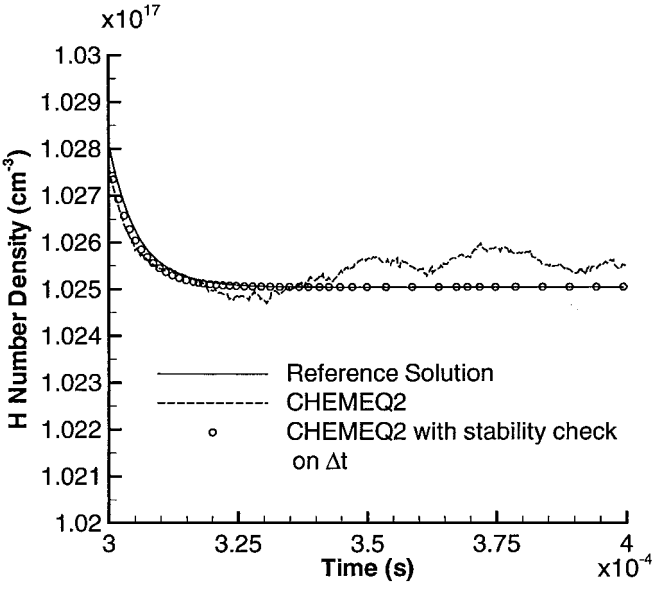

FIG. 9. Hydrogen number density as the integration approaches equilibrium, $\varepsilon = 10^{-5}$, $N_c = 10$. The dashed line is the standard CHEMEQ2 result. The profile given by open circles includes the stability constraint on Δt (see Eq. (48)).

in Fig. 9 was generated using CHEMEQ2 with the additional requirement that

$$\left\|\Delta y_i^{c(N_c)}\right\| < \left\|\Delta y_i^{c(N_c-1)}\right\|, \tag{48}$$

where N_c is the specified maximum number of iterations. The oscillations in the number density disappear when this constraint is added, and the predicted equilibrium value agrees well with the reference solution.

5.3. *Reacting-Flow Solutions*

Two reacting-flow cases will be briefly discussed here. These results are provisional, as no rigorous, systematic studies have been performed. A thorough comparison between integration methods would include the effects of implementation choices, accuracy requirements, and stiffness. The stiffness issues are not limited to the chemical mechanism itself but also include coupling of the chemical time scales and the fluid dynamics time scales (i.e., how much the integrator subdivides the global time step in order to perform the chemistry integration). Such a study is planned for the future. However, from our experiences, we expect the results described below to be typical.

Uphoff *et al.* [33] studied two-dimensional detonation formation using an H_2/O_2 mechanism with 18 reactions and eight species. They compared process-split reacting-flow calculations using CHEMEQ and METAN1 [34] as the chemistry integrator. METAN1 is a general stiff solver which employs a semi-implicit mid-point rule and extrapolation to a "zero step size" solution [35–37]. For this specific set of calculations, CHEMEQ performed the required chemical integrations in approximately one-sixth the time required by METAN1. Documentation of accuracy parameters used and solution options chosen for the calculations is not available.

An additional calculation was performed in order to compare the efficiency of α-QSS to a Gear method. A one-dimensional hydrogen–air premixed flame was simulated using a process-split method [38] which employed FCT for integrating the fluid convection [39]. The chemistry integration was performed using CHEMEQ2, and also using DEBDF, which employs a variable-order Gear method as implemented in LSODE. DEBDF is part of SLATEC, a library of computational subroutines available on Silicon Graphics and Cray computers [40]. CHEMEQ2 performed the required calculations in approximately one-sixth the time required by DEBDF, which is coincidentally the same factor seen in the detonation comparison versus METAN1. No extensive accuracy studies have been performed to ensure that the comparison was fair. For example, the accuracy parameters for CHEMEQ2 and DEBDF were simply set to the same value, even though the two codes do not use these parameters in exactly the same way.

6. DISCUSSION AND SUMMARY

The α-QSS method is intended to be a general purpose integrator for equations that are reasonably represented by the form in Eq. (2). In the current predictor–corrector form, the method works best as a very low-overhead, moderately accurate technique. Accuracy can be enhanced by increasing the number of corrector iterations performed. For some systems, convergence is very fast, and accuracy is greatly enhanced within a few iterations. For systems that converge more slowly, reducing the accuracy parameter ε used to determine the time step is more effective in improving accuracy.

For the slowly converging H_2–air integration, the corrector provided a very accurate solution once convergence was obtained. This suggests that Eq. (32) should be used as a constraint equation to be satisfied by a more efficient iteration than simple back-substitution. Future development of the algorithm will therefore focus on an implicit form of Eq. (32),

$$y_i(\Delta t) = y_i(0) + \frac{\Delta t(\tilde{q} - \bar{p}y_i(0))}{1 + \bar{\alpha}\Delta t\,\bar{p}}, \tag{49}$$

in order to find an iteration scheme that approaches the converged solution more quickly than Eq. (32). Equation (49) is implicit since $\{\overline{\alpha_i}\}$, $\{\tilde{q}_i\}$, and $\{\overline{p_i}\}$ depend upon $\{y_i(\Delta t)\}$. This new iteration scheme could also help with the instability seen in Fig. 9 as equilibrium is approached.

Additional analysis should also be undertaken to find a better time step criterion for α-QSS. The criterion based on Eq. (41) does not directly measure the accuracy of the solution and allows the corrector iteration to become unstable under certain circumstances. Figure 5 illustrates that the rms error for the converged solution is roughly two orders of magnitude smaller than the "error" prescribed by ε. This discrepancy arises because Δt is chosen based on the difference between the predicted value and the final corrected value, despite the the fact that α-QSS is exact for some conditions when these values differ greatly. Monitoring the convergence of the corrector iteration can eliminate the instability that the accuracy-based criterion does not prevent. Also, the behavior of CHEMEQ, which effectively uses $\alpha = 1/2$ in its asymptotic corrector, suggests that an approximation for $\alpha(p\Delta t)$ much simpler than those in Eqs. (36) and (38) could be used provided it still recovered the proper limits as $p\Delta t \to 0$ and $p\Delta t \to \infty$.

ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research and completed while the first author was a National Research Council–Naval Research Laboratory Research Associate. The authors also thank T. R. Young, Jr., and J. P. Boris, who developed the asymptotic method in CHEMEQ, for their help and advice. We also thank G. Patnaik, who streamlined CHEMEQ for workstation use and performed the hydrogen–air flame calculations discussed in Section 5.3.

REFERENCES

1. J. D. Hoffman, *Numerical Methods for Engineers and Scientists* (McGraw–Hill, New York, 1992).
2. J. D. Lambert, *Numerical Methods for Ordinary Differential Systems; The Initial Value Problem* (Wiley, New York, 1991).
3. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow* (Elsevier, New York, 1987).
4. T. R. Young and J. P. Boris, A numerical technique for solving ordinary differential equations associated with the chemical kinetics of reactive-flow problems, *J. Phys. Chem.* **81**, 2424 (1977).
5. T. R. Young, Jr., *CHEMEQ—A Subroutine For Solving Stiff Ordinary Differential Equations*, NRL Memorandum Report 4091 (Naval Research Laboratory, Washington, DC, 1980).
6. E. S. Oran, T. R. Young, and J. P. Boris, Application of time-dependent numerical methods to the description of reactive shocks, in *Proceedings of 17th International Symp. on Combustion* (The Combustion Institute, Pittsburgh, 1979), p. 43.
7. K. Kailasanath, E. S. Oran, and J. P. Boris, A theoretical study of the ignition of premixed gasses, *Combust. Flame* **47**, 173 (1982).

8. E. S. Oran, T. R. Young, J. P. Boris, and A. Cohen, Weak and strong ignition. I. Numerical simulations of shock tube experiments, *Combust. Flame* **48**, 135 (1982).
9. E. S. Oran and J. P. Boris, Weak and strong ignition. II. Sensitivity of the hydrogen–oxygen system, *Combust. Flame* **48**, 149 (1982).
10. K. Kailasanath and E. S. Oran, Ignition of flamelets behind incident shock waves and the transition to detonation, *Combust. Sci. Technol.* **34**, 345 (1983).
11. J. W. Weber, Jr., E. S. Oran, J. D. Anderson, Jr., and G. Patnaik, Load balancing and performance issues for the data parallel simulation of stiff chemical nonequilibrium flow, *AIAA J.* **35**, 147 (1998).
12. E. S. Oran, J. W. Weber, Jr., E. I. Stefaniw, M. H. Lefebvre, and J. D. Anderson, Jr., A numerical study of a two-dimensional H_2–O_2–Ar detonation using a detailed chemical reaction model, *Combust. Flame* **113**, 147 (1998).
13. G. A. Doschek, J. P. Boris, C.-C. Cheng, J. T. Mariska, and E. S. Oran, Numerical simulation of cooling coronal flare plasma, *Astrophys. J.* **258**, 373 (1982).
14. C.-C. Cheng, E. S. Oran, G. A. Doschek, J. P. Boris, and J. T. Mariska, Numerical simulations of loops heated to solar flare temperatures. I. Gasdynamics, *Astrophys. J.* **265**, 1090 (1983).
15. C.-C. Cheng, E. S. Oran, G. A. Doschek, J. P. Boris, and J. T. Mariska, Numerical simulations of loops heated to solar flare temperatures. II. X-ray and UV spectroscopy, *Astrophys. J.* **265**, 1103 (1983).
16. C. J. Avro, CHEMSODE: A stiff ODE solver for the equations of chemical kinetics, *Comput. Phys. Commun.* **97**, 304 (1996).
17. R. Lorenzini and L. Passoni, Test of numerical methods for the integration of kinetic equations in tropospheric chemistry, *Comput. Phys. Commun.* **117**, 241 (1999).
18. G. Patnaik and K. Kailasanath, Numerical predictions of the cell-split limit in lean premixed hydrogen–air flames in microgravity, presented at the Joint Meeting of the United States Sections of the Combustion Institute, Washington, DC (March 1999).
19. K. Kailasanath, G. Patnaik, and C. Li, Computational studies of pulse detonation engines: A status report, AIAA Paper 99-2634, presented at the 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Los Angeles (1999).
20. D. R. Mott, *New Quasi-Steady-State and Partial-Equilibrium Methods for Integrating Chemically Reacting Systems.* Ph.D. thesis (University of Michigan, 1999).
21. D. R. Mott and E. S. Oran, *CHEMEQ2: A Solver for the Stiff Differential Equations of Reaction Kinetics*, NRL Memorandum Report (Naval Research Laboratory, Washington, DC), in preparation.
22. J. G. Verwer and M. van Loon, An evaluation of explicit psuedo-steady-state approximation schemes for stiff ODE systems from chemical kinetics, *J. Comput. Phys.* **113**, 347 (1994).
23. J. G. Verwer and D. Simpson, Explicit methods for stiff ODEs from atmospheric chemistry, *Appl. Numer. Math.* **18**, 413 (1995).
24. L. O. Jay, A. Sandu, F. A. Porta, and G. R. Carmichael, Improved quasi-steady-state-approximation methods for atmospheric chemistry integration, *SIAM J. Sci. Comput.* **18**, 182 (1997).
25. K. Radhakrishnan and D. T. Pratt, Fast algorithm for calculating chemical kinetics in turbulent reacting flow, *Combust. Sci. Technol.* **58**, 155 (1988).
26. A. C. Hindmarsh, LSODE and LSODEI, two new initial value ordinary differential equation solvers, *ACM SIGNUM Newslett.* **15**, 10 (1980).
27. A. C. Hindmarsh, ODEPACK: A systemized collection of ODE solvers, in *Scientific Computing*, edited by R. S. Stepleman *et al.* (North-Holland, Amsterdam, 1983), pp. 55–64. [Also UCRL-880-07, Lawrence Livermore Laboratory, Livermore, CA, 1993]
28. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice Hall, Englewood Cliffs, NJ, 1971).
29. K. Radhakrishnan, New integration techniques for chemical kinetic rate equations. I. Efficiency comparison, *Combust. Sci. Technol.* **46**, 59 (1986).
30. F. A. Williams, *Combustion Theory, The Fundamental Theory of Chemically Reacting Systems*, 2nd ed. (Benjamin–Cummings, Menlo Park, CA, 1985).
31. S. H. Lam and D. A. Goussis, The CSP method for simplifying kinetics, *Int. J. Chem. Kinet.* **26**, 461 (1994).

32. M. Frenklach, H. Wang, and M. J. Rabinowitz, Optimization and analysis of large chemical kinetic mechanisms using the solution mapping method—Combustion of methane, *Prog. Energy Combust. Sci.* **18**, 47 (1992).
33. U. Uphoff, D. Hänel, and P. Roth, A grid refinement study for detonation simulation with detailed chemistry, in *Proc. of 6th Int. Conf. Numerical Combustion, New Orleans, March 4–6, 1996.*
34. P. Deuflhard, U. Nowak, and U. Poehle, Scientific Software Group, Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin, 1989.
35. P. Deuflhard, A semi-implicit midpoint rule for stiff systems of ordinary differential equations, *Numer. Math.* **41**, 373 (1983).
36. P. Deuflhard, Order and stepsize control in extrapolation methods, *Numer. Math.* **41**, 399 (1983).
37. P. Deuflhard, Uniqueness Theorems for Stiff ODE Initial Value Problems, Preprint SC-87-3 (Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin, 1987).
38. G. Patnaik, K. J. Laskey, K. Kailasanath, E. S. Oran, and T. A. Brun, *FLIC—A Detailed, Two-Dimensional Flame Model*, NRL Memorandum Report 6555 (Naval Research Laboratory, Washington, DC, 1989).
39. J. P. Boris, Alexandra M. Landsberg, Elaine S. Oran, and John H. Gardner. *LCPFCT—A Flux-Corrected Transport Algorithm for Solving Generalized Continuity Equations*. NRL Memorandum Report 6410-93-7192 (Naval Research Laboratory, Washington, DC, 1993).
40. W. H. Vandevender and K. H. Haskell, The SLATEC mathematical subroutine library, *SIGNUM Newslett.* **17**, 16 (September 1982).