

Lawrence Berkeley National Laboratory

Recent Work

Title

ON SOLVING THE STIFF ODE'S OF THE KINETICS OF CHEMICALLY REACTING GAS FLOW

Permalink

<https://escholarship.org/uc/item/0xz0j2z8>

Author

Karasalo, I.

Publication Date

1979-08-01



Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA, BERKELEY, CA

Physics, Computer Science & Mathematics Division

To be submitted for publication

ON SOLVING THE STIFF ODE'S OF THE KINETICS OF
CHEMICALLY REACTING GAS FLOW

Ilkka Karasalo and John Kurylo

August 1979

RECEIVED
LAWRENCE
BERKELEY LABORATORY

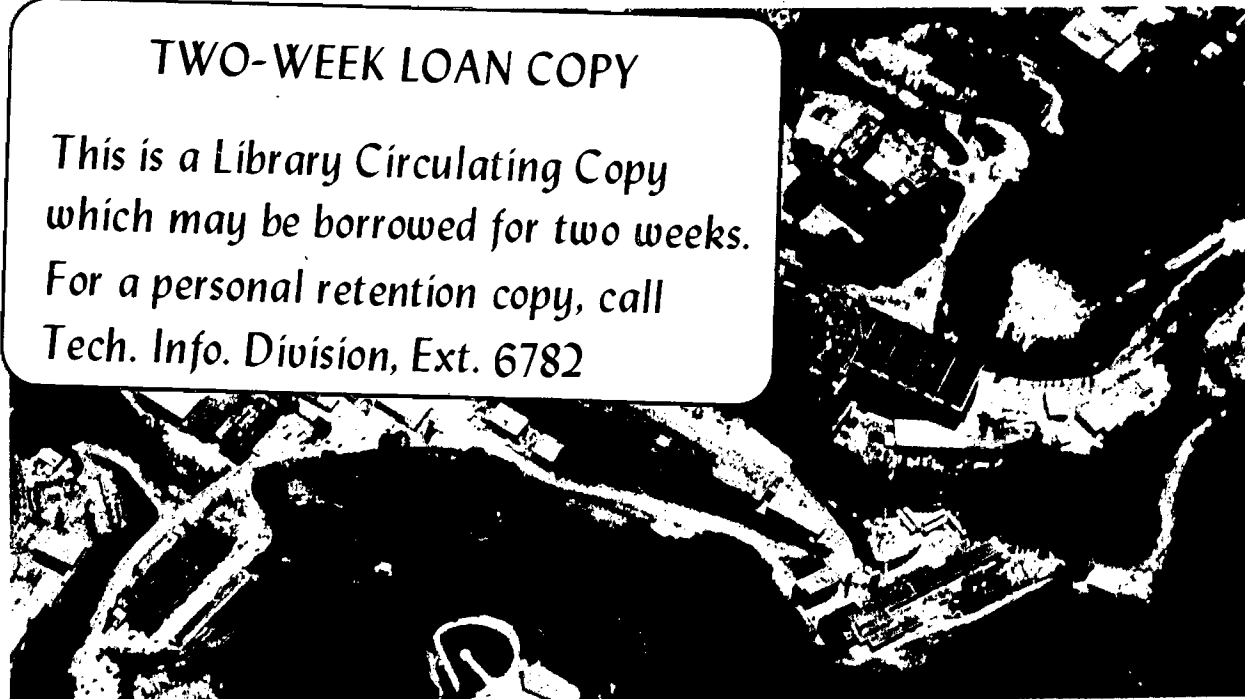
NOV 16 1979

LIBRARY AND
DOCUMENTS SECTION

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.*

*For a personal retention copy, call
Tech. Info. Division, Ext. 6782*



LBL-9069C.2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

ON SOLVING THE STIFF ODE'S OF THE KINETICS OF
CHEMICALLY REACTING GAS FLOW

Ilkka Karasalo and John Kurylo

Physics, Computer Science, and Mathematics Division
Lawrence Berkeley Laboratory, University of California
Berkeley, California 94720

ABSTRACT

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 suitable number of different 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.

This work was supported by the Physics, Computer Science and Mathematics Divisions of the U. S. Department of Energy under contract No. W-7405-ENG-48.

INTRODUCTION AND SUMMARY

Operator splitting, or fractional step, techniques²⁴ 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.^{13,14,17} Reasons for the popularity of operator splitting techniques for this class of problems include the following:

1. They often offer a reasonable compromise between the poor storage economy of fully coupled implicit schemes and the unacceptably small time steps required for stability by explicit schemes.
2. They provide a convenient way for treating different parts of the governing system of partial differential equations by different discretization formulas and stepsize strategies to improve computational performance.
3. Their generally low orders of accuracy are acceptable since the accuracy requirements in practice are moderate.

In this paper we study the computational efficiency of operator splitting techniques when applied on the equations of reacting gas flow (See Ref. 23, p. 2-9) in such a way that, in particular, heat release and species production due to chemical reactions are computed in a separate fractional step. In such a fractional step, the values of the temperature and concentrations of the reacting species are advanced by

solving, for each spatial gridpoint, an initial value problem for a system of 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 solution for the system of ordinary differential equations, the initial value problems must be treated by methods suitable for "stiff" ordinary differential equations. However, straightforward use of general purpose algorithms for stiff initial value problems would be far from optimal here, partly for the following reasons (See also Ref. 16, pp. 14-15):

1. A substantial portion of the computational work consists 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 fractional steps, much of this work is likely to be redundant.

2. As a consequence of the operator splitting, initial rapid transients occur generally at each spatial gridpoint and each fractional "chemistry step". As a rule, these lead to several changes of stepsize and possibly order during each ODE-integration. Most such changes require a re-decomposition of the iteration matrix and hence are rather costly at the matrix sizes of interest if standard LU-decomposition techniques are used.

3. 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 truncation and rounding 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 is required by the global accuracy requirements.

In this paper, we present computational results showing that the above drawbacks can largely be eliminated by making some rather simple modifications of the general purpose stiff ODE algorithm being used. In summary, these modifications consist of:

1. Keeping in store a suitable number of different copies of the Jacobian matrix reduced to Hessenberg form and reusing these copies until corrector convergence becomes slow.
2. Allowing stepsize and order to vary more frequently because step/order changes are now comparatively cheap.
3. Keeping concentration values non-negative during prediction and corrector iterations.

We have used for our test runs a version by Hindmarsh¹⁰ of the algorithm by Gear (See Ref. 6, pp. 158-168), and studied as a test case the ignition and propagation of a one-dimensional, constant pressure, laminar flame obtained for three different realistic chemical kinetics models of different fuel-air mixtures. We present tables showing the effects of our modifications on the average CPU-time per meshpoint

consumed by the fractional step algorithm, in particular by various parts of the fractional "chemistry step".

The main results of this study are contained in Table 4 and show that a substantial reduction, approximately 40-50% in our test cases, of the overall computational work can be obtained by our modifications. This work reduction is gained at the cost of an affordable (in particular bounded as the number of spatial gridpoints increases) increase of the auxiliary storage needed. We note that similar results also may be expected to hold in cases with multidimensional geometry and less restrictive assumptions on the fluid dynamics.

TEST EXAMPLES: EQUATIONS AND THE FRACTIONAL STEPS

The partial differential equations used in our test runs describe the one-dimensional flow of a chemically reacting multicomponent mixture of ideal gas where pressure variations, viscosity, radiative heat transfer, and thermal and pressure diffusion are assumed to be negligible and where a Fick's law (See Ref. 23, p. 11), is assumed to govern the diffusion of each species, (See Ref. 23, pp. 2-9 and Ref. 12, p. 413)

$$\rho_t + (v\rho)_x = 0 \quad (2.1)$$

$$T_t + vT_x = \frac{(\lambda T_x)_x}{\rho C_p} - \frac{h^T \omega}{\rho C_p} + \frac{T_x C_p^T D_d Y_x}{C_p} \quad (2.2)$$

$$\underline{Y}_t + v\underline{Y}_x = \frac{\omega}{\rho} + \frac{(\rho D_d \underline{Y}_x)_x}{\rho} \quad (2.3)$$

$$p_o = R_o \cdot T \cdot \rho \cdot \sum_{i=1}^N \frac{Y_i}{W_i} \quad (2.4)$$

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 $\underline{Y} = (Y_1, \dots, Y_N)^T$ (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
$W_i, i = 1, \dots, N$	molecular weights of the species
λ	thermal conductivity of mixture (assumed constant)
$D_d = \text{diag}(d_1, \dots, d_N)$	diagonal matrix of binary diffusion coefficients of the species (assumed constant)
$\underline{h} = \underline{h}(T)$	vector of specific enthalpies per unit mass of the species (known as function of T (See Ref. 8, p. 32))
$\underline{C}_p = \underline{C}_p(T) = \frac{dh}{dT}$	vector of specific heats at constant pressure of the species
$C_p = C_p(\underline{Y}, T) = \underline{Y}^T \cdot \underline{C}_p(T)$	specific heat at constant pressure of gas mixture
$\underline{\omega} = \underline{\omega}(\underline{Y}, T)$	rates of production of species by chemical reactions, see Eq. (3.1) below.

The gas mixture is contained in a semi-infinite tube in $x \geq 0$, closed at $x = 0$ and is initially at rest with zero temperature- and species mass fraction gradients 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:

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

$$\begin{aligned}
 T_x(0, t) &= \alpha (\theta(t) - T(0, t)), \quad t \geq 0 \\
 T_x(\infty, t) &= 0, \quad t \geq 0 \\
 \underline{Y}_x(0, t) &= \underline{Y}_x(\infty, t) = \underline{0}, \quad t \geq 0 \\
 v(0, t) &= 0, \quad t \geq 0
 \end{aligned}
 \tag{2.6}$$

T_0 and \underline{Y}_0 in Eq. (2.5) are such that the mixture is initially at chemical quasi-equilibrium, $\underline{\omega}(\underline{Y}_0, T_0) \approx \underline{0}$. The constant α and the function $\theta(t)$ in Eq. (2.6) are assumed known. The condition $\underline{Y}_x(0, t) = \underline{0}$ 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 meshpoints $(i \cdot \Delta x, j \cdot \Delta t)$, $0 \leq i \leq N_x = L/\Delta x$, $0 \leq j$, with associated approximate solution values \underline{Y}_i^j , T_i^j , v_i^j and ρ_i^j . The values \underline{Y}_i^{j+1} , T_i^{j+1} , v_i^{j+1} and ρ_i^{j+1} are computed from \underline{Y}_i^j , T_0^j , v_i^j , ρ_i^j , $i = 0, \dots, N_x$ by taking the following fractional steps in sequence:

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

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

with initial values

$$\begin{aligned} \underline{Y} (j \cdot \Delta t) &= \underline{Y}_i^j \\ T (j \cdot \Delta t) &= T_i^j \end{aligned} \quad , \quad i = 0, \dots, N_x$$

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

$$\begin{aligned} \underline{Y}_i^{j+1,1} &= \underline{Y} ((j+1) \cdot \Delta t) \\ T_i^{j+1,1} &= T ((j+1) \cdot \Delta t) \end{aligned} \quad , \quad i = 0, \dots, N_x \quad (2.8)$$

The species production rate function $\omega (\underline{Y}, T)$ 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

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

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

$$\underline{Y}_i (j \cdot \Delta t) = \underline{Y}_i^{j+1,1}, \quad i = 0, \dots, N_x,$$

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

$$\underline{Y}_i^{j+1,2} = \underline{Y}_i ((j+1) \Delta t), \quad i = 0, \dots, N_x \quad (2.10)$$

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

$$\frac{dT_i}{dt} = \frac{1}{C_p(\underline{Y}_i^{j+1,2}, T_i^{j+1,1})} \left\{ \frac{\lambda}{\rho_i^j} \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + \frac{C_p(T_i^{j+1,1})^T D_d(\underline{Y}_{i+1}^{j+1,2} - \underline{Y}_{i-1}^{j+1,2})}{2 \Delta x} \cdot \frac{T_{i+1} - T_{i-1}}{2 \Delta x} \right\} \quad (2.11)$$

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

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

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

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

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

$$\frac{d\underline{Y}_i}{dt} = -v_i^j \frac{\underline{Y}_{i+1} - \underline{Y}_{i-1}}{2 \Delta x}, \quad i = 0, \dots, N_x, \quad (2.13)$$

$$\frac{dT_i}{dt} = -v_i^j \frac{T_{i+1} - T_{i-1}}{2 \Delta x}$$

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

$$\underline{Y}_i(j \cdot \Delta t) = \underline{Y}_i^{j+1,2}, \quad i = 0, \dots, N_x,$$

$$T_i(j \cdot \Delta t) = T_i^{j+1,2}$$

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

$$\begin{aligned} \underline{Y}_i^{j+1} &= \underline{Y}_i((j+1) \cdot \Delta t) \\ T_i^{j+1} &= T_i((j+1) \cdot \Delta t) \end{aligned} \quad , \quad i = 0, \dots, N_x. \quad (2.14)$$

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

$$\rho_{t \ i}^{j+1} = \frac{\rho_i^{j+1} - \rho_i^j}{\Delta t} \quad , \quad i = 0, \dots, N_x. \quad (2.15)$$

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

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

The truncation error of the above scheme is expected to be $O(\Delta t + \Delta x^2)$ as $\Delta t, \Delta x \rightarrow 0$, 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 step-size 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 equations (2.7) of the chemistry step (which will consume by far the largest portion of the computational work anyway (See Table 4 below)).

THE CHEMISTRY STEP

The species production rate function $\underline{\omega}(\underline{Y}, T)$ in the differential equations (2.7) is of the form (See Ref. 23, pp. 2-4):

$$\underline{\omega}(\underline{Y}, T) = D_W^{-1} (V''^T - V'^T) \left\{ D_{k_f}(T_i^j) \underline{Z}^{V'} - D_{k_b}(T_i^j) \underline{Z}^{V''} \right\}, \quad (3.1)$$

where

$$D_W = \text{diag}(W_1, \dots, W_N)$$

diagonal matrix of molecular weights of the species

$$V', V''$$

M×N matrices of (integer) stoichiometric coefficients for reactants and products respectively. (M = number of reactions).

$$D_{k_f}(T), D_{k_b}(T)$$

M×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 $K_{f_i}(T) = K_{o_{f_i}} \cdot T^{\alpha_{f_i}} \cdot \exp(-E_{f_i}/R_o T)$, $i = 1, \dots, M$, where $K_{o_{f_i}}$, α_{f_i} and E_{f_i} are constants. $D_{k_b}(T)$ is defined similarly.

$$\underline{Z} = (Z_1, \dots, Z_N)^T$$

vector of concentrations (in moles per unit volume) of the N species,

$$Z_i = \rho Y_i / W_i, \quad i = 1, \dots, N.$$

$$\underline{Z}^{V'}$$

vector of the M products $\prod_{k=1}^N Z_k^{V'_{ik}}$,

$i = 1, \dots, M$. $\underline{Z}^{V''}$ is defined similarly.

Some of the N species, such as N_2 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 \underline{Y} -components when solving the system Eq. (2.7) numerically, thus reducing its dimensionality to $NA + 1$ where $NA \leq N$ is the number of remaining (non-inert) species. By putting $\underline{y} = (Y_1, \dots, Y_{NA}, T)^T$ and defining $\underline{f}(\underline{y})$ accordingly, Eq. (2.7), (3.1) assume the normal form of an autonomous system of ordinary differential equations

$$\frac{d\underline{y}}{dt} = \underline{f}(\underline{y}). \quad (3.2)$$

Unless Δt , the timestep of the fractional step algorithm, is chosen much smaller than needed for accuracy, the product $-\text{Re}\lambda \cdot \Delta t$ will be much larger than one (maybe 10^3 or larger in our test examples) for some eigenvalues λ to the Jacobian matrix $J = J(\underline{y}) = \partial \underline{f} / \partial \underline{y}$. Hence Eq. (3.2) must be treated by methods suitable for stiff ordinary differential equations. Our test runs were made using a version by Hindmarsh¹⁰ of the widely used algorithm by Gear (See Ref. 6, pp. 158-168) with the purpose of improving its performance by taking advantage of particular features of the above problem in the following way:

1. We subdivide the temperature range of interest into K sub-intervals by introducing constants $T_1 < T_2 < \dots < T_{K-1}$ (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 Ref. 22, p. 353, and Ref. 5), i.e., we compute matrices P , L , and H such that

$$J(\underline{y}) = P L H L^{-1} P^T \quad (3.3)$$

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 $i \geq j > 1$ with $l_{ii} = 1$, $i = 1, \dots, NA+1$. Furthermore, by the requirement that the number of atoms of each chemical element be conserved in each reaction

$$\text{rank } J = \text{rank } H \leq NA + 1 - NE \quad (3.4)$$

independently of \underline{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. 22, 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 \underline{y} in Eq. (3.3) will be in one of the temperature subintervals defined above, say the k 'th, and we store the matrix P and the matrices L , H into the k 'th of the reserved storage areas. Subsequently, whenever the temperature component of \underline{y} in Eq. (3.2) is in the k 'th 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 re-evaluating and reducing $J(\underline{y})$ 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 2-4 below).

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

$$(I - \alpha_p h J(\underline{y})) \underline{x} = \underline{b} \quad (3.5)$$

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

$$P L (I - \alpha_p h H) L^{-1} P^T \underline{x} = \underline{b} \quad (3.6)$$

which can be solved by back-substitutions after LU decomposition of only the $(NA + 1 - NE) \times (NA + 1 - NE)$ upper left corner of the Hessenberg matrix $I - \alpha_p h H$, (See the comment after Eq. (3.4) above). The overhead work for this LU decomposition, which has to be redone whenever the product $\alpha_p \cdot h$ changes significantly, is thus roughly $(NA + 1 - NE)^2/2$ operations to be compared to the roughly $(NA + 1)^3/3$ 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 $3(NA + 1)^2/2$ operations as compared to $(NA + 1)^2$ in the direct LU-decomposition case. However, step/order changes requiring LU decomposition are sufficiently frequent to make the net gain substantial (See Tables 3 and 4 below).

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 stepsize, as compared to $p + 2$ steps in the original strategy (See Ref. 10). We further permit step increase by up to a factor of 10^4 at all times (instead of only initially as in Ref. 10) except for the first increase after a stepsize reduction. The risk of these

modifications introducing unwanted error growth caused by too frequent stepsize changes^{1,7,9,11} is expected to be small because: the order generally is 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 stepsize is as a rule non-decreasing in each initial value problem (cf Ref. 1, p. 102 and Ref. 9, 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 3 below), mainly by using fewer steps in the initial transients which in general restrict the starting stepsize to being 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 values⁴ by rounding or truncation errors in the following way: Let \underline{y}^{pr} be the predicted solution vector computed by polynomial extrapolation. Then we use $\underline{y}^o = (y_1^o, \dots, y_{NA+1}^o)$ as starting point for the corrector iterations, where

$$y_i^o = \max \left\{ y_i^p, 0 \right\}, i = 1, \dots, NA+1, \quad (3.7)$$

and subsequently adjust any negative components occurring in \underline{y} to zero after each corrector iteration. We do, however, use the actual predicted value \underline{y}^{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 conserves linear invariants of the solution, such as element masses, within rounding error tolerance independent of both local error and truncation error from solving the corrector equations. However, such linear invariants are still conserved to within the requested local error tolerance which, of course, is quite satisfactory.

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 1 shows the source references used for obtaining the details of these models, together with the number of species and reactions appearing in each model.

Table 1. Source references and dimensionality data for kinetics models used in test runs.

Test case number	Kinetics model, references	Number of species, N	Number of reacting species, NA	Number of reactions, M
1	H ₂ - Air ^{13,15,2}	8	7	13
2	CH ₄ - Air ^{18,21}	19	18	56
3	CH ₄ - Air ^{19,21}	26	25	86

The entries of the last column of Table 1 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 Eq. (3.1).

The thermodynamic functions $\underline{h}(T)$ and $\underline{C}_p(T) = dh/dT$ were given as piecewise polynomials in T as in Ref. 8, p. 32. The constant pressure p_0 was chosen to be $1.013 \cdot 10^7$ dyn/cm² (= 1 atm). The choice of values for the thermal conductivity λ and the diffusion coefficients d_1, \dots, d_n is not crucial for our results, and we simply put $d_1 = \dots = d_n = d$, choosing the constants d (5 cm²/sec) and λ ($5 \cdot 10^3$ dyn/sec) so as to obtain burning velocities, species concentration profiles, and flame thicknesses roughly comparable to those in Ref. (13) and (20) in the steady state flame. The function $\theta(t)$ in the boundary condition Eq. (2.6) for the temperature was chosen to be:

$$\theta(t) = \begin{cases} T_s + (T_e - T_s) \cdot t/t_r & , 0 \leq t \leq t_r \\ T_e & , t \geq t_r \end{cases} \quad (4.1)$$

with $t_r = 10^{-4}$ sec, $T_s = 300^\circ\text{K}$, $T_e = 2400^\circ\text{K}$ (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 conditions to the right were given at $L = 1 \text{ cm}$ and the step-sizes $\Delta x = 5 \cdot 10^{-3} \text{ cm}$, $\Delta t = 2.5 \cdot 10^{-6} \text{ sec}$ were found to give sufficient accuracy for our purposes. At these stepsizes the bounds $\Delta t \cdot \|S_D\|_\infty \lesssim 2$, $\Delta t \cdot \|S_H\|_\infty \lesssim 1.8$, 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 $0 \leq t \leq 0.5 \text{ msec}$ in the H_2 - air case and $0 \leq t \leq 0.6 \text{ msec}$ 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 $\epsilon = 10^{-3}$, 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. (10) by

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

where β_p is a number dependent of the order p , $\underline{e} = (e_1, \dots, e_{NA+1})^T$ is the difference between the corrected and predicted solution values, and $y_{\max,1}, \dots, y_{\max,NA+1}$ are weights to be chosen by the user. We update these weights after each integration step in Eq. (3.1) to be the maximum values of y_1, \dots, y_{NA+1} in any solution to Eq. (2.7), (3.2) computed since $t = 0$.

In Table 2, we list some of the main subalgorithms needed in solving the initial value problem (2.7), (3.2) and show the CPU-times required on a Digital VAX 11/780 computer for executing each of these once.

Table 3 shows the average number of integration steps, function evaluations, and matrix handling operations per initial value problem performed in our runs of the three test cases. The entries in the "modified ODE solver" columns were obtained using the algorithm in Ref. 10 with the modifications described above. The entries under "original ODE-solver" are shown for comparison and were obtained using the original algorithm of Ref. 10 with, however, the prediction-correction scheme modified

as described above. Our attempts to use Ref. 10 without this modification 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 rows 7 and 8 of Table 4 are the average CPU-times in milliseconds per meshpoint ($i\Delta x$, $i\Delta t$) needed for the chemistry step and for all other fractional steps, respectively. The entries in row 9 are the sums of the corresponding entries in rows 7 and 8 and hence show the average total CPU-time required per meshpoint by the fractional step algorithm. Rows 1-6 show a breakdown of the entries in row 7 into the average CPU-times per meshpoint spent in the main subalgorithms of the chemistry step. We remark that we actually applied the ODE-solver on Eq. (2.7) only if $|T_i^j - T_0| \gtrsim 0.5^\circ\text{K}$, i.e. only at meshpoints in and behind the flamefront, otherwise approximating the solution of Eq. (2.7) by the constant initial values \underline{y}_i^j , T_i^j . The averages shown in Table 4 are based on the subset of meshpoints where the ODE-solver was applied (roughly 15-25% of all meshpoints in the strip $0 \leq X \leq L$, $0 \leq t \leq t_{\text{end}}$ in all cases), and the CPU-times of the other fractional steps have been normalized accordingly.

CONCLUDING REMARKS

In summary, Tables 3 and 4 indicate the following results for our test cases:

1. a roughly 85% reduction of the CPU time spent on Jacobian evaluations and matrix decompositions.

2. a 20%-40% reduction of the number of integration steps, counteracted, however, by perhaps 10-25% slower corrector convergence rate and more time-consuming backsolves, resulting in a net change of the CPU time spent on function evaluations and solution 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 stepsize and order selection and other overhead.

The net effect of these changes is a reduction of the overall CPU time for the fractional step algorithm by 38%-53%, increasing with the size of the chemical kinetics model as the overhead for Jacobian evaluation and matrix decomposition grows dominant.

A large portion of the computational work in the chemistry step (See Table 3) is caused by the initial rapid transients occurring in the solution of Eq. (2.7), (3.2) at each meshpoint. The introduction of these transients is largely an artificial by-product of the operator splitting technique and indeed is the main price to be paid for the computational advantages of this technique. It is not unreasonable to expect, however, that much of these transients could in fact be resolved only crudely without adverse effects on the global accuracy of the solution. In particular, one might consider applying a 'numerically superstable' integration formula like the implicit Euler method on Eq. (2.7), (3.2) choosing the stepsize without regard to local error estimates in the initial steps. Some preliminary computational experiments using the above equations and kinetics models support this idea at least in part: A constant stepsize predictor-corrector scheme related to the implicit

Euler formula (and complemented with some ad hoc criteria for accepting/rejecting solutions to Eq. (2.7), (3.2) based on their global behavior) was found to give quite acceptable global accuracy for the H_2 -Air model (Case 1), when using in the steady-state flame only two integration steps, two evaluations of $\underline{f}(\underline{y})$, three backsolves and no evaluations of $J(\underline{y})$ per meshpoint (cf. the corresponding data for the variable step algorithm in Table 3, Case 1). For the CH_4 -Air model, (Cases 2 and 3) however, sufficient global accuracy was obtained only by choosing the constant step-size too small to be competitive with the variable step technique.

Table 2. CPU-times (in milliseconds) per execution of some algorithms needed in the chemistry step.

	<u>Case 1</u>	<u>Case 2</u>	<u>Case 3</u>
Evaluation of $\underline{f}(\underline{y})$ in Eq. (3.2), (2.7), (3.1)	2.7	9.2	13.8
Evaluation of $J(\underline{y}) = \partial \underline{f} / \partial \underline{y}$	12.2	54.3	92.2
LU decomposition of $I - \alpha_p h J(\underline{y})$ 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 $J(\underline{y})$ to Hessenberg form as in Eq. (3.3)	7.7	87.6	216
LU decomposition of $I - \alpha_p h H$ 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 $\underline{h}(T_i^j), \underline{C}_p(T_i^j)$ and \underline{C}_p in Eq. (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 3. Average number of integration steps, function evaluations and matrix handling operations per initial value problem (2.7), (3.2).

	Case 1		Case 2		Case 3	
	Original ODE-solver	Modified ODE-solver	Original ODE-solver	Modified ODE-solver	Original ODE-solver	Modified ODE-solver
Integration steps	8.9	7.0	12.8	9.8	17.5	10.5
Evaluations of $\underline{f}(\underline{y})$	14.0	11.9	16.4	14.9	21.7	16.3
Evaluation of $J(\underline{y}) := \partial \underline{f} / \partial \underline{y}$	3.9	0.08	4.5	0.16	5.9	0.20
Reduction of $J(\underline{y})$ to Hessenberg form	---	0.08	---	0.16	---	0.20
LU decompositions	3.9	5.2	4.5	7.7	5.9	7.7
Solutions of decomposed linear system	13.0	10.9	15.4	13.9	20.7	15.3

Table 4. Average CPU times (in milliseconds) per meshpoint in the fractional step algorithm.

	Case 1		Case 2		Case 3	
	Original ODE-solver	Modified ODE-solver	Original ODE-solver	Modified ODE-solver	Original ODE-solver	Modified ODE-solver
<u>Chemistry step:</u>						
Evaluation of $f(\underline{y})$	38	32	151	137	299	225
Solution of decomposed linear systems	17	20	89	131	220	275
Evaluation of $J(\underline{y})$	48	1	244	9	544	18
Matrix decomposition and reduction	21	8	214	63	662	138
Evaluation of parameters \underline{h} , \underline{C} , \underline{D} and \underline{k}_f , \underline{k}_b	3	3	13	13	20	20
Execution of ODE-solver and other overhead	31	32	128	132	156	196
Chemistry step, total	158	96	839	488	1901	872
Other fractional steps	5	5	12	12	37	37
All fractional steps	163	101	851	500	1938	909

REFERENCES

1. R. K. Brayton, F. G. Gustavson, G. D. Hachtel, "A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas", IEEE Proceedings, 60, (1), 98-108 (1972).
2. N. J. Brown-Williamson, (private communication).
3. L. Edsberg, "Some mathematical properties of mass action kinetics", TRITA-NA-7505, The Royal Institute of Technology, Stockholm (1975).
4. L. Edsberg, "Numerical methods for mass action kinetics", in Numerical Methods for Differential Systems. L. Lapidus, W. E. Schiesser, eds., (Academic Press, New York, 1976), pp. 181-195.
5. W. H. Enright, "Improving the efficiency of matrix operations in the numerical solution of stiff ODE's", Technical Report 98, Department of Computer Science, University of Toronto, (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, "The effect of variable mesh size on the stability of multistep methods", SIAM J. Numer. Anal., 11, 5, 1025-1043 (1974).

8. S. Gordon and B. J. McBride, "Computer program for calculation of complex chemical equilibrium compositions, rocket performance, incident and reflected shocks, and Chapman-Jouguet detonations", NASA SP-273, National Aeronautics and Space Administration, Washington, D.C. (1971).
9. G. K. Gupta and C. S. Wallace, "A new step-size changing technique for multistep methods", Math. Comp., 33, 125-138 (1979).
10. A. C. Hindmarsh, "GEAR: Ordinary differential equation system solver", UCID-30001, Rev. 3, Lawrence Livermore Laboratory, Livermore, California (Dec. 1974).
11. F. T. Krogh, "Algorithms for changing the step size", SIAM J. Numer. Anal., 10, (5), 949-965 (1973).
12. S. B. Margolis, "Time-dependent solution of a premixed laminar flame", J. Comp. Phys. 27, 410-427 (1978).
13. J. A. Miller and R. J. Kee, "Chemical nonequilibrium effects in hydrogen-air laminar jet diffusion flames", J. Phys. Chem. 81, 2534-2542 (1977).
14. A. W. Rizzi and H. E. Bailey, "Split space-marching finite volume method for chemically reacting supersonic flow", AIAA Journal, 14, (5), 621-628 (May 1966).
15. R. Schefer and F. Robben, "Catalyzed combustion in a flat plate boundary layer, II. Numerical calculations", Lawrence Berkeley Laboratory Report LBL-6842 (1977).
16. L. F. Shampine and C. W. Gear, "A user's view of solving stiff ordinary differential equations", SIAM Review, 21, (1), 1-17 (1979).

17. P. D. Thomas and K. H. Wilson, "Efficient computation of stiff chemically reacting flow in turbulent free jets", AIAA Journal, 14, (5), 629-636 (1976).
18. C. K. Westbrook, et.al., "A numerical model of chemical kinetics of combustion in a turbulent flow reactor", J. Phys. Chem. 81, 2542-2554 (1977).
19. C. K. Westbrook and F. L. Dryer, "A comprehensive mechanism for methanol oxidation", Combustion Science and Technology, 20, 267 (1979).
20. C. K. Westbrook and F. L. Dryer, "Prediction of laminar flame properties of methanol-air mixtures", UCRL-82717, Lawrence Livermore Laboratory, Livermore, (April 1979).
21. C. K. Westbrook, (private communication).
22. J. H. Wilkinson, The algebraic eigenvalue problem, Oxford University Press, London (1965).
23. F. A. Williams, Combustion theory, Addison Wesley, Reading, Mass. (1965)
24. N. N. Yanenko, The method of fractional steps, Springer, New York (1971).

LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.