JOURNAL OF COMPUTATIONAL PHYSICS **40**, 167–182 (1981)

On Solving the Stiff ODE's of the Kinetics of Chemically Reacting Gas Flow*

ILKKA KARASALO AND JOHN KURYLO

Lawrence Berkeley Laboratory, University of California, Berkeley, California 94720

Received October 22, 1979; revised February 22, 1980

We study the efficiency of computational methods for the stiff ordinary differential equations of chemical kinetics that arise when the partial differential equations of chemically reacting gas flow are treated by a fractional step technique. In this application, the overhead work associated with evaluating partial derivatives and decomposing matrices for the Newton-like corrector iterations used in most algorithms for stiff ODE's can be eliminated for the most part by keeping in store a small number of suitably chosen copies of the Jacobian matrix, reduced to Hessenberg form to facilitate changes of stepsize and order. Computational results in the case of ignition and propagation of a one-dimensional, premixed laminar flame with different realistic chemical kinetic models are presented to show the reduction of computational work obtained by modifying a modern general-purpose ODE-code in this manner.

1. INTRODUCTION AND SUMMARY

Operator splitting, or fractional step, techniques [26] are used frequently for the numerical simulation of the flow of multicomponent gas mixtures undergoing rapid chemical reactions such as flame ignition and flame propagation [15, 16, 19]. Reasons for the popularity of operator splitting techniques for this class of problems include the following: First, they often offer a reasonable compromise between on one hand the poor storage economy of fully coupled implicit schemes, in particular for problems with multidimensional geometry, and on the other the unacceptably small timesteps needed for stability by explicit schemes. Second, their generally low orders of accuracy are acceptable since the accuracy requirements are in practice rather moderate. Third, they provide a straightforward way of treating different parts of the governing system of partial differential equations by different numerical methods and are easy to implement using standard software for the different fractional steps as building blocks.

In this paper we assume that an operator splitting technique is being applied on the equations of reacting gas flow (see Ref. [25, pp. 2–9]) in such a way that, in particular, heat release and species production due to chemical reactions are

---

* This work was supported by the Engineering, Mathematical and Geosciences Division of the U.S. Department of Energy under contract W-7405-ENG-48.

0021-9991/81/030167–16$02.00/0

computed in a separate fractional "chemistry step." Then, if a detailed model of the chemical kinetics is used, the number of species and reaction steps are likely to be such that the by far dominating portion of the total computational effort is consumed by such chemistry steps. Thus, effective numerical treatment of the chemistry steps is of first-order importance for the overall computational economy of the fractional step algorithm, very much regardless of the choice of numerical methods for other fractional steps.

In the chemistry step the values of the temperature and the concentrations of the reacting species are advanced by solving, for each spatial grid point, an initial value problem for a system of nonlinear ordinary differential equations over a time interval equal to the length of the fractional step. Since this time interval generally exceeds by several orders of magnitude the time constants associated with the most rapidly changing particular solutions of the system of ordinary differential equations, the initial value problems must be treated by methods suitable for "stiff" ordinary differential equations. Straightforward use of a general purpose algorithm for stiff initial value problems would, however, be far from optimal for the following reasons (see also Ref. [18, pp. 14–15]): First, a substantial portion of the computational work would consist of evaluating the Jacobian matrices needed for solving the nonlinear equations arising from the implicit discretization formulas used by the stiff ODE solver. Because initial values and parameters of the ODE's generally vary smoothly in space and because approximately the same set of initial value problems—slightly displaced in space—is solved in adjacent chemistry steps, much of this work is likely to be redundant. Second, as a consequence of the operator splitting, initial rapid transients occur generally at each spatial grid point and each chemistry step. As a rule, these lead to several changes of step size and possibly order during each ODE integration. Most such changes require a redecomposition of the iteration matrix and hence are rather costly at the matrix sizes of interest if standard LU-decomposition techniques are used. Third, even though the exact solutions to the ODE's may be shown to remain positive at all times [3], a general purpose algorithm is likely to introduce negative species concentration values at some point because of rounding and truncation errors. Since particular solutions with sufficiently large negative concentration values may become unstable, the algorithm may therefore fail unless the local error bound is chosen smaller than required for sufficient global accuracy.

Our purpose is to show by computational experiment that the above drawbacks in the treatment of the chemistry steps can largely be eliminated by making some rather simple modifications of the general purpose stiff ODE solver being used. In summary, these modifications are as follows: First, keeping in store a small number of suitably chosen copies of the Jacobian matrix, reduced to Hessenberg form to facilitate changes of stepsize and order, and reusing these copies until corrector convergence becomes slow. Second, allowing step size and order to vary more frequently because step/order changes are now comparatively cheap. Third, keeping concentration values non-negative during prediction and corrector iterations.

We use as a test case the ignition and propagation of a one-dimensional, premixed, laminar flame obtained for three different detailed chemical kinetics models of realistic size. We assume for simplicity pressure to be constant and the diffusion of each species to be governed by a Fick's law (see Ref. [25, p. 11]), neglecting thermal and pressure diffusion, viscosity, and radiative heat transfer. We use a fixed, uniform spatial mesh and choose a straightforward method of lines technique for fractional steps involving spatial derivatives. We remark, that for this simplified test problem, fractional step methods need not necessarily be more favourable than alternative techniques based on fully coupled, implicit discretization formulas [14], possibly combined with adaptive spatial gridding [8, 13]. Our purpose is, however, restricted to improving the treatment of the chemistry steps only, using techniques which do not make use of simplifications of the geometry, the choice of spatial mesh, or the modeling of fluid dynamics and transport phenomena. Hence, in particular, our techniques are immediately applicable also to multidimensional cases, where the use of fully coupled, implicit discretization formulas may become cumbersome.

We used in our test runs a version by Hindmarsh [11] of the stiff ODE solver by Gear (see Ref. [6, pp. 158–168]) and studied the effects of our modifications of this ODE solver on the average CPU time per meshpoint consumed by the various parts of the chemistry step, and the resulting net effect on the overall computational work per mesh point. The main results are contained in Table IV and show that a substantial reduction of the overall computational work, 30–50% in our test cases increasing with the size of the kinetics model, can be gained at the cost of an increase of storage which is affordable and, in particular, remains bounded as the number of mesh points increases.

2. The Test Problem

The partial differential equations of our test problem describe the one-dimensional flow of a chemically reacting multicomponent mixture of ideal gas where a Fick's law is assumed to govern the diffusion of each species and where pressure variations, viscosity, radiative heat transfer, and thermal and pressure viscosity are assumed to be negligible (see Refs. [25, pp. 2–11; 14, p. 413]):

$$\rho_t + (v\rho)_x = 0, \tag{2.1}$$

$$T_t + vT_x = \frac{(\lambda T_x)_x}{\rho C_p} - \frac{\mathbf{h}^T \boldsymbol{\omega}}{\rho C_p} + \frac{T_x \mathbf{C}_p^T D_d \mathbf{Y}_x}{C_p}, \tag{2.2}$$

$$\mathbf{Y}_t + v\mathbf{Y}_x = \frac{\boldsymbol{\omega}}{\rho} + \frac{(\rho D_d \mathbf{Y}_x)_x}{\rho}, \tag{2.3}$$

$$p_0 = R_0 \cdot T \cdot \rho \cdot \sum_{i=1}^{N} \frac{Y_i}{W_i}. \tag{2.4}$$

Equations (2.1)–(2.4) express, respectively, the laws of conservation of mass, energy, and species and the ideal gas law. Here ρ (mass density of gas mixture), v (mass average velocity of gas mixture), T (temperature), and $\mathbf{Y} = (Y_1, \ldots, Y_N)^T$ (mass fractions of the species) are the unknown functions of x (space) and t (time). The remaining quantities in Eq. (2.1)–(2.4) are:

p_0	pressure (assumed constant)
R_0	universal gas constant
$W_i, i = 1, \ldots, N$	molecular weights of the species
λ	thermal conductivity of mixture (assumed constant)
$D_d = \operatorname{diag}(d_1, \ldots, d_N)$	diagonal matrix of binary diffusion coefficients of the species (assumed constant)
$\mathbf{h} = \mathbf{h}(T)$	vector of specific enthalpies per unit mass of the species (known as function of T (see Ref. [9, p. 32]))
$\mathbf{C}_p = \mathbf{C}_p(T) = d\mathbf{h}/dT$	vector of specific heats at constant pressure of the species
$C_p = C_p(\mathbf{Y}, T) = \mathbf{Y}^T \cdot \mathbf{C}_p(T)$	specific heat at constant pressure of gas mixture
$\boldsymbol{\omega} = \boldsymbol{\omega}(\mathbf{Y}, T)$	rates of production of species by chemical reactions (see Eq. (3.1) below).

The gas mixture is contained in a semi-infinite tube in $x \geqslant 0$, closed at $x = 0$ and is initially at rest with the temperature and species mass fraction gradients zero everywhere. The flame is ignited by transferring heat from an external source into the tube at $x = 0$, leading to the following initial and boundary conditions:

$$\begin{aligned} T(x, 0) &= T_0, & x &\geqslant 0, \\ \mathbf{Y}(x, 0) &= \mathbf{Y}_0, & x &\geqslant 0, \\ v(x, 0) &= 0, & x &\geqslant 0, \end{aligned} \tag{2.5}$$

$$\begin{aligned} T_x(0, t) &= a(\theta(t) - T(0, t)), & t &\geqslant 0, \\ T_x(\infty, t) &= 0, & t &\geqslant 0, \\ \mathbf{Y}_x(0, t) &= \mathbf{Y}_x(\infty, t) = \mathbf{0}, & t &\geqslant 0, \\ v(0, t) &= 0, & t &\geqslant 0. \end{aligned} \tag{2.6}$$

T_0 and $\mathbf{Y}_0$ in Eq. (2.5) are such that the mixture is initially at chemical quasi-equilibrium, $\boldsymbol{\omega}(\mathbf{Y}_0, T_0) \approx \mathbf{0}$. The constant a and the function $\theta(t)$ in Eqs. (2.6) are assumed known. The condition $\mathbf{Y}_x(0, t) = \mathbf{0}$ and $v(0, t) = 0$ express the lack of diffusion and mass average velocities, respectively, at the closed end of the tube.

In practice we replace $x = \infty$ with $x = L$ where $L > 0$ is sufficiently large and introduce mesh points $(i\Delta x, j\Delta t)$, $0 \leqslant i \leqslant N_x = L/\Delta x$, $0 \leqslant j$, with associated approximate solution values $\mathbf{Y}_i^j$, T_i^j, v_i^j, and ρ_i^j. The values $\mathbf{Y}_i^{j+1}$, T_i^{j+1}, v_i^{j+1}, and ρ_i^{j+1} are computed from $\mathbf{Y}_i^j$, T_0^j, v_i^j, ρ_i^j, $i = 0, \ldots, N_x$ by taking the following fractional steps in sequence:

1. *Chemistry step.* Solve the $N_x + 1$ systems of $N + 1$ ordinary differential equations

$$\begin{aligned} \frac{d\mathbf{Y}}{dt} &= \frac{\boldsymbol{\omega}(\mathbf{Y}, T)}{\rho}, \\ \frac{dT}{dt} &= -\frac{\mathbf{h}(T_i^j)^T \boldsymbol{\omega}(\mathbf{Y}, T)}{\rho \cdot C_p(\mathbf{Y}_i^j, T_i^j)}, \\ p_0 &= R_0 \cdot T \cdot \rho \cdot \sum_{k=1}^{N} \frac{Y_k}{W_k} \end{aligned} \tag{2.7}$$

with initial values

$$\begin{aligned} \mathbf{Y}(j\,\Delta t) &= \mathbf{Y}_i^j, \\ T(j\,\Delta t) &= T_i^j, \end{aligned} \qquad i = 0, \ldots, N_x,$$

over the time interval $(j\Delta t, (j+1) \cdot \Delta t)$ to obtain

$$\begin{aligned} \mathbf{Y}_i^{j+1,1} &= \mathbf{Y}((j+1)\,\Delta t), \\ T_i^{j+1,1} &= T((j+1)\,\Delta t), \end{aligned} \qquad i = 0, \ldots, N_x. \tag{2.8}$$

The species production rate function $\boldsymbol{\omega}(\mathbf{Y}, T)$ is described in greater detail in Eq. (3.1) below.

2. *Diffusion step.* Solve the N linear tridiagonal systems of $N_x + 1$ ordinary differential equations

$$\frac{d\mathbf{Y}_i}{dt} = \frac{D_d}{\rho_i^j} \frac{(\rho_{i+1}^j + \rho_i^j)(\mathbf{Y}_{i+1} - \mathbf{Y}_i) - (\rho_i^j + \rho_{i-1}^j)(\mathbf{Y}_i - \mathbf{Y}_{i-1})}{2\,\Delta x^2} \tag{2.9}$$

with $\mathbf{Y}_{-1} = \mathbf{Y}_1$, $\mathbf{Y}_{N_x+1} = \mathbf{Y}_{N_x-1}$, $\rho_{-1} = 2\rho_0 - \rho_1$, $\rho_{N_x+1} = 2\rho_{N_x} - \rho_{N_x-1}$, and with initial values

$$\mathbf{Y}_i(j\,\Delta t) = \mathbf{Y}_i^{j+1,1}, \qquad i = 0, \ldots, N_x,$$

over the time interval $(j\Delta t, (j+1)\,\Delta t)$ to obtain

$$\mathbf{Y}_i^{j+1,2} = \mathbf{Y}_i((j+1)\,\Delta t), \qquad i = 0, \ldots, N_x. \tag{2.10}$$

3. *Heat conduction step.* Solve the linear tridiagonal system of $N_x + 1$ ordinary differential equations

$$\begin{aligned} \frac{dT_i}{dt} = \frac{1}{C_p(\mathbf{Y}_i^{j+1,2}, T_i^{j+1,1})} \Bigg\{ & \frac{\lambda}{\rho_i^j} \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} \\ & + \frac{\mathbf{C}_p(T_i^{j+1,1})^T D_d (\mathbf{Y}_{i+1}^{j+1,2} - \mathbf{Y}_{i-1}^{j+1,2})}{2\,\Delta x} \cdot \frac{T_{i+1} - T_{i-1}}{2\,\Delta x} \Bigg\} \end{aligned} \tag{2.11}$$

with $\mathbf{Y}_{-1}^{j+1,2} = \mathbf{Y}_{1}^{j+1,2}$, $\mathbf{Y}_{N_x+1}^{j+1,2} = \mathbf{Y}_{N_x-1}^{j+1,2}$, $T_{N_x+1} = T_{N_x-1}$, $T_1 - T_{-1} = 2\Delta x \cdot \alpha \cdot (\theta(t) - T_0)$, and with initial values

$$T_i(j\,\Delta t) = T_i^{j+1,1}, \qquad i = 0,\dots, N_x,$$

over the time interval $(j\Delta t,\ (j+1)\Delta t)$ to obtain

$$T_i^{j+1,2} = T_i((j+1)\,\Delta t), \qquad i = 0,\dots, N_x. \tag{2.12}$$

4. *Convection step.* Solve the $N+1$ linear tridiagonal systems of $N_x + 1$ ordinary differential equations

$$\left.\begin{aligned} \frac{d\mathbf{Y}_i}{dt} &= -v_i^j \frac{\mathbf{Y}_{i+1} - \mathbf{Y}_{i-1}}{2\,\Delta x}, \\ \frac{dT_i}{dt} &= -v_i^j \frac{T_{i+1} - T_{i-1}}{2\,\Delta x}, \end{aligned}\right. \qquad i = 0,\dots, N_x, \tag{2.13}$$

with $\mathbf{Y}_{N_x+1} = \mathbf{Y}_{N_x-1}$, $T_{N_x+1} = T_{N_x-1}$ (note that $v_0^j = 0$), and with initial values

$$\left.\begin{aligned} \mathbf{Y}_i(j\,\Delta t) &= \mathbf{Y}_i^{j+1,2}, \\ T_i(j\,\Delta t) &= T_i^{j+1,2}, \end{aligned}\right. \qquad i = 0,\dots, N_x,$$

over the time interval $(j\,\Delta t,\ (j+1)\,\Delta t)$ to obtain

$$\left.\begin{aligned} \mathbf{Y}_i^{j+1} &= \mathbf{Y}_i((j+1)\,\Delta t), \\ T_i^{j+1} &= T_i((j+1)\,\Delta t), \end{aligned}\right. \qquad i = 0,\dots, N_x. \tag{2.14}$$

5. *Mass continuity step.* Compute ρ_i^{j+1} using the values $\mathbf{Y}_i^{j+1}$ and T_i^{j+1} in (2.14) and the ideal gas law (2.4) and put

$$\rho_{ti}^{j+1} = \frac{\rho_i^{j+1} - \rho_i^j}{\Delta t}, \qquad i = 0,\dots, N_x. \tag{2.15}$$

Let $\tilde{\rho}_t^{j+1}(x)$ be the cubic spline fitted to the $N_x + 1$ values ρ_{ti}^{j+1} in (2.15) (with side conditions $d^2\tilde{\rho}_t^{j+1}(x)/dx^2 = 0$ at $x = 0$ and $x = L$) and then compute

$$v_i^{j+1} = \frac{1}{\rho_i^{j+1}} \int_0^{i\cdot\Delta x} \tilde{\rho}_t^{j+1}(x)\, dx. \tag{2.16}$$

The truncation error of the above scheme is expected to be $O(\Delta t + \Delta x^2)$ as $\Delta t, \Delta x \to 0$, in good agreement with computational results. The equations (2.9), (2.11), and (2.13) of the diffusion, heat conduction, and convection steps were treated by the implicit trapezoidal rule with stepsize Δt, the length of the fractional step. The above subsplitting into steps 2–5 and the associated discretization formulas and stepsize strategies probably could be replaced by more effective choices; however, our main interest is with the treatment of the Eqs. (2.7) of the chemistry step (see Section 1).

3. The Chemistry Step

The species production rate function $\boldsymbol{\omega}(\mathbf{Y}, T)$ in the differential equations (2.7) is of the form (see Ref. [25, pp. 2–4]):

$$\boldsymbol{\omega}(\mathbf{Y}, T) = D_W(V''^T - V'^T)\{D_{k_f}(T_i^j)\,\mathbf{Z}^{V'} - D_{k_b}(T_i^j)\,\mathbf{Z}^{V''}\}, \tag{3.1}$$

where

$D_W = \operatorname{diag}(W_1, \ldots, W_N)$	diagonal matrix of molecular weights of the species
V', V''	$M \times N$ matrices of (integer) stoichiometric coefficients for reactants and products respectively. (M = number of reactions.)
$D_{k_f}(T), D_{k_b}(T)$	$M \times M$ diagonal matrices of reaction rate constants for forward and backward reactions, respectively. The diagonal elements of $D_{k_f}(T)$ are of the form $K_{fi}(T) = K_{ofi} \cdot T^{\alpha_{fi}} \cdot \exp(-E_{fi}/R_0 T)$, $i = 1, \ldots, M$, where $K_{ofi}\,\alpha_{fi}$ are constants. $D_{k_b}(T)$ is defined similarly.
$\mathbf{Z} = (Z_1, \ldots, Z_N)^T$	vector of concentrations (in moles per unit volume) of the N species, $Z_i = \rho Y_i/W_i$, $i = 1, \ldots, N$.
$\mathbf{Z}^{V'}$	vector of the M products $\prod_{k=1}^{N} Z_k^{V'_{ik}}$, $i = 1, \ldots, M$. $Z^{V''}$ is defined similarly.

Some of the N species, such as nitrogen in our test examples, may occur only as "third bodies" in the reactions and thus give rise to zero rows in the matrix $V''^T - V'^T$ in Eq. (3.1). In such cases, we exclude the corresponding constant Y-components when solving the system Eq. (2.7) numerically, thus reducing its dimensionality to $NA + 1$ where $NA \leqslant N$ is the number of remaining (non-inert) species. By putting $\mathbf{y} = (Y_1, \ldots, Y_{NA}, T)^T$ and defining $\mathbf{f}(\mathbf{y})$ accordingly, Eqs. (2.7) and (3.1) assume the normal form of an autonomous system of ordinary differential equations

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}). \tag{3.2}$$

Unless Δt, the time step of the fractional step algorithm, is chosen much smaller than needed for accuracy, the product $-\operatorname{Re} \lambda \cdot \Delta t$ will be much larger than one (maybe 10^3 or larger in our test examples) for some eigenvalues λ to the Jacobian matrix $J = J(\mathbf{y}) = \partial \mathbf{f}/\partial \mathbf{y}$. Hence Eq. (3.2) must be treated by methods suitable for stiff ordinary differential equations. Our test runs were made using a version by Hindmarsh [11] of the widely used algorithm by Gear (see Ref. [6, pp. 158–168]) with, however, the following modifications designed to make advantage of particular features of the chemistry step problem, cf. Section 1:

(1) We subdivide the temperature range of interest into K subintervals by introducing constants $T_1 < T_2 < \cdots < T_{K-1}$ (we used $K = 10$ in our test runs) and reserve storage space for K real $(NA + 1) \times (NA + 1)$ matrices and K integer vectors with $NA + 1$ components. Whenever a Jacobian matrix is evaluated, we reduce it to Hessenberg form by stabilized elementary similarity transformations (see Refs. [24, p. 353; and 5]), i.e., we compute matrices P, L, and H such that

$$J(\mathbf{y}) = PLHL^{-1}P^T, \tag{3.3}$$

where P is a permutation matrix, the elements h_{ij} of H are zero for $i > j + 1$, and the elements l_{ij} of L are nonzero only for $i = j = 1$ or $i \geqslant j > 1$ with $l_{ii} = 1$, $i = 1, \ldots, NA + 1$. Furthermore, by the requirement that the number of atoms of each chemical element be conserved in each reaction

$$\operatorname{rank} J = \operatorname{rank} H \leqslant \mathrm{NA} + 1 - \mathrm{NE} \tag{3.4}$$

independently of $\mathbf{y}$, where NE is the number of different chemical elements appearing in the NA reacting species. In all our test examples, when choosing the permutation matrix P by the simple pivoting strategy used in Ref. [24, p. 354], this rank deficiency of H showed by the last NE rows of H being zero to working accuracy, thus allowing for premature termination of the decomposition (3.3).

The value of the temperature component of $\mathbf{y}$ in Eq. (3.3) will be in one of the temperature subintervals defined above, say the kth, and we store the matrix P and the matrices L and H into the kth of the reserved storage areas. Subsequently, whenever the temperature component of $\mathbf{y}$ in Eq. (3.2) is in the kth subinterval, we use these matrices P, L, and H by Eq. (3.3) as the approximate Jacobian matrix needed in the Newton-like corrector iterations updating P, L, and H by reevaluating and reducing $J(\mathbf{y})$ only when corrector convergence becomes unacceptably slow. In all our test runs such updates, summed over all temperature intervals, occurred on the average less than once in every five initial value problems, thus keeping the overhead work for the decompositions (3.3) acceptably small (see Tables II–IV).

The linear system of equations appearing in the corrector iterations is of the form shown in Ref. [6, pp. 216–217],

$$(I - \alpha_p hJ(\mathbf{y}))\,\mathbf{x} = \mathbf{b}, \tag{3.5}$$

where h is the current step size, α_p is a number dependent of the order p, and I is the identity matrix. By Eq. (3.3), Eq. (3.5) becomes

$$PL(I - \alpha_p hH)L^{-1}P^T\mathbf{x} = \mathbf{b} \tag{3.6}$$

which can be solved by back substitutions after LU decomposition of only the $(\mathrm{NA} + 1 - \mathrm{NE}) \times (\mathrm{NA} + 1 - \mathrm{NE})$ upper left corner of the Hessenberg matrix

$I - \alpha_p hH$ (see the comment after Eq. (3.4)). The overhead work for this LU decomposition, which has to be redone whenever the product $\alpha_p \cdot h$ changes significantly, is thus roughly $(\mathrm{NA}+1-\mathrm{NE})^2/2$ operations to be compared to the roughly $(\mathrm{NA}+1)^3/3$ operations needed for LU decomposition of Eq. (3.5) directly (here "operation" stands for one addition and one multiplication or division in floating point arithmetic). The back substitutions in Eq. (3.6) require roughly $3(\mathrm{NA}+1)^2/2$ operations as compared to $(\mathrm{NA}+1)^2$ in the direct LU-decomposition case. However, step/order changes requiring LU decomposition are sufficiently frequent to make the net gain substantial (see Tables III and IV).

(2) We relax the stepsize selection strategy so that a step increase at order p is attempted whenever p steps have been taken with constant step size, as compared to $p+2$ steps in the original strategy (see Ref. [11]). We further permit step increase by up to a factor of 10^4 at all times (instead of only initially as in Ref. [11]), except for the first increase after a stepsize reduction. The risk of these modifications introducing unwanted error growth caused by too frequent stepsize changes [1, 7, 10, 11] is expected to be small because the order is generally low due to the moderate accuracy requirements, the total number of integration steps per initial value problem is on the average very small, and the step size is as a rule non-decreasing in each initial value problem (cf. Refs. [1, p. 102; 10, p. 133]). In our test runs, these modifications were found to reduce the average number of integration steps per initial value problem considerably (see Table III), mainly by using fewer steps in the transients where in general the stepsize is initially several orders of magnitude smaller than the interval of integration.

(3) We modify the prediction–correction scheme to avoid introducing possibly harmful negative species mass fraction values [4] by rounding or truncation errors in the following way: Let $\mathbf{y}^{\mathrm{pr}}$ be the predicted solution vector computed by polynomial extrapolation. Then we use $\mathbf{y}^0 = (y_0^1, \ldots, y_{\mathrm{NA}+1}^0)$ as starting point for the corrector iterations, where

$$y_i^0 = \max\{y_i^p, 0\}, \qquad i = 1, \ldots, \mathrm{NA}+1, \tag{3.7}$$

and subsequently adjust any negative components occurring in $\mathbf{y}$ to zero after each corrector iteration. We do, however, use the actual predicted value $\mathbf{y}^{\mathrm{pr}}$ when computing the accumulated correction vector needed for the local error estimate and for updating the Nordsieck matrix after corrector convergence. We note that, unlike the original scheme, this modified corrector procedure no longer necessarily conserves global linear invariants of the solution, such as element masses, to within rounding error tolerance independent of truncation errors from the time discretization and the finite number of corrector iterations. However, the local errors that may occur in such linear invariants are still within the requested local error tolerance which, in principle, is quite satisfactory. Furthermore, it can in fact be shown that our corrector procedure may accumulate error (other than rounding error) in linear invariants to (3.2) only when negative components happen to be present in $\mathbf{y}$ after the last corrector iteration preceding corrector convergence in an accepted step. In our test runs, when using the predictor (3.7), negative corrected y-components occurred only occasionally and the global error of the linear invariants was always several orders of magnitude below the local error tolerance.

4. COMPUTATIONAL RESULTS

Our test runs of the initial boundary value problems (2.1)–(2.6) were performed with three different models of the chemical kinetics of the gas mixture. Table I shows the source references used for obtaining the details of these models, together with the number of species and reactions appearing in each model. The entries of the last column of Table I assume that the formalism of Eq. (3.1) has been extended so as to allow the sum of all species concentrations, $Z_{N+1} = \sum_{i=1}^{N} Z_i$, to appear as a third body in the reactions. In the actual coding of Eq. (3.1) we include this extension by augmenting the matrices V' and V'' (which are stored row-wise in compact form) by identical $(N+1)$th columns. We note that this extension is done for the practical purpose of reducing computational work and storage requirements only, since such "third body" reactions could equivalently be split into N simpler reactions covered by the formalism of (3.1).

The thermodynamic functions $\mathbf{h}(T)$ and $\mathbf{C}_p(T) = d\mathbf{h}/dT$ were given as piecewise polynomials in T (see Ref. [9, p. 32]). The constant pressure p_0 was chosen to be 1.013×10^6 dyn/cm^2 (=1 atm). The choice of values for the thermal conductivity λ and the diffusion coefficients $d_1, \ldots, d_n$ is not crucial for our results, and we simply put $d_1 = \cdots = d_n = d$, choosing the constants d (5 cm^2/sec) and λ (5.10^3 dyn/sec) so as to obtain burning velocities, species concentration profiles, and flame thicknesses roughly comparable to those in Refs. [15, 22] in the steady-state flame. The function $\theta(t)$ in the boundary condition Eq. (2.6) for the temperature was chosen to be

$$\theta(t) = \begin{cases} T_s + (T_e - T_s) \cdot t/t_r, & 0 \leqslant t \leqslant t_r, \\ T_e, & t \geqslant t_r \end{cases} \tag{4.1}$$

with $t_r = 10^{-4}$ sec, $T_s = 300$ K, $T_e = 2400$ K (for the H_2–air model) and 2700 K (for the CH_4–air models). The constant α was set to 200 cm^{-1} in all cases. The boundary

TABLE I

Source References and Dimensionality Data for Kinetics Models Used in Test Runs

Test Case No.	Kinetics Model [Refs.]	No. of Species, N	No. of Reacting Species, NA	No. of Reactions, M
1	H_2–air [2, 15, 17]	8	7	13
2	CH_4–air [20, 23]	19	18	56
3	CH_4–air [21, 23]	26	25	86

conditions to the right were given at $L = 1$ cm and the stepsizes $\Delta x = 5 \times 10^{-3}$ cm, $\Delta t = 2.5 \times 10^{-6}$ sec were found to give sufficient accuracy for our purposes. At these stepsizes the bounds $\Delta t \cdot \|S_D\|_\infty \lesssim 2$, $\Delta t \|S_H\| \lesssim 1.8$, and $\Delta t \cdot \|S_C\|_\infty \lesssim 0.6$ hold in all test cases, where S_D, S_H, and S_C are the tridiagonal matrices in Eq. (2.9), (2.11), and (2.13), respectively. The time interval was $0 \leqslant t \leqslant 0.5$ msec in the H_2–air case and $0 \leqslant t \leqslant 0.6$ msec in the CH_4–air cases, corresponding roughly to three times the length of the flame ignition period. In all cases, the local error bound for the initial value problems, Eq. (2.7), in the chemistry step was chosen to $\varepsilon = 10^{-3}$, based on repeated runs with successively decreasing ε until visual convergence of graphs of temperature, flow velocity, and species concentration profiles was obtained. The norm of the local error is estimated in Ref. [11] by

$$\|\text{local error}\|^2 = \beta_p \sum_{i=1}^{\mathrm{NA}+1} \left(\frac{e_i}{y_{\max,i}}\right)^2, \tag{4.2}$$

where β_p is a number dependent of the order p, $\mathbf{e} = (e_1, \ldots, e_{\mathrm{NA}+1})^T$ is the difference between the corrected and predicted solution values, and $y_{\max,1}, \ldots, y_{\max,\mathrm{NA}+1}$ are weights to be chosen by the user. We update these weights after each integration step in Eq. (3.2) to be the maximum values of $y_1, \ldots, y_{\mathrm{NA}+1}$ in any solution to Eq. (3.2) computed since $t = 0$.

In Table II, we list some of then main subalgorithms needed for solving the initial value problems of the chemistry step, and show the CPU times required on a Digital VAX 11/780 computer for executing each of these once.

Table III shows the average numbers of integration steps, function evaluations, and

TABLE II

CPU Times (in Milliseconds) per Execution of Some Algorithms Needed in the Chemistry Step

	Case 1	Case 2	Case 3
Evaluation of $\mathbf{f}(\mathbf{y})$ in Eqs. (3.2), (2.7), (3.1)	2.7	9.2	13.8
Evaluation of $J(\mathbf{y}) = \partial\mathbf{f}/\partial\mathbf{y}$	12.2	54.3	92.2
LU decomposition of $I - \alpha_p hJ(\mathbf{y})$ in Eq. (3.5)	5.5	47.3	112
Solution of Eq. (3.5) using LU factors	1.3	5.8	10.6
Reduction of $J(\mathbf{y})$ to Hessenberg form as in Eq. (3.3)	7.7	87.6	216
LU decomposition of $I - \alpha_p hH$ in Eq. (3.6)	1.4	6.4	12.3
Solution of Eq. (3.6) using LU factors	1.8	9.4	18.0
Evaluation of $\mathbf{h}(T_i^j)$, $\mathbf{C}_p(T_i^j)$, and C_p in Eqs. (2.2), (2.3), (2.7)	1.2	2.8	4.6
Evaluation of $D_{k_f}(T_i^j)$ and $D_{k_b}(T_i^j)$ in Eq. (3.1)	2.2	10.0	15.4

TABLE III

Average Number of Integration Steps, Function Evaluations, and Matrix Handling Operations per Initial Value Problem (2.7), (3.2) Based on Total Work in $0 \leqslant t \leqslant t_{\mathrm{av}1}$ (First Row in Row Groups) and in $0 \leqslant t \leqslant t_{\mathrm{av}2}$ (Second Row in Row Groups)

	Case 1 $t_{\mathrm{av}1} = 0.2$ msec $t_{\mathrm{av}2} = 0.5$ msec		Case 2 $t_{\mathrm{av}1} = 0.2$ msec $t_{\mathrm{av}2} = 0.6$ msec		Case 3 $t_{\mathrm{av}1} = 0.2$ msec $t_{\mathrm{av}2} = 0.6$ msec	
	Original ODE Solver	Modified ODE Solver	Original ODE Solver	Modified ODE Solver	Original ODE Solver	Modified ODE Solver
Integration steps	13.3	10.3	14.0	11.2	16.9	10.7
	8.9	7.0	12.8	9.8	17.5	10.5
Evaluations of **f**(y)	19.3	18.1	18.9	18.2	21.6	17.0
	14.0	11.9	16.4	14.9	21.7	16.3
Evaluations of J(y)	4.7	0.19	4.5	0.21	5.6	0.20
	3.9	0.08	4.5	0.16	5.9	0.20
Reductions of J(y) to Hessenberg form	—	0.19	—	0.21	—	0.20
	—	0.08	—	0.16	—	0.20
LU decompositions	4.7	7.1	4.5	7.2	5.6	7.3
	3.9	5.2	4.5	7.7	5.9	7.7
Solutions of decomposed linear system	18.3	17.1	17.9	17.2	20.6	16.0
	13.0	10.9	15.4	13.9	20.7	15.3

TABLE IV

Average CPU Times per Mesh Point and Fractional Step Cycle, Based on Average Work as Shown in Table 3

	Case 1		Case 2		Case 3	
	Original ODE Solver	Modified ODE Solver	Original ODE Solver	Modified ODE Solver	Original ODE Solver	Modified ODE Solver
Evaluation of $\mathbf{f}(\mathbf{y})$	52	49	174	167	298	235
	38	32	151	137	299	225
Solution of decomposed linear system	24	31	104	162	218	289
	17	20	89	131	220	275
Evaluation of $J(\mathbf{y})$	57	2	245	12	515	18
	48	1	244	9	544	18
Matrix decomposition and reduction	26	11	213	65	626	133
	21	8	214	63	662	138
Evaluation of parameters $\mathbf{h}, \mathbf{C}_p, D_{k_f}$, and D_{k_b}	3	3	13	13	20	20
	3	3	13	13	20	20
Execution of ODE solver and other overhead	50	53	169	162	228	198
	31	32	128	132	156	196
Chemistry step, total	212	149	918	581	1905	893
	158	96	839	488	1901	872
Other fractional steps	5	5	12	12	37	37
	5	5	12	12	37	37
All fractional steps	217	154	930	593	1942	930
	163	101	851	500	1938	909

matrix handling operations per mesh point, based on the total work spent in the chemistry steps in $0 \leqslant t \leqslant t_{\mathrm{av}}$, for two different values of t_{av} for each test case. The entries in the "modified ODE solver" columns were obtained using the algorithm in Ref. [11] with the three modifications described in Section 3 above. The entries under "original ODE solver" are shown for comparison and were obtained using the original algorithm of Ref. [11] with, however, the prediction–correction scheme modified as described in Section 3. Our attempts to use Ref. [11] without any modification of this kind failed in all three cases at the chosen local error tolerance because of introduction of negative mass fraction values leading eventually to instability.

The entries in row groups 7 and 8 of Table IV are the average CPU times in milliseconds per mesh point and fractional step cycle spent in the chemistry step and in all other fractional steps, respectively. The entries in row group 9 are the sums of corresponding entries in row groups 7 and 8 and hence show the average CPU time required per meshpoint and fractional step cycle by the overall fractional step algorithm. Row groups 1–6 show a breakdown of the entries in row group 7 into the average CPU times per mesh point spent in the main subalgorithms of the chemistry step.

We remark, that the ODE solvers were actually applied only at mesh points where the temperature exceeded $T_0 + 0.5$ K (see Eq. (2.5)), otherwise approximating the solution of (3.2) to be constant. The averages shown in Tables III and IV are based on the subset of mesh points where the ODE solvers were applied (i.e., mesh points in and behind the flame), and the CPU times for other fractional steps as shown in Table IV were normalized accordingly. Further, the first t_{av} value was chosen to coincide roughly with the end of the flame ignition period (0.2 msec in all cases), and hence the entries in the first rows in the row groups in Tables III and IV are averages based on work at mesh points with rapid chemistry only. At the second t_{av} value (0.5 msec in case 1 and 0.6 msec in cases 2 and 3), the average work per mesh point has decreased due to the growing number of mesh points in the zone of slow-chemistry combustion products behind the flame. This decrease would, of course, vanish if the number of such mesh points were limited by adaptive gridding techniques, for which case the averages at the first t_{av} value should be more representative.

5. CONCLUDING REMARKS

We stress, first, the importance of preventing instability at relaxed local error tolerances in the chemistry step by some simple device like our modification 3 (see Section 3). In addition, Tables III and IV indicate the following effects due to our modifications 1 and 2 in the test cases:

(1) Roughly 85% reduction of the CPU time spent on Jacobian evaluations and matrix decompositions (due to modification 1).

(2) Reduction of the number of integration steps by 20–40% (modification 2) counteracted, however, by perhaps 10–25% slower corrector convergence rate and more time-consuming backsolves (modification 1), resulting in a net change of the CPU time spent on function evaluations and solutions of factorized linear systems in the range −3% to +12%.

(3) Up to 25% increase of the CPU time spent in the ODE solver for step size and order selection and other overhead (modification 2).

The net effect of modifications 1 and 2 is a reduction of the overall CPU time for the fractional step algorithm by 29–53%, increasing with the size of the chemical kinetics model as the overhead for Jacobian evaluation and matrix decomposition grows dominant. The techniques used to gain these reductions are restricted to the handling of the ODE's of the chemistry step using information at the current mesh point only, and are thus immediately applicable to operator splitting methods for more general cases of reacting gas flow, including multidimensional cases with more detailed modeling of the fluid dynamics or the transport processes and with more sophisticated spatial gridding techniques.

Our choice of strategy for selecting and storing the Jacobian matrices (see modification 1 in Section 3) by using the value of the temperature only was motivated by the strongly nonlinear form of the reaction rate constants in Eq. (3.1) as functions of temperature. While this strategy could be applied in general, more efficient strategies using, in addition, local species concentration values and more detailed a priori knowledge of the overall flow field, could probably be found in any particular case. For example, in a diffusion flame, a certain temperature range may occur at multiple space locations with essentially different species concentration vectors (see, e.g., Ref. [15, p. 2538]), and there the above strategy would probably require reevaluation of Jacobians associated with such a temperature range at least once per chemistry step. A slightly more elaborate strategy, probably better suited to this situation, could then be, e.g., to choose between two sets of temperature subintervals based on the local ratio between the fuel and oxidizer concentrations.

REFERENCES

1. R. K. BRAYTON, F. G. GUSTAVSON, AND G. D. HACHTEL, *IEEE Proc.* **60** (1972), 98–108.
2. N. J. BROWN-WILLIAMSON, private communication.
3. L. EDSBERG, The Royal Institute of Technology, Stockholm, TRITA-NA-7505, 1975.
4. L. EDSBERG, Numerical methods for mass action kinetics, *in* "Numerical Methods for Differential Systems" (L. Lapidus and W. E. Schiesser, Eds.), pp. 181–195, Academic Press, New York, 1976.
5. W. H. ENRIGHT, Department of Computer Science, University of Toronto, Technical Report 98, July 1976.
6. C. W. GEAR, "Numerical Initial Value Problems in Ordinary Differential Equations," Prentice–Hall, Englewood Cliffs, N.J., 1971.
7. C. W. GEAR AND K. W. TU, *SIAM J. Numer. Anal.* **11** (1974), 1025–1043.
8. R. J. GELINAS, S. DOSS, AND K. MILLER, SIAM 1979 Fall Meeting, Denver, Colo., November 1979.

9. S. Gordon and B. J. McBride, National Aeronautics and Space Administration, Washington, D.C., NASA SP-273, 1971.
10. G. K. Gupta and C. S. Wallace, *Math. Comp.* **33** (1979), 125–138.
11. A. C. Hindmarsh, Lawrence Livermore Laboratory, UCID-30001, Rev. 3, 1974.
12. F. T. Krogh, *SIAM J. Numer. Anal.* **10** (1973), 949–965.
13. C. M. Lund, Lawrence Livermore Laboratory, UCRL 52504, 1978.
14. S. B. Margolis, *J. Computational Phys.* **27** (1978), 410–427.
15. J. A. Miller and R. J. Kee, *J. Phys. Chem.* **81** (1977), 2534–2542.
16. A. W. Rizzi and H. E. Bailey, *AIAA J.* **14** (1966), 621–628.
17. R. Schefer and F. Robben, Lawrence Berkeley Laboratory Report LBL-6842, 1977.
18. L. F. Shampine and C. W. Gear, *SIAM Rev.* **21** (1979), 1–17.
19. P. D. Thomas and K. H. Wilson, *AIAA J.* **14** (1976), 629–636.
20. C. K. Westbrook et al., *J. Phys. Chem.* **81** (1977), 2542–2554.
21. C. K. Westbrook and F. L. Dryer, *Combust. Sci. Technol.* **20** (1979), 267.
22. C. K. Westbrook and F. L. Dryer, Lawrence Livermore Laboratory, UCRL-82717, 1979.
23. C. K. Westbrook, private communication.
24. J. H. Wilkinson, "The Algebraic Eigenvalue Problem," Oxford Univ. Press, London, 1965.
25. F. A. Williams, "Combustion Theory," Addison–Wesley, Reading, Mass., 1965.
26. N. N. Yanenko, "The Method of Fractional Steps," Springer, New York, 1961.