

AD-A122 170

PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON STIFF
COMPUTATION APRIL 12. (U) UTAH UNIV SALT LAKE CITY DEPT
OF CHEMICAL ENGINEERING R C AIKEN 1982

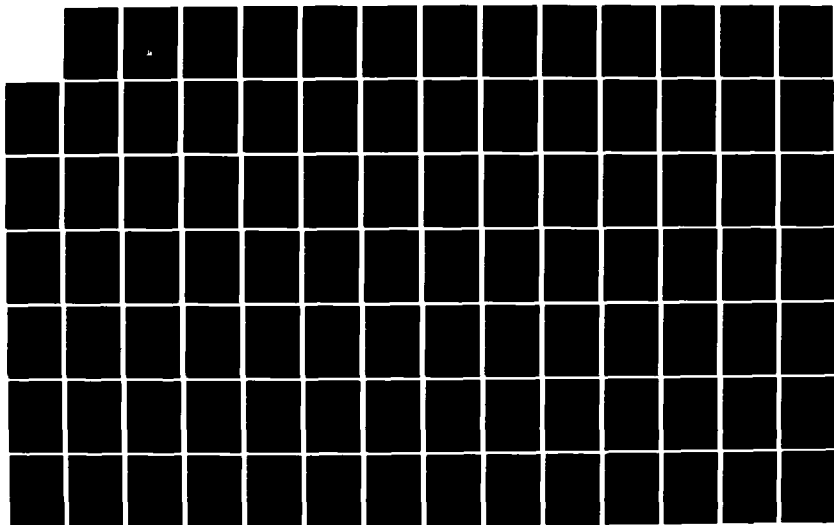
1/5

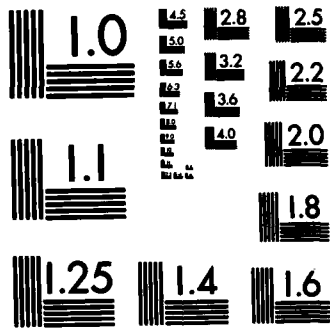
UNCLASSIFIED

AFOSR-TR-82-1036-VOL-2 AFOSR-82-0038

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFOSR-TR- 82 - 1036

2

AD A 122 170

Proceedings

International Conference on Stiff Computation

April 12, 13, 14, 1982
at Park City, Utah



AFOSR-82-0038

Approved for public release;
distribution unlimited.

Vol. II

Sponsored by the U.S. Air Force
Office of Scientific Research

82 12 08 045

ENC FILE COPY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|--|--|---|
| 1. REPORT NUMBER AFOSR-TS- 82-1086 | 2. GOVT ACCESSION NO. AD-A122170 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) PROCEEDINGS, INTERNATIONAL CONFERENCE ON STIFF COMPUTATION, VOLUMES I, II, AND III | | 5. TYPE OF REPORT & PERIOD COVERED TECHNICAL |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Richard C. Aiken | | 8. CONTRACT OR GRANT NUMBER(s) AFOSR-82-0038 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Chemical Engineering University of Utah Salt Lake City UT 84112 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A3 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Directorate of Mathematical & Information Sciences Air Force Office of Scientific Research Bolling AFB DC 20332 | | 12. REPORT DATE 12-14 April 1982 |
| | | 13. NUMBER OF PAGES 385 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) UNCLASSIFIED |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES Proceedings, International Conference on Stiff Computation, April 12-14, 1982 Park City, Utah. | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) These three volumes constitute the written contributions of speakers at the International Conference on Stiff Computation, held April 12-14, 1982, at Park City, Utah. As this collection was prepared in advance of the meeting, a few contributions were too late to be included here. The purpose of this meeting was to bring together theorists, software developers, and users on common ground to consider the state of the art - and practice - of stiff computation. Most of the papers in these proceedings will appear formally in the form of a monograph. | | |

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

**PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON STIFF COMPUTATION**

**April 12-14, 1982
Park City, Utah**

Volume II

**AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DTIC
This technical report has been reviewed and is
approved for public release in accordance with AFM AFR 193-12.
Distribution is unlimited.
MATTHEW J. HANCOCK
Chief, Technical Information Division**

VOLUME II

APRIL 12, 1982

primary contents include:

L.F. SHAMPINE, Sandia National Laboratories:

WHAT IS STIFFNESS?

F.T. KROGH, Jet Propulsion Laboratory:

NOTES ON PARTITIONING IN THE SOLUTION OF STIFF EQUATIONS;

G.D. BYRNE, Exxon Research and Engineering Company:

ANOTHER VIEW OF STIFF DIFFERENTIAL SYSTEMS;

W.D. SEIDER, (speaker), C.W. WHITE, III, G.J. PROKOPAKIS, University of Pennsylvania:

(4) STIFF ORDINARY DIFFERENTIAL EQUATIONS IN CHEMICAL PROCESS ANALYSIS;

P.M. DEW (speaker), T.S. CHUA, University of Leeds:

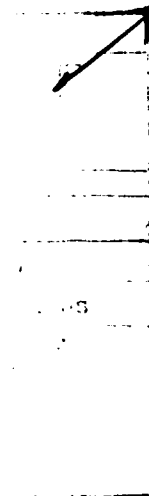
(5) NUMERICAL INTEGRATION OF STIFF DIFFERENTIAL/ALGEBRAIC EQUATIONS WITH SEVERE DISCONTINUITIES;

F.C. HOPPENSTEADT (speaker), P. ALFELD, University of Utah:

EXPLOSION MODE ANALYSIS OF AN H_2-O_2 REACTION

W.L. MIRANKER, IBM:

(6) AN OVERVIEW OF THE HIGHLY OSCILLATORY INITIAL VALUE PROBLEM;



A

R.M.M. MATTHEIJ, Katholieke Universiteit,
Netherlands:

(7) RICCATI TYPE TRANSFORMATIONS AND
DECOUPLING OF SINGULARLY PERTURBED ODE.

R.E. O'MALLEY, JR., Rensselaer Polytechnic
Institute (speaker),
R.M.M. MATTHEIJ, Katholieke Universiteit:

DECOUPLING OF BOUNDARY VALUE PROBLEMS
FOR TWO-TIME SYSTEMS

T.A. BICKART, Syracuse University:

P-STABLE AND $P(\alpha, \beta)$ -STABLE
INTEGRATION/INTERPOLATION METHODS
IN THE SOLUTION OF RETARDED
DIFFERENTIAL-DIFFERENCE EQUATIONS

D.J. Nolting, USAF Academy (speaker),
D.J. Rodabaugh, Lockheed:

(8) STIFFLY STABLE LINEAR MULTISTEP METHODS.

S.O. FATUNLA, Trinity College, Dublin:

P-STABLE HYBRID SCHEMES FOR INITIAL
VALUE PROBLEMS

APRIL 13, 1982

G. DAHLQUIST, The Royal Institute of
Technology:

SOME COMMENTS ON STABILITY AND ERROR
(4) ANALYSIS FOR STIFF NONLINEAR PROBLEMS,

W. LINIGER, IBM:

(10) CONTRACTIVITY OF MULTISTEP AND ONE-LEG
METHODS WITH VARIABLE STEPS.

J.R. CASH, Imperial College of Science and
Technology:

A SURVEY OF RUNGE-KUTTA METHODS FOR THE
NUMERICAL INTEGRATION OF STIFF
DIFFERENTIAL SYSTEMS

W.E. SCHIESSER, Lehigh University:

SOME CHARACTERISTICS OF ODE PROBLEMS
GENERATED BY THE NUMERICAL METHOD OF
LINES

B.A. FINLAYSON, University of Pennsylvania:

SOLUTION OF STIFF EQUATIONS RESULTING
FROM PARTIAL DIFFERENTIAL EQUATIONS

S.W. CHURCHILL, University of Pennsylvania:

STIFFNESS IN HEAT TRANSFER

J.O.L. WENDT, University of Arizona (speaker)

W.A. HAHN, Exxon Production Research Center:

INTEGRATION OF THE STIFF, BOUNDARY
VALUED ODE'S FOR THE LAMINAR, OPPOSED
JET DIFFUSION FLAME

J.E. DOVE, University of Toronto (speaker),

S. RAYNOR, Harvard University:

A MASTER EQUATION STUDY OF THE RATE AND
MECHANISM OF VIBRATIONAL RELAXATION AND
DISSOCIATION OF MOLECULAR HYDROGEN BY
HELIUM (abstract)

C.A. COSTA (speaker), M.Q. DIAS, J.C. LOPES,

A.E. RODRIGUES:

DYNAMICS OF FIXED BED ADSORBERS
(abstract)

F.E. CELLIER, ETZ-Zurich:

STIFF COMPUTATION: WHERE TO GO?

What is Stiffness?

L. F. Shampine
Sandia National Laboratories
Albuquerque, New Mexico 87185

1. Introduction

The numerical solution of the initial value problem for a system of N first order ordinary differential equations is considered:

$$y' = f(x,y) \quad a < x < b \quad (1.1)$$

$$y(a) \text{ given} \quad (1.2)$$

Modern codes based on step-by-step methods begin with $y(a) = y_0$ at $x_0 = a$ and step from a to b successively producing approximations y_n to $y(x_n)$ on a mesh $x_0 < x_1 < \dots < x_M = b$. At the step from x_n the code selects a step size h_{n+1} so that the resulting approximation at $x_{n+1} = x_n + h_{n+1}$ is to satisfy a certain accuracy requirement.

The standard assumption about $f(x,y)$ is that it has as many continuous derivatives as needed and that it satisfies a Lipschitz condition,

$$|f(x,u) - f(x,v)| < L |u-v| \quad (1.3)$$

for $a < x < b$ and all y . This guarantees (1.1, 1.2) has a unique solution $y(x)$. The "classical situation" is that $L(b-a)$ is not "large." In this situation classical numerical methods such as Runge-Kutta and Adams are quite satisfactory with one major exception -- too frequent output can drastically reduce the step size. There are a variety of other reasons why the step size might have to be restricted, but in the classical situation, none can lead to a severe restriction of the step size.

When $L(b-a)$ is "large," the matter is quite different. In the first place, the class of mathematical problems must be restricted in order for step-by-step methods to have a chance of solving the problem adequately. It develops that in some extremely important circumstances, classical methods suffer a variety of restrictions on the step size which are so severe that such methods are impractical. This is what is usually termed "stiffness."

Many authors have sought a definition of stiffness involving only the mathematical problem (1.1, 1.2). Unfortunately the situation is far more complex than that. In this paper we shall explore those factors determining the step size in a sequence like that of the usual treatment of Runge-Kutta methods. All the seeds of step size restrictions are present in the classical situation. We shall see how a large Lipschitz constant affects the algorithms and shall describe ways to circumvent or overcome the difficulties. We aim to provide some feeling as to the kinds of problems leading to severe step size restrictions for classical methods and to show how the computational problem, formula, and implementation are related to "stiffness." Some remarks will be made about how such restrictions can be recognized automatically.

2. Mathematical Problem

We are interested in approximating a vector function $y(x)$ which satisfies (1.1) as an identity for given function $f(x,y)$ and interval $[a,b]$. The solution $y(x)$ is to have the specified initial value $y(a)$ at $x = a$. The assumption that $f(x,y)$ is continuous and satisfies a Lipschitz condition (1.3) for $a < x < b$ and all y guarantees that the equation (1.1) has a unique solution for any given initial value y_n at any point x_n in $[a,b]$. It is easy to reduce the requirements on f to holding only in a neighborhood of the solution $y(x)$. Although this reduction is of great theoretical and practical

value, we do not want to complicate matters by going into details.

One-step methods exemplify step-by-step methods. They have the form

$$y_{n+1} = y_n + h_{n+1} \Phi(x_n, y_n; h_{n+1}) . \quad (2.1)$$

Having reached x_n , such methods are supplied only the values x_n, y_n, h_{n+1} , and the ability to evaluate f . It is clear that the best one can hope to do is to approximate $u(x_{n+1})$ where $u(x)$ is the "local solution" of (1.1) with initial value y_n :

$$u' = f(x, u) , \quad u(x_n) = y_n . \quad (2.2)$$

Some of the classical methods use a small number of the most recent previously computed approximations, but the situation is not really any different for them.

The question which immediately arises is how well does $u(x)$ approximate $y(x)$ for $x > x_n$. This is a question of the stability of the solution $y(x)$. A classical result states that if $u(x), v(x)$ are two solutions of (1.1), then

$$|u(x+\Delta) - v(x+\Delta)| < e^{L\Delta} |u(x) - v(x)| \quad (2.3)$$

where L is the Lipschitz constant of (1.3). The result is sharp as the single equation $y' = Ly$ shows.

We simply cannot solve unstable problems in practice by the kinds of methods we consider. A single error of ϵ at x_n moves us from $y(x)$ to the $u(x)$ of (2.2). If $u(x)$ diverges strongly from $y(x)$, the error of ϵ is amplified accordingly. Of course what "too" much means depends on the accuracy desired and the computing budget. In the classical situation, (2.3) implies that $y(x)$ is reasonably stable.

A large Lipschitz constant tells us that some solutions of (1.1) either diverge or converge rapidly in a relative sense. Suppose equality holds in (1.3) for x_0, u_0, v_0 near $y(x_0)$. Let $u(x)$ be the local solution of (1.1) with $u(x_0) = u_0$ and similarly define $v(x)$. Further let $\delta(x) = u(x) - v(x)$. Then (1.3) states that

$$\frac{|\delta'(x_0)|}{|\delta(x_0)|} = L \gg 1 .$$

There are problems of great practical importance for which the Lipschitz constant is "large." Here and later we say the Lipschitz constant is large as an abbreviation for the statement that $(b-a)L$ is large; the reader should remember that the length of the interval is also important. As noted, a problem with a large Lipschitz constant may be unstable. For the kinds of numerical methods we consider, we must assume that the solution is stable. In order to formulate quantitative results, we must supplement the Lipschitz condition with some other condition guaranteeing stability.

The most studied class of problems with large Lipschitz constants consists of linear problems with constant Jacobians:

$$y' = Jy + F(x) . \quad (2.4)$$

It is often assumed for convenience, and we do so, that J has a complete set of eigenvectors $\{v_i\}$ and associated eigenvalues $\{\lambda_i\}$. The general solution can then be written in terms of a particular solution $p(x)$ as

$$y(x) = p(x) + \sum_{i=1}^N \alpha_i v_i e^{\lambda_i(x-a)} . \quad (2.5)$$

This class is so simple that many numerical procedures can be analyzed in considerable detail and so provides insight as to more general problems.

The representation (2.5) makes it clear that stability is equivalent to the requirement that for all i , either $\text{Re}(\lambda_i) < 0$ or if $\text{Re}(\lambda_i) > 0$, then $(b-a)\text{Re}(\lambda_i)$ is not large. Notice that $\|J\| = L > |\lambda_i|$ for all i , so the presence of some λ_j with $(b-a)\text{Re}(\lambda_j) \ll -1$ implies a large Lipschitz constant.

It is often suggested that the general problem (1.1) be modelled near a point $x_n, y(x_n)$ by a problem of the form (2.4):

$$z' = f(x_n, y(x_n)) + f_x(x_n, y(x_n)) + f_y(x_n, y(x_n))(z - y(x_n))$$

where f_y is the Jacobian matrix of first partial derivatives of f . This is a time-honored tactic of applied mathematics. We say that our numerical methods should perform well on problems of the form (2.4) and that we hope the linearization stated will provide a useful guide for more general problems.

Recent work has considered f which satisfy

$$\langle f(x, u) - f(x, v), u - v \rangle < l \|u - v\|^2 \tag{2.6}$$

for a suitable inner product $\langle \cdot, \cdot \rangle$. This is a one-sided Lipschitz condition because in general

$$-L \|u - v\|^2 < \langle f(x, u) - f(x, v), u - v \rangle < L \|u - v\|^2 .$$

The new thing is that l might be a lot less than L and even negative. By differentiating $D(x) = \exp(-2lx) \|u(x) - v(x)\|^2$, one can show that

$$\|u(x+\Delta) - v(x+\Delta)\| < e^{l\Delta} \|u(x) - v(x)\| , \tag{2.7}$$

which can be a great improvement over (2.3).

A variant is to consider problems of the form

$$y' = Jy + g(x, y) \tag{2.8}$$

where g satisfies a Lipschitz condition

$$\|g(x,u) - g(x,v)\| < \rho \|u-v\| \quad (2.9)$$

and μ is the smallest constant such that

$$\langle Jz, z \rangle < \mu \|z\|^2 \quad \text{all } z$$

for the constant matrix J . The quantity $\mu = \mu[J]$ is called the logarithmic "norm" of J and it may be negative. One can then prove

$$\|u(x+\Delta) - v(x+\Delta)\| < e^{(\mu+\rho)\Delta} \|u(x) - v(x)\| .$$

Notice that this class is included in (2.6) with $l = \rho + \mu$. This class is a natural one for the investigation of several kinds of methods and, as we shall see later, is natural for certain practical reasons as well.

In summary, continuity of f and a Lipschitz condition guarantee existence and uniqueness of the solution $y(x)$ of (1.1, 1.2). We cannot solve (1.1, 1.2) in practice using step-by-step methods if $y(x)$ is not moderately stable. In the classical situation of $(b-a)L$ not large, the Lipschitz condition alone guarantees adequate stability. Otherwise we must assume stability, or supplement the Lipschitz condition with other hypotheses about the problem which guarantee stability. We have mentioned three such subsets of Lipschitzian problems for which stability can be demonstrated even when the Lipschitz constant is large.

A qualitative way to describe the problems with large L which interest us is that $y(x)$ is stable in the direction of integration and some solution of (1.1) approaches $y(x)$ very rapidly. A way we prefer for describing the latter condition is that $y(x)$ is very unstable in the opposite direction. Notice that the stability result (2.3) shows that in the classical situation, $y(x)$

is stable in both directions. For stiffness to occur, there must be a strong directional effect.

3. Classical Formulas

The Adams family of formulas for approximating the solution of (1.1) arises from an equivalent integral form:

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} y'(t) dt \quad . \quad (3.1)$$

Approximations to $y'(t)$ arise naturally from approximations y_1 to $y(x_1)$ by

$$y'(x_1) = f(x_1, y(x_1)) = f(x_1, y_1) \quad . \quad (3.2)$$

The Adams formulas approximate $y'(t)$ in (3.1) by interpolating values $f_i = f(x_i, y_i)$ previously computed. Thus an Adams-Bashforth formula of order p forms the (unique) polynomial $P(x)$ of degree p interpolating to f_{n-1} for $i = 0, 1, \dots, p$ and then defines

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} P(t) dt \quad .$$

This explicit formula is defined for any set of mesh points but simplifies in the case of constant step size h to a formula of the form

$$y_{n+1} = y_n + h \sum_{i=0}^p \alpha_i f_{n-1} \quad .$$

The simplest example is the (forward) Euler method

$$y_{n+1} = y_n + hf_n \quad . \quad (3.3)$$

The Adams-Moulton formula on p points interpolates to the unknown value $f(x_{n+1}, y_{n+1})$, too:

$$P(x_{n+1-i}) = f_{n+1-i} \quad , \quad i = 0, 1, \dots, p \quad .$$

This formula defines y_{n+1} implicitly. For constant step size it has the form

$$y_{n+1} = y_n + h\beta_0 f(x_{n+1}, y_{n+1}) + h \sum_{i=1}^p \beta_i f_{n+1-i} \quad .$$

The simplest example is the backward Euler method

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \quad .$$

The backward differentiation formulas (BDF) have a similar origin. Now the polynomial $P(x)$ interpolates to solution values:

$$P(x_{n+1-i}) = y_{n+1-i} \quad i = 0, 1, \dots, p \quad ,$$

and the differential, rather than the integral, form of the equation is used:

$$P'(x_{n+1}) = f(x_{n+1}, P(x_{n+1})) = f(x_{n+1}, y_{n+1}) \quad .$$

For constant step size h this results in a formula of the form

$$y_{n+1} = h\gamma_0 f(x_{n+1}, y_{n+1}) + \sum_{i=1}^p \gamma_i y_{n+1-i} \quad .$$

The backward Euler formula happens to be one of the BDF.

These formulas are representative of classical methods using previously computed solution values. In the case of implicit methods such as Adams-Moulton and BDF there are the immediate questions as to whether they are well-defined, and how they are to be evaluated in practice. In the classical use this is

always done by simple iteration. A predicted value $y_{n+1}^{(0)}$ is first formed with an explicit formula. The iteration method will be exemplified for the backward Euler method:

$$y_{n+1}^{(m+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(m)}) \quad m = 0, 1, \dots \quad (3.4)$$

It is easy to see that a sufficient condition for convergence is that

$$hL < 1.$$

It is necessary that

$$h\rho(f_y(x_{n+1}, y_{n+1})) < 1$$

where f_y is the Jacobian matrix of f and $\rho(M)$ is the spectral radius of the matrix M . The sufficient condition guarantees contraction in the norm being used. It is virtually the necessary condition in practice because only a very few iterations are allowed in the codes and convergence must be observed.

There is a variation on the use of the Adams-Moulton methods which does a fixed number of iterations, a common choice being one. An example is the use of the Euler method as predictor and one "correction" with the backward Euler method:

$$y_{n+1}^{(0)} = y_n + hf(x_n, y_n) \quad ,$$

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}^{(0)}) \quad .$$

This variation results in explicit methods which behave similarly, but by no means identically, to the implicit methods.

The one-step methods were mentioned earlier. The most important classical examples are the Runge-Kutta methods. An s -stage method has the form

$$y_{n+1} = y_n + \sum_{i=1}^s c_i k_i$$

where

$$k_i = f\left(x_n + ha_i, y_n + h \sum_{j=1}^s b_{ij} k_j\right) \quad i = 1, \dots, s .$$

The constants a_i satisfy

$$a_i = \sum_{j=1}^s b_{ij} \quad i = 1, \dots, s ,$$

and the constants b_{ij} , c_i define the method. If $b_{ij} = 0$ for all $j > i$, the k_i are evaluated explicitly in the order $i = 1, \dots, s$ and the method is explicit. Otherwise it is implicit. The Euler method (3.3) is also an example of an explicit Runge-Kutta formula and the backward Euler method is an example of an implicit Runge-Kutta formula. In the classical use of implicit Runge-Kutta formulas, they are also evaluated by simple iteration:

$$k_i^{(n+1)} = f\left(x_n + ha_i, y_n + h \sum_{j=1}^s b_{ij} k_j^{(n)}\right) \quad n = 0, 1, \dots .$$

A sufficient condition for convergence is $h\gamma L < 1$, where γ is a constant depending only on the formula.

The classical formulas, as exemplified above, are all explicit, or are evaluated by simple iteration. In every case, simple iteration converges if the step size satisfies $hL < \tau$ for a constant τ which depends only on the method and which is not particularly small. In the classical situation $hL < (b-a)L$ and L is not large, hence simple iteration does not pose a severe

restriction on the step size. Obviously the situation is quite different if the Lipschitz constant is large. It is important to appreciate that a severe restriction on the step size arises only when $hL \gg 1$. Thus the problem must have a large Lipschitz constant, the solution must be stable, and the solution must be easy to approximate in the sense that the desired accuracy can be achieved with a step size h such that $hL \gg 1$. Later we shall explore circumstances leading to this situation. The restriction is one manifestation of the complex of phenomena called stiffness. The difficulty of defining stiffness is evident here. For some kinds of methods there is no step size restriction of this kind at all. The ones that do have such restrictions suffer in varying degrees. Clearly stiffness depends on the method as well as the problem.

The key to solving problems with large Lipschitz constants is to resort to a more powerful iteration method for evaluating implicit formulas. The only popular way is to linearize the algebraic equations by the simplified Newton's method. In the case of the backward Euler method this is

$$y_{n+1}^{(m+1)} = y_n + h \left[f(x_{n+1}, y_{n+1}^{(m)}) + J(y_{n+1}^{(m+1)} - y_n^{(m)}) \right] \quad (3.5)$$

Here $J = f_y(x_{n+1}, y_{n+1})$. Newton's method does not use a fixed matrix J , rather uses $f_y(x_{n+1}, y_{n+1}^{(m)})$ to calculate $y_{n+1}^{(m+1)}$. Generally the formation of approximate Jacobians is very expensive. For example, each iteration of (3.4) requires only one evaluation of f . In the common case of forming J by differences for a system of N equations, N evaluations of f are made just to form J . For this reason it is considered impractical to use Newton's method itself. In (3.5) one must repeatedly solve linear systems with matrix $I-hJ$. Simply decomposing this matrix into triangular factors is rather expensive, and the repeated substitution processes to solve for the $y_{n+1}^{(m+1)}$ are a cost by no means negligible.

The simplified Newton iteration is so much more expensive than simple iteration that it can be worthwhile only if the step size must be severely restricted to get convergence with simple iteration. Of course a great deal of research has been, and is being, devoted to minimizing the costs of this iteration. For example, the s -stage Runge-Kutta process involves a system of sN algebraic equations at each step. This can be reduced in various ways to dealing with systems of size N . The structure of the Jacobians can be used to reduce the costs of forming these matrices and of solving the resulting linear systems.

An important cost-saving device is to use an approximate Jacobian J for as many steps as possible. Convergence of the iteration and the rate of convergence depend on how well J approximates $f_y(x_{n+1}, y_{n+1})$ for the various x_{n+1} . As long as convergence is adequate, one should continue to use J . It is a fundamental practical assumption that the Jacobian is roughly constant for distances much larger than $1/L$. This is one reason why the class of problems of (2.8) is particularly significant for analyzing practical computation.

Codes intended for stiff problems are based on implicit methods for reasons we shall examine below. Until recently the simplified Newton iteration was used exclusively during an integration. Whenever hL is not large, this is grossly inefficient. The inefficiency is not as bad as it might seem at first because efficient reuse of an approximate Jacobian reduces the waste. Still, the inefficiency is significant when solving a problem which, e.g., does not have a large Lipschitz constant. Codes intended for non-stiff problems use simple iteration exclusively if they are based on implicit methods. This is grossly inefficient if the problem has a large Lipschitz constant and its solution allows a step size h such that $hL \gg 1$.

It is worth remarking that the same formula, e.g., the backward Euler, might be used in both kinds of code. Recent research has considered how to recognize when simple iteration is feasible and efficient. There has been considerable success at recognizing and coping automatically with this manifestation of stiffness.

4. Output

So far we have considered approximate solutions only at mesh points. If the user wants results at specific points, the code will have to reduce its step size as necessary so that these points are in the mesh. A more efficient alternative is natural with the Adams formulas and the BDF. They are based on polynomial interpolants which can simply be evaluated at the places specified by the user. In this way the step size of the code is not affected by the output requirements. (Actually it is affected weakly in the popular codes for various software reasons.)

One might not think that output could severely restrict the step size, but for some methods it certainly can. Runge-Kutta methods of many stages must take "large" steps to compensate for the expense of a step. The classical extrapolated mid-point rule can be viewed in typical implementations as a family of Runge-Kutta methods of a great many stages. Obviously the methods described for problems with large Lipschitz constants do a great deal of work at each step. This pays off only when the step size is large. The fact that the BDF are not impacted by output is one reason for their popularity. Recent research has considered ways to "interpolate," or at least to weaken the effect of output on methods which produce values only at mesh points.

The effects of output depend on the formula and how it is implemented. Excessive output for certain methods has exactly the same effect as what is called stiffness. However, it is easier to see the origin of the difficulty and there is usually not a severe restriction on the step size. Furthermore, alternative methods not seriously affected by output have been widely available. We shall refer no more to this issue, but the reader should keep in mind that the computational problem (which includes specification of output points) may determine whether the step size possible is severely restricted and further that this depends on the formula and how it is implemented.

5. Stability

In our description of convergence we shall restrict our attention to one-step methods, more particularly to Runge-Kutta methods, in order to avoid technical details arising from use of previously computed solution values. Suppose the integration has reached (x_n, y_n) . A one-step method produces

$$y_{n+1} = y_n + h_{n+1} \Phi(x_n, y_n; h_{n+1}) .$$

Let us define

$$z_{n+1} = y(x_n) + h_{n+1} \Phi(x_n, y(x_n); h_{n+1}) ,$$

the value the formula would produce if it had available the correct solution value at x_n . A fundamental decomposition of the error is

$$y(x_{n+1}) - y_{n+1} = (y(x_{n+1}) - z_{n+1}) + (z_{n+1} - y_{n+1}). \quad (5.1)$$

The difference $y(x_{n+1}) - z_{n+1}$ in (5.1) is the local truncation error, lte, which measures how well the formula approximates the behavior of the differential equation. Notice that

$$y(x_{n+1}) = y(x_n) + h_{n+1} \Phi(x_n, y(x_n); h_{n+1}) + lte .$$

The difference $z_{n+1} - y_{n+1}$ measures the stability of the formula. We need to relate it to $y(x_n) - y_n$ in a way analogous to the stability results for the differential equation itself.

The step size h_{n+1} is chosen so that

$$|lte| < h_{n+1} \epsilon \tag{5.2}$$

for a tolerance ϵ provided by the user of the code. We shall see how this can be accomplished in the next section. Let h_{\min} and h_{\max} be the smallest and largest step sizes allowed by (5.2) for any x in $[a, b]$.

It is easy to establish the stability result for Runge-Kutta methods that

$$|z_{n+1} - y_{n+1}| < (1+h_{n+1}\mathcal{L}) |y(x_n) - y_n| \tag{5.3}$$

which comes directly from the Lipschitz continuity of the function Φ :

$$|\Phi(x, u; h) - \Phi(x, v; h)| < \mathcal{L}|u-v| . \tag{5.4}$$

These results hold only for $h\mathcal{L} < c$. When c is small, $\mathcal{L} \dot{=} L$ and the factor $(1+h_{n+1}\mathcal{L})$ in (5.3) approximates the analogous factor $\exp(h_{n+1}L)$ of the differential equation.

Combining (5.2) and (5.3) leads to

$$\begin{aligned} |y(x_{n+1}) - y_{n+1}| &< (1+h_{n+1}\mathcal{L}) |y(x_n) - y_n| + h_{n+1} \epsilon \\ &< (1+h_{\max}\mathcal{L}) |y(x_n) - y_n| + h_{\max} \epsilon \end{aligned}$$

which then implies

$$\max_k |y(x_k) - y_k| < \frac{\epsilon}{\mathcal{L}} \left[\exp \left(\mathcal{L}(b-a) \frac{h_{\max}}{h_{\min}} \right) - 1 \right] . \tag{5.5}$$

In the classical situation the stability of all the Runge-Kutta methods is adequate and the convergence result (5.5) satisfactory. Let us consider the stability of the formulas when L is large. The constraint that hL be not large is not artificial as the simplest examples show. (It is also not due to implicit formulas as we shall see in a moment.)

In section 2 we discussed the stability of solutions of

$$y' = Jy + F(x) \quad (2.4)$$

when J is a (constant) matrix with a linearly independent set of eigenvectors. The difference of two solutions, $w = u-v$, satisfies $w' = Jw$. The (constant) change of variables $w = M\delta$, where the columns of M are the eigenvectors of J , uncouples the equations into the set

$$\delta_i'(x) = \lambda_i \delta_i(x) \quad i = 1, \dots, N$$

where λ_i is a (possibly complex) eigenvalue of J . It is easy to verify that the same scheme works for the Runge-Kutta methods with the result that

$$\delta_{i,n+1} = R(h\lambda_i)\delta_{i,n}$$

where $\delta_{i,n} = \delta_i(x_n)$. Here $R(z)$ is a polynomial for explicit Runge-Kutta methods and is a rational function for implicit methods.

For the sake of simplicity we consider the case of $\text{Re}(\lambda_i) < 0$ each i so that $\delta_i(x)$ is non-increasing. It is noteworthy that this means that in a suitable norm,

$$\|u(x+\Delta) - v(x+\Delta)\| < \|u(x) - v(x)\| \quad (5.6)$$

The formula preserves this stability property of the differential equation only if $|R(h\lambda_i)| < 1$ for each λ_i . The region of absolute stability S of

the method is the set of z such that $\text{Re}(z) < 0$ and $|R(z)| < 1$. The step size h must be constrained so that $h\lambda_i \in S$ for each eigenvalue of J .

All the explicit Runge-Kutta methods have finite regions of absolute stability. This follows because $|P(z)| \rightarrow \infty$ as $|z| \rightarrow \infty$ for these non-trivial polynomials. Indeed, all the classical explicit methods have finite regions of absolute stability. We point out that $L = \|J\| > |\lambda_i|$ so that in the classical situation this stability constraint on h is not severe, as follows already from the general result for Lipschitzian problems (5.3). Notice that in the norm of (5.6), if the formula is absolutely stable with the step size h , the convergence results are much stronger than in (5.5):

$$\begin{aligned} \|y(x_{n+1}) - y_{n+1}\| &< \|y(x_n) - y_n\| + h_{n+1} \epsilon \\ &< \dots < (h_1 + h_2 + \dots + h_{n+1}) \epsilon \\ &= (x_{n+1} - a) \epsilon, \end{aligned}$$

using $y(a) = y_0$. This reflects the additional information available about the stability of the differential equation.

Obviously, even to solve the very special class of problems (2.4) with $hL \gg 1$, we need methods with infinite stability regions. As already noted, we must resort to implicit methods. Very stable methods exist, e.g., the backward Euler method has

$$|R(z)| = \left| \frac{1}{1-z} \right| < 1 \quad \text{all} \quad \text{Re}(z) < 0,$$

but we then must pay the price of implicitness discussed in the preceding section. A great deal of effort has been devoted to finding formulas with good stability properties which can also be evaluated relatively cheaply. It should be appreciated that the formulas in common use do not have ideal

stability even for the class (2.4). For example, all the popular BDF of orders at most 6 have infinite absolute stability regions. Still, some are finite for eigenvalues near the imaginary axis. It is quite easy, for example, to write down a problem for which the BDF of order 3 has no stability restriction, but the one of order 4 does, and the restriction suffered is just as severe as that of an explicit Runge-Kutta formula. Evidently stiffness depends on the formula as well as the problem.

In our view these stability constraints arising from the simple class (2.4) are necessary requirements for a reasonable numerical procedure. Of course we hope that the behavior for (2.4) is indicative of the behavior of the procedure for more general problems. A considerable amount of practical experience suggests that it is provided that one is a little cautious. For example, the trapezoidal rule has $|R(z)| < 1$ for all z with $\text{Re}(z) < 0$. However, $R(z) \rightarrow -1$ as $|z| \rightarrow \infty$ so that the formula is barely stable for $|h\lambda_1| \gg 1$. One expects, and sees, slowly damped oscillations in the numerical solutions. Furthermore, it is not difficult to write down problems only a little different from those of (2.4) for which this rule is unstable.

Naturally stability results for classes broader than (2.4), e.g., (2.6) or (2.8), are of great interest. We mention one such result. Suppose we consider the class (2.6) with $l = 0$ so that no two solution curves spread apart. A Runge-Kutta method is called BN-stable if the approximate solutions also do not spread apart then. The backward Euler method is an example of a BN-stable formula.

We have said that the step size should be restrained so as to lie in the region of absolute stability, but codes do not impose this restraint directly. It is interesting that the classical methods, as implemented in good codes, do bring this about. Some details are provided in section 7.

The commonly used procedures with infinite stability regions seem not to have stability difficulties often, but typical codes do not handle them well when they do occur. Providing that a method with adequate stability is used, the principal restraint on the step size is the iteration restraint discussed in the preceding section. We do not wish to minimize the importance of a better understanding of stability nor of the need for more stable formulas which are more easily evaluated, but very stable formulas which are apparently adequate for most practical problems are already known.

By resorting to implicit methods one can avoid stability restrictions in a practically useful way. There are costs involved. One is the implicitness already discussed. Another is accuracy. Restricting the choice of parameters defining a Runge-Kutta method so as to assure good stability properties leaves less freedom to develop very accurate formulas. As a rule, the classical methods with poor stability are much more accurate than those at present being used because of their good stability. As a consequence, the classical formulas permit significantly larger step sizes (hence are more efficient) when accuracy dominates the selection of the step size.

6. Accuracy

In section 5 the local truncation error, lte, of a one-step method was defined by the relation

$$y(x_{n+1}) = y(x_n) + h \Phi(x_n, y(x_n); h) + lte .$$

By Taylor series expansion one finds that

$$lte = h^{p+1} \tau(x_n, y(x_n)) + O(h^{p+2}) \quad (6.1)$$

for sufficiently smooth f . In such a case the formula is said to be of order p .

For methods of order $p > 1$ it is always possible to choose h so that

$$|lte| < h \epsilon \quad . \quad (6.2)$$

Indeed for small h , or equivalently small tolerances ϵ , (6.1) shows that the largest h satisfying (6.2) is approximately

$$h_{n+1} \doteq (\epsilon / |\tau(x_n, y(x_n))|)^{1/p} \quad .$$

The smallest step size needed in the integration is approximately

$$h_{\min} \doteq \left(\frac{\epsilon}{\max_{[a,b]} |\tau(x, y(x))|} \right)^{1/p}$$

and the largest,

$$h_{\max} \doteq \min \left((b-a), \left(\frac{\epsilon}{\min_{[a,b]} |\tau(x, y(x))|} \right)^{1/p} \right) \quad .$$

By way of example both the forward and backward Euler methods are seen to have

$$lte = \frac{h^2}{2} y''(x_n) + O(h^3) \quad . \quad (6.3)$$

The general case of two stage, second order explicit Runge-Kutta formulas form a one-parameter family of formulas. In terms of the parameter $\alpha \neq 0$, they are

$$\Phi(x, y; h) = (1-\alpha)f(x, y) + \alpha f \left(x + \frac{h}{2\alpha}, y + \frac{h}{2\alpha} f(x, y) \right) \quad .$$

The local truncation error is

$$lte = h^3 \left[\left(\frac{1}{8\alpha} - \frac{1}{6} \right) y'''(x) - \frac{1}{8\alpha} f_y(x, y(x)) y''(x) \right] + O(h^4) \quad . \quad (6.4)$$

With the exception of output, none of the restrictions considered in this paper can be severe in the classical situation when $L(h-a)$ is not large. It is true that, for example, stability might restrict the step size some so that one needs to consider stability regions when designing a code for non-stiff problems, but there cannot be the kind of restriction that we call stiffness.

Some valuable information can be gleaned easily from the expression for the local truncation error of a method. We are now interested in problems with solutions $y(x)$ which are smooth, but $f_y(x, y(x))$ is "large." If the term τ in (6.1) involves f_y , it is often easy to see that the step size must be restricted so that hL is not large in order to yield the desired accuracy. This excludes some methods immediately from their use for problems with large L . For example, the second term in (6.4) will impose an unwelcome restriction on the step size when f_y is large.

The Adams-Moulton methods are implicit methods which use past solution values. When the step size is a constant h , the local truncation error for the method of order k has

$$\tau = \gamma_k^* y^{(k+1)}(x) ,$$

where γ_k^* is a constant characteristic of the method. A popular variant of these formulas which is explicit was mentioned earlier. This variant has

$$\tau = \alpha_{k,0}^* f_y(x, y(x)) \gamma_k y^{(k)}(x) + \gamma_k^* y^{(k+1)}(x) .$$

The coefficients $\alpha_{k,0}^*$, γ_k are constants. It is evident that this variant is unsuitable for problems with large $f_y(x, y(x))$ despite whatever merits it might have in the classical situation.

These observations do not exclude some formulas which are, in fact,

useless. For example, in the asymptotic expression (6.3) we do not see a difference between the explicit and implicit Euler methods. Still this is an easy way to see that some formulas cannot be helpful when solving problems with large Lipschitz constants.

Considerable insight can be gleaned from a study of the class (2.4) of problems

$$y' = Jy + F(x) .$$

The representation (2.5) of the solution,

$$y(x) = p(x) + \sum_{i=1}^N \alpha_i v_i e^{\lambda_i(x-a)} ,$$

tells us how all integral curves behave qualitatively. Suppose the eigenvalues are ordered so that

$$\operatorname{Re}(\lambda_N) < \dots < \operatorname{Re}(\lambda_1) < \dots < \operatorname{Re}(\lambda_1) ,$$

where $(b-a)\operatorname{Re}(\lambda_N) \ll -1$ and if $\operatorname{Re}(\lambda_1) > 0$, then $(b-a)\operatorname{Re}(\lambda_1)$ is not large.

The constants α_i are determined by the initial conditions. The terms corresponding to "large" $\operatorname{Re}(\lambda_1)$ vanish rapidly as x increases. We might say that the rapidly varying components decay out quickly so that the solution looks smoother and smoother. Of course just when some term is computationally negligible depends on the norm and tolerance used.

Generally we expect to need a small step size to get accuracy when the solution is changing rapidly. This arises from the fact that τ in (6.1) consists of various derivatives of the solution and of f . When the solution changes rapidly, one expects τ to be relatively large, although this is not necessarily so. Here we see that

$$y^{(m)}(x) = p^{(m)}(x) + \sum_{i=1}^N \alpha_i \lambda_i^m v_i e^{\lambda_i(x-a)}$$

There is a region of rapid change near $x = a$ in which all derivatives of $y^{(m)}(x)$ are large and a small step size is necessary. As x increases, the exponential terms dominate the λ_i^m factors so that the solution looks smoother and a larger step size is permissible. It must be understood that if $p(x)$ itself is not easy to approximate, e.g., it is not smooth, then neither is $y(x)$ and one cannot expect $hL \gg 1$ will be possible.

If p_n is the numerical result of solving (2.4) with initial value $p(a)$ and y_n the result with $y(a)$, the difference

$$y_n - p_n = w_n$$

is the result of solving the homogeneous problem with the Runge-Kutta method. Uncoupling the equations again leads us to considering how well $\delta_1'(x) = \lambda_1 \delta(x)$ is approximated by our Runge-Kutta method. We saw earlier that

$$\delta_1(x_{n+1}) = \exp(\lambda_1 h_{n+1}) \delta_1(x_n)$$

and

$$\delta_{1,n+1} = R(\lambda_1 h_{n+1}) \delta_{1,n} .$$

We have already considered when $|R(\lambda_1 h_{n+1})| < 1$ in the stability analysis. We are led to the same consideration again on grounds of accuracy. When a solution component is negligible in the norm used, we want it to stay negligible in the numerical solution, too. The asymptotic expression (6.3) for the truncation error did not distinguish the forward and backward Euler methods. They certainly differ dramatically in the present situation.

In qualitative terms, $hL \gg 1$ is possible only when the solution $y(x)$ is easy to approximate for the method at hand. There is typically a region of rapid change after which the solution smooths out so that it becomes easier to approximate. Just how fast and how smooth the solution becomes depends on the stability properties of the differential equation and on the norm and tolerance supplied by the user. With nonlinear problems the solution may go through other regions of rapid change. Van der Pol's equation undergoing relaxation oscillations is a familiar example.

7. Error Estimator

There are a number of related ways in which large Lipschitz constants pose additional restrictions on the step size. Fundamental to them is the question of approximating $y'(x)$. With some methods this is a very natural task. Indeed, the Adams methods are best viewed as methods for approximating $y'(x)$, rather than $y(x)$. Traditionally the approximation

$$y'_n = f(x_n, y_n) \doteq y'(x_n)$$

is used. The function f is usually evaluated at (x_n, y_n) anyway, so this approximation is often returned by codes as an approximate derivative. The error of the approximation is easily analyzed. For dimensional reasons, the error of the scaled derivative $hy'(x_n)$ is more relevant. Now

$$\begin{aligned} |hy'_n - hy'(x_n)| &= h|f(x_n, y_n) - f(x_n, y(x_n))| \\ &< hL|y_n - y(x_n)| \end{aligned}$$

If we know only that

$$|y_n - y(x_n)| < \mu ,$$

the best we can say is that

$$|hy'_n - hy'(x_n)| < hL\mu .$$

Obviously this approximation is satisfactory only when hL is of modest size. How can we generate a reasonable approximation when the Lipschitz constant is large? One way is to use only approximations to the smooth solution $y(x)$ and so avoid the large derivatives associated with local solutions. For example, a Taylor expansion results in

$$hy'(x_n) = h \left(\frac{y(x_{n+1}) - y(x_n)}{h} \right) - \frac{h^2}{2} y''(x_n) + O(h^3)$$

which suggests the finite difference approximation

$$hy'(x_n) \doteq h \left(\frac{y_{n+1} - y_n}{h} \right) .$$

What is the error of this approximation?

$$\begin{aligned} |h \left(\frac{y_{n+1} - y_n}{h} \right) - hy'(x_n)| &< |(y_{n+1} - y_n) - (y(x_{n+1}) - y(x_n))| \\ &\quad + |(y(x_{n+1}) - y(x_n)) - hy'(x_n)| \\ &< 2\mu + \frac{h^2}{2} |y''(x_n)| + O(h^3) , \end{aligned}$$

if

$$|y_{n+1} - y(x_{n+1})| < \mu \quad \text{and} \quad |y_n - y(x_n)| < \mu .$$

This is a perfectly reasonable approximation even when the Lipschitz constant is large.

The truncation error expressions for the Adams-Moulton formulas and the BDF involve only derivatives of the solution. In typical Adams codes these

derivatives are approximated by divided differences of the $f(x_n, y_n)$ values. As we have seen, these values do not give a good approximation to $y'(x_n)$ and differencing them results in even worse approximations for higher derivatives. If the code should try step sizes too large for stability, these local truncation error estimates are affected; the higher the order of the formula, the more the estimator is affected. Indeed, propagated error due to instability "looks" to the code like a solution which is not smooth. Modern Adams codes, like ODE, consider several orders at a time and select that which appears most efficient. The result is that the order is lowered and the step size reduced until the computation is stable. This is used in ODE to diagnose stiffness when the code does a lot of work. The typical BDF code is intended for problems with large Lipschitz constants, hence must difference the y_n values to obtain reasonable estimates of the truncation error. In some codes this is not obvious because of the representation of the formula, but they are all doing much the same thing.

Although described in a variety of ways, all the popular procedures for estimating local truncation error in explicit Runge-Kutta codes can be viewed as taking each step with two formulas of different orders and estimating the error of the lower order formula by comparison. Obviously this estimate is disturbed if the step size is such that either formula is unstable. The interesting behavior can be used to recognize stiffness. Basically the situation is the same as with the Adams codes. Instability looks like inaccuracy arising from a solution which is not smooth, so the step size is reduced until the computation is stable.

8. Summary

It is assumed that the function f of (1.1) is continuous and satisfies the Lipschitz condition (1.3). This guarantees the initial value problem (1.1, 1.2) has a unique solution $y(x)$. In the classical situation when $L(b-a)$ is not large, this also implies that $y(x)$ is moderately stable. Step-by-step methods for approximating (1.1) numerically select at each step a step size h_{n+1} intended to satisfy the local truncation error requirement (5.2). There are other restrictions on the step size, too, but in the classical situation they cannot be a severe restriction except in the case of very frequent output for some methods and some implementations.

To consider problems with $L(b-a)$ large, we must assume $y(x)$ is stable. Some mathematical conditions guaranteeing this have been stated. Some solution curves must approach $y(x)$ very rapidly so as to cause L to be large, or, as we prefer to put it, in the reverse direction $y(x)$ is very unstable. There is really nothing new unless $h_{acc}L \gg 1$ for the step size h_{acc} yielding the desired accuracy. For this to happen $y(x)$ must be "easy" to approximate. Typically $y(x)$ has a region of rapid change in which $h_{acc}L$ is not large, but h_{acc} increases rapidly as $y(x)$ smoothes out in the course of the integration. All the classical methods must restrict the step size h actually used so that hL is not large in order to integrate even very simple problems stably. There are formulas which seem to have satisfactory stability properties, but they are all implicit. The classical way of evaluating implicit formulas also must restrict h so that hL is not large. This can be avoided by resorting to the expensive simplified Newton iteration. Implicit is the assumption that the Jacobian changes slowly along $y(x)$ so that it need not be evaluated "often." The classical ways the local truncation error is estimated also impose a

restriction that hL be not large. However, there are estimation procedures which do not suffer such restrictions.

Except for the case of too frequent output, a problem is "stiff" in a region for a given code if the step size must be severely reduced from that value which would yield the requested accuracy. The classical situation has no stiff problems. We have described certain characteristics of the mathematical problem which must be present to have stiffness. Unfortunately stiffness also depends on the computational problem, e.g., norm and tolerance. Furthermore, whether a given code exhibits stiffness depends on the particular formula, how it is implemented, and how it relates to the particular mathematical problem. Apparently minor changes can change a severe restriction into none at all!

Notes on Partitioning in the Solution of Stiff Equations

by
Fred T. Krogh

Section 366
Computing Memorandum 488
March 11, 1982

California Institute of Technology
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract

We describe here in rough form an algorithm for solving the linear systems that arise in the solution of stiff systems via the backward differentiation formulas. The algorithm is fairly complicated, but has low computational cost and should be especially attractive for sparse systems. Identification of those equations in the system that can be integrated with the Adams formulas is a part of the algorithm.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under NASA Contract NAS7-100.

Introduction

The solution of a stiff differential equation via the backward differentiation formulas involves at each step the solution of an equation of the form

$$(1) \quad f(t_{n+1}, y_{n+1}) - p'_{n+1} - \alpha(y_{n+1} - p_{n+1}) = 0$$

where the subscript $n+1$ indicates the current step number, y_{n+1} is the numerical solution at the current time, t_{n+1} , to the differential equation

$$(2) \quad y' = f(t, y),$$

p_{n+1} and p'_{n+1} are initial estimates for y_{n+1} and $f(t_{n+1}, y_{n+1})$, and α is a parameter proportional to $1/h$, where h is the stepsize. The solution of (1) is usually obtained from a constant slope Newton method of the form

$$(3) \quad \hat{G} \delta y_{n+1}^{(j)} - r_{n+1}^{(j)} = - [f(t_{n+1}, y_{n+1}^{(j)}) - p'_{n+1} - \alpha(y_{n+1}^{(j)} - p_{n+1})]$$

where $y_{n+1}^{(0)} = p_{n+1}$, $y_{n+1}^{(j+1)} = y_{n+1}^{(j)} + \delta y_{n+1}^{(j)}$, \hat{G} is an approximation to $G = J - \alpha I$, and $J = \delta f / \delta y$ evaluated at $(t_{n+1}, y_{n+1}^{(j)})$.

Iteration (3) has linear convergence with $|\delta y_{n+1}^{(j)}| = O(|\mu|^j)$

where

$$(4) \quad \mu = \text{largest eigenvalue of } \hat{G}^{-1} (\hat{G} - G)$$

In addition, one can use the value of μ as a guide to the number of iterations that are required to maintain stability of the numerical method, see [1] and [2]. The larger μ , the more corrector iterations required, until a point is reached that a new \hat{G} is essential.

Errors in J and in α both contribute to u . Errors in J outside of our direct control are assumed to be zero for the purposes of what we develop here. Of course a code must be prepared to cope with such errors.

At some point the code will obtain a factorization of

$$(5) \quad G_f = J - \hat{\alpha} I$$

and we wish to use that factorization as long as possible. Initially we will have $\hat{\alpha} = \alpha$ (or perhaps, $\hat{\alpha} =$ expected future α) and we wish to keep a useful G as α changes due to changes in the stepsize.

We propose here a factorization that is characterized by having low cost and good fill-in characteristics when being used on problems which have a sparse J . A by-product of the factorization is that equations which can be integrated with explicit methods are identified. What is given here is in a very unfinished form.

We first suggested using an automatic partitioning of stiff systems into equations integrated with Adams methods and equations integrated with BDF methods in [3], and gave some results with such a code in [4]. This early work on automatic partitioning attempted to do this job without requiring a Jacobian matrix. We now believe it was a mistake to attempt so much with so little information. A test suggested by Shampine [5] does a good job in identifying when at least one equation in a system is stiff, and at such a point it seems to us (now) that there is little reason to avoid the calculation of the full Jacobian matrix.

Enright and Kamel [6] give an excellent summary of work that has been done on partitioning. Enright [7] has done some work on investigating stability for algorithms which involve mixing different methods.

An Overview

Row and column interchanges play an important role in the algorithm which we describe, but in order to bring out the essential ideas they are little more than alluded to in this section. An interchange of two rows is frequently accompanied by an interchange of the corresponding two columns thus mapping diagonals to diagonals. The factorization of G_f has the form

$$G_f = LUR$$

where U is upper triangular, and L and R are both unit lower triangular composed of a product of elementary matrix transformations. The total number of elementary matrix transformations in L and R is $\leq N$ (the dimension of G_f) and all are of a different "size". (I.e. the number of rows or columns affected by the transformations are all different.) Thus there is room in the space occupied by G_f to store L, U, and R. For example if $N = 7$, and the factorization consists of removing factors in the order R L L R L R, then the lower triangle has the appearance

```
l
l l
l l l
l l l r
l l r r r
r r r r r r
```

where l indicates an element of L and r an element of R.

Let $\beta = \alpha - \hat{\alpha}$, i.e. (the current value of α)-(the value of α used in forming G_f). We want to define a matrix \hat{G} which is a function of β such that for $\beta \neq 0$ the spectral radius of $\hat{G}^{-1}(G-\hat{G})$ is less than the spectral radius of $G_f^{-1}(G-G_f)$. In addition the factorization of \hat{G} should be readily available from the factorization of G_f . In [1] we examine $\hat{G} = c G_f$

where c is a parameter depending on β . That approach may give a fruitful mix with what we describe below, but such considerations would complicate considerably what we present. The factorization suggested by Enright [8] and [7], might also be useful in combination with what we suggest here.

If diagonals of G_f are mapped to diagonals of U , then we can write

$$G_f = LUR, \quad G = LUR - \beta I, \quad \hat{G} = L(U - \beta D)R$$

$$\hat{G}^{-1}(G - \hat{G}) = \beta^{-1}G(LDR - I)$$

where D is a diagonal matrix selected to minimize $\|LDR - I\|$.

Let $D = \text{diag}(d_k)$, $L = (l_{i,j})$, $R = (r_{i,j})$, where $l_{i,i} = r_{i,i} = 1$, and $l_{i,j} = r_{i,j} = 0$ for $j > i$. Because of the way L and R are constructed, if $j < i < k$ either $l_{i,j} = 0$ or $r_{k,i} = 0$. Thus LDR has the form $D + \bar{L}D + D\bar{R}$ where \bar{L} and \bar{R} are the parts of L and R below the diagonal. Thus the matrix norm is minimized if

$$d_k = 1 / (1 + \|\bar{L}_k\|^2 + \|\bar{R}_k\|^2)$$

where \bar{L}_k is the k -th column of \bar{L} , \bar{R}_k is the k -th row of \bar{R} , and either \bar{L}_k or \bar{R}_k is zero. Observe that the smaller we can make \bar{L} and \bar{R} the smaller the norm of $G - \hat{G}$. Thus if a diagonal can be used as a pivot, we multiply on the left by L if $\|\bar{L}_k\| \leq \|\bar{R}_k\|$, and otherwise multiply on the right by R .

When diagonals in G_f do not map to diagonals in U , things become significantly more complicated. In this case D is permuted and some elements of βD fall in L , some in U , and some in R . The formula given above for d_k is still a good approximation for those elements of βD that fall in U , for those that fall in L or R we use

$$d_k = |u_{k,k}| / (|u_{k,k}|^2 + \|\bar{U}_k\|^2)$$

where \bar{U} is the part of U above the diagonal, \bar{U}_k is the k -th row of \bar{U} , and $u_{k,k}$ is the k -th diagonal of U .

Although the algorithm appears fairly complicated, the actual arithmetic required should usually be less than that required for a simple LU factorization. (We sometimes set matrix elements to zero to save Gaussian elimination steps.) We believe the approach, or some variant, has real promise, but what is given here is perhaps best described as working notes.

Notation and More Details.

In the following section we give an algorithm in a form something like that of SFTRAN3, [9]. We believe that anyone familiar with structured programming will not be bothered by this form of presentation, even if totally unfamiliar with SFTRAN3. This section defines variables used in the algorithm and gives a few more details on why certain choices are made.

A = G_f initially. When done, this area of storage contains U above the diagonal, and information defining L and R below the diagonal.

$a_{i,j}$ The i,j element of A.

s,e Start and end indices for the part of A remaining to be factored. This includes $(a_{i,j})$, $s \leq i \leq e$, $s \leq j \leq e$.

$$r_i = \sum_{j=s}^e |a_{i,j}| - |a_{i,i}|$$

$$c_j = \sum_{i=s}^e |a_{i,j}| - |a_{j,j}|$$

q A flag set as follows.

q = 0 Up to this point there have been only row and column interchanges that take diagonals to diagonals. While q = 0, those equations to be integrated with an Adams method are identified.

q > 0 $a_{q,q}$ did not start out as a diagonal element. Interchanges are always such that there is at most one i , $s \leq i \leq e$ for which $a_{i,i}$ did not start out as a diagonal.

$q=-1$ For $s \leq i \leq e$, $a_{i,i}$, started as a diagonal element. While $q = 0$, and perhaps later, equations are identified which are such that the coupling to the equation, the coupling from the equation, or both can be ignored. This identification is based on the following. If

$$B = \begin{bmatrix} b_{1,1} & r \\ c & \frac{r}{B} \end{bmatrix}, \hat{B}_1 = \begin{bmatrix} b_{1,1} & 0 \\ 0 & \frac{r}{B} \end{bmatrix}, \hat{B}_2 = \begin{bmatrix} b_{1,1} & r \\ 0 & \frac{r}{B} \end{bmatrix}, \hat{B}_3 = \begin{bmatrix} b_{1,1} & 0 \\ c & \frac{r}{B} \end{bmatrix},$$

then μ_1 the spectral radius of $\hat{B}_1^{-1}(B-\hat{B}_1)$ satisfies

$$\mu_1 \leq \rho^{1/2}, \mu_2 \leq \rho, \mu_3 \leq \rho, \text{ where}$$

$$\rho = \|\hat{B}_1^{-1}\| \cdot \|c\| \cdot \|r\| / |b_{1,1}|. \text{ These results follow immediately from an examination of the eigenvalues of } \hat{B}_1^{-1}(B-\hat{B}_1).$$

ρ used for temporary storage of $r_i c_i / |a_{i,i}|$. A computer code would require more care in dealing with potential overflow or divide by zero.

ρ_{\max}, ρ_{\min} maximum and minimum values computed for ρ , $s \leq i \leq e$.

i_{\max}, i_{\min} indices corresponding to ρ giving the ρ_{\max} and ρ_{\min} . If ρ_{\min} is not too large, the diagonal at i_{\min} is used as a pivot. Otherwise, the pivot is $\max_{s \leq k \leq e} \{|a_{i,k}|, |a_{k,i}|\}$ where

$i = q$ if $q > 0$, and otherwise $i = i_{\max}$. It is hoped that the use of $i = i_{\max}$ will lead to smaller off diagonal elements in later steps.

ϵ A bound on relative errors allowed in \hat{G} . Let $M_{m,k}$ denote the values in Table 1 of [1]. $M_{m,k}$ gives a bound on μ , the spectral radius of $\hat{G}^{-1}(\hat{G}-G)$ for a BDF method using a predictor of order k and m iterations on each step. Choose $\epsilon = c M_{j,k}$ where c is a parameter ($\sim 1/2$) to be determined, and j is probably 1, but might be 2 depending on other aspects of the algorithm.

B_A

If at least one direction of coupling can be removed and the associated diagonal of J times h is $> B_A$ then an Adams method is used on that equation. This is justified by arguing that if all is working as it should, then the integration for everything feeding into this equation is stable and hence can be regarded as a function of t , cf. Milne and Reynolds [10]. We regard this test as a first step. It will sometimes use BDF when it should use Adams. (But probably not vice versa.) The Adams method is preferred when it works since it allows a larger order and gives better accuracy. $B_A = -.2$.

B_B

If at least one direction of coupling can be removed and the associated diagonal of J times h is $< B_B$ or if no coupling can be removed, then a BDF method should be used on the equation. $-.6 = B_B \leq 2 B_A$. When coupling can be removed and $B_B < \text{associated diagonal of } J \text{ times } h \leq B_A$, then the method for the equation should not be changed. Algorithms for the switch are given in [11] in the case when the history of the solution is saved as modified divided differences. Formulas for the mix of algorithms are given in [12]. These formulas allow for the direct integration of equations with order greater than one. We believe that such methods will be helpful for mixed differential algebraic systems by sometimes reducing the nilpotency of the system.

B_D

Bound used to decide diagonal is sufficiently large that there is no need to use some element off the diagonal. If for $i = i_{\min}$, $\min(r_i, c_i) < B_D |a_{i,i}|$ then the diagonal is used as a pivot.

α_A

Value to be used for α when equation is integrated using the Adams formulas. See [12] for more details.

$e_i, u_i, v_i, w_i, 1 \leq i \leq N$ This gives information necessary to update A when α or α_A changes. e_i gives an index of a differential equation, u_i, v_i give the row, column indices where the diagonal for equation e_i got mapped, and w_i gives the weight by which a change in β is multiplied in order to update A . Thus changing step size or order require N multiplies in order to update the factorization. Because

permutations are only applied to the active part of A, a row or column may contain several elements that started on the diagonal. Some things would become cleaner if permutations were applied to the entire matrix.

$p_j, x_j \quad 1 \leq j \leq N$

This gives information on the interchanges in the order in which they are done. The row or column index is supplied by p_j and x_j tells what to do with it.

$x_j = 1$

exchange columns s to e of rows p_j and s , then exchange rows s to e of column p_j and s . Increment s .

$= 2$

exchange rows s to e of columns p_j and e , then exchange columns s to e of rows p_j and e . Decrement e .

$= 3$

Same as 1, except ignore henceforth (treat as 0) elements below the diagonal in this column.

$= 4$

Same as 2, except ignore henceforth elements to the left of the diagonal in this row.

$= 5$

Same as 3, except elements to the right of the diagonal in this row are also ignored.

> 10

Let $k=s$ if $p_j > 0$, $k = e$ otherwise. Exchange columns s to e of rows k and $x_j - 10$, then exchange rows s to e of columns k and $x_j - 10$. Then if $p_j > 0$ exchange columns s to e rows p_j and s and increment s . Otherwise exchange rows s to e of columns p_j and e and decrement e .

$\mu, \rho, \rho_M, \rho_\beta, g, \|\hat{G}^{-1}\|$ μ is the current estimate for the spectral

radius of $\hat{G}^{-1}(\hat{G}-G) = g [|\beta| \rho_\beta + \rho_M]$,

$g = \frac{|\text{smallest diagonal of } U \text{ when } G_f \text{ was factored}|}{|\text{smallest diagonal of } U \text{ currently}|}$

We suggest simplifying g by using the smallest diagonal seen since the factorization in the above denominator.

$\beta = \alpha - \hat{\alpha}$ (as usual)

ρ_β gives the linear part (which is most of it) of the change in μ with respect to β . This is a messy one to compute. If we applied row and column interchanges to the whole of A, then we would have

$$G_f = P_L L U R_P$$

$$G = P_L L U R_R - \beta I$$

$$\hat{G} = P_L (L - \beta D_L) (U - \beta D_U) (R - \beta D_R) P_R$$

where P_L and P_R are permutation matrices, and D_L , D_U , and D_R are sparse weighting matrices that are non zero only where the permutations carry them back to diagonals of the original matrix. Thus

$$\frac{d}{d\beta} [G - \hat{G}] = P_L (D_L U R + L D_U R + L U D_R) P_R^{-1} - I$$

We propose to estimate $du/d\beta$ by using a power method on $\hat{G}^{-1} \left[\frac{d}{d\beta} (G - \hat{G}) \right]$ which if permutations were as above, would be

$$P_R^{-1} R^{-1} U^{-1} L^{-1} P_L^{-1} P_L (D_L U R + L D_U R + L U D_R) P_R^{-1} - I$$

With permutations as we have defined them here, the permutations get imbedded in the multiplications by R and L, and to do the multiplications by D_R and D_L one must do the permutations in a similar way, using the information in x_j , p_j , e_i , u_i , v_i , w_i , all defined above. (Note that in doing this power method one does only matrix vector multiplies, and some temporary vector space will be needed.) The power method is used whenever a new matrix is factored, but only a few iterations should be required since no great accuracy is needed. During the power method an estimate for $\|\hat{G}^{-1}\|$ is generated from $\|z_{k+1}\| / \|z_k\|$ where $\hat{G}^{-1} z_{k+1} = z_k$.

ρ_M is a measure of the error purposely introduced into the initial factorization. We have only indicated errors that are introduced to save applying an entire elementary transformation. But a better sparse code should consider dropping any element that changes a zero to a nonzero. Houbak and Thomsen [13] report good results from setting matrix elements to 0 during a factorization for some problems.

$\|\hat{G}^{-1}\|$ an estimate for the norm of \hat{G}^{-1} . Initially the estimate is given by $1/|\alpha|$.

The Algorithm

Usual BDF/Adams stuff

$$\mu = g \cdot (\beta \rho_\beta + \rho_M)$$

If (μ is too big) Then

 Compute new Jacobian, J, if desired.

$$A = J - \alpha I$$

$$\|\hat{G}^{-1}\| = g \|\hat{G}^{-1}\|$$

DO (Factor New Iteration Matrix)

 Compute ρ_β and $\|\hat{G}^{-1}\|$ using power method

$$\rho_M = \|\hat{G}^{-1}\| \rho_M$$

End If

Continue with usual BDF/Adams stuff

(β , g, and A must be updated if α changes.)

Procedure (Factor New Iteration Matrix)

```

q = 0
s = 1
e = N
set  $e_i, u_i, v_i = 1$  for  $1 \leq i \leq e$ 
j = 1
 $\rho_M = 0$ 
DO (Get Row and Column Norms)
Do Until (s > e)
  i = 1
  min
   $x_j = 1$ 
  If ( $r_i < c_i$ )  $x_j = 2$ 
  If ( $\rho_{\min} \| \hat{G}^{-1} \| < \epsilon$ ) Then (Not full coupling)
     $x_j = x_j + 2$ 
     $\rho_M = \max(\rho_M, \rho_{\min} / \epsilon)$ 
    If ( $(\rho_{\min} \| \hat{G}^{-1} \| < \epsilon^2/4)$  & ( $(r_i + c_i) < |a_{i,i}|/10$ )) Then
       $x_j = 5$ 
       $\rho_M = \max(\rho_M, \rho_{\min} / \epsilon^2)$ 
    End If
  If (q = 0) Then (Set for desired Method)
    If ( $h \cdot (a_{i,i} + \alpha) \geq B_B$ ) Then
      If ( $h \cdot (a_{i,i} + \alpha) > B_A$ ) Then
        Set  $e_i$ -th equation for Adams if it is not already.
      End If
      If ( $e_i$ -th equation set for Adams)  $a_{i,i} = a_{i,i} + (\alpha - \alpha_A)$ 
    Else
      Set  $e_i$ -th equation for BDF if it is not already.
    End If
  End If
End If
DO (Fix Matrix)

```

```

Else
  If (q = 0) Then
    Do For k = s to e
      Set equation  $e_k$  for BDF if it is not already.
    End for
  End If
  If (min ( $r_i, c_i$ ) <  $B_D \cdot |a_{1,1}|$ ) Then
    Do (Fix Matrix)
  Else
    (find the pivot)
    m = q
    If (m ≤ 0) m =  $i_{max}$ 
     $x_j = m + 10$ 
     $a_{max} = 0$ 
    Do For k = s to e
      If ( $|a_{k,m}| > a_{max}$ ) Then
        (Max on the column)
         $a_{max} = |a_{k,m}|$ 
        i = k
      End If
      If ( $|a_{m,k}| > a_{max}$ ) Then
        (Max on the row)
         $a_{max} = |a_{m,k}|$ 
        i = -k
      End If
    End For
    DO (Fix Matrix)
  End If
End If
End Until
End Procedure

```

Procedure (Fix Matrix)

$p_j = i$

Do Case (x_j)

Case $x_j = 1$

$k = s$

If ($i \neq k$) Then

DO (Exchange Rows and Columns)

End If

If ($i = q$) Then

$q = -1$

Else

If ($k = q$) $q = i$

$$w_k = |a_{k,k}|^2 / (|a_{k,k}|^2 + c_k^2)$$

End If

Compute and apply elementary matrix transformation which zeros column s in rows $s + 1$ to e using $a_{s,s}$ as a pivot.

$s = s + 1$

DO (Get Row and Column Norms)

Case $x_j = 2$

$k = e$

If ($i \neq k$) Then

DO (Exchange Rows and Columns)

End If

If ($i = q$) Then

$q = -1$

Else

If ($k = q$) $q = i$

$$w_k = |a_{k,k}|^2 / (|a_{k,k}|^2 + r_k^2)$$

End If

Compute and apply elementary matrix transformation which zeros row e in columns $s + 1$ to e using $a_{e,e}$ as a pivot.

$e = e - 1$

DO (Get Row and Column Norms)

Case $x_j = 3$ and $x_j = 5$

$k = s$

```

If (i ≠ k) DO (Exchange Rows and Columns)
If (i = q) Then
    q = -1
Else
    If (k = q) q = i
    wk = 1.
End If
s = s + 1
DO (Update Row and Column Norms)
Case xj = 4
k = e
If (i ≠ k) DO (Exchange Rows and Columns)
If (i = q) Then
    q = -1
Else
    If (k = q) q = i
    wk = 1.
End If
e = e - 1
DO (Update Row and Column Norms)
Case xj ≥ 6
k = s
If (i < 0) k = e
i = xj - 10
If (i ≠ k) Then
    DO (Exchange Rows and Columns)
    If (q = i) q = k
End If
i = pj
If (i > 0) Then
    If (k ≠ q) wk = |ak,i|2 / (|ak,i|2 + ck2)
    If (i ≠ k) Then
        DO (Exchange Rows)
        wi = |ak,k| / (|ak,k|2 + rk2)
        q = i
    Else If (k = q) Then

```



```

    q = -1
  End If
  Compute and apply elementary matrix transformation which zeros
  column s in rows s + 1 to e using  $a_{s,s}$  as a pivot.
  s = s + 1
  DO (Get Row and Column Norms)
Else
  i = -1
  If (k ≠ q)  $w_k = |a_{i,k}|^2 (|a_{i,k}|^2 + r_k^2)$ 
  If (i ≠ k) Then
    DO (Exchange Columns)
     $w_i = |a_{k,k}|^2 (|a_{k,k}|^2 + c_k^2)$ 
    q = i
  Else If (k = q) Then
    q = -1
  End If
  Compute and apply elementary matrix transformation which zeros row
  e in columns s + 1 to e using  $a_{e,e}$  as a pivot.
  e = e - 1
  DO (Get Row and Column Norms)
End If
End Case
j = j + 1
End Procedure

```

Procedure (Exchange Rows)

```
DO For m = s to e
    exchange  $a_{i,m}$  and  $a_{k,m}$ 
End For
Exchange  $e_i$  and  $e_k$ 
 $u' = u_i$ 
 $v' = v_i$ 
 $v_i = v_k$ 
 $u_i = i$ 
If ( $v_k \neq u_k$ )  $u_i = u_k$ 
 $v_k = v'$ 
 $u_k = k$ 
If ( $v' \neq u'$ )  $u_k = u'$ 
End Procedure
```

Procedure (Exchange Columns)

```
Do for m = s to e
    exchange  $a_{m,i}$  and  $a_{m,k}$ 
End For
Exchange  $e_i$  and  $e_k$ 
 $u' = u_i$ 
 $v' = v_i$ 
 $u_i = u_k$ 
 $v_i = i$ 
If ( $v_k \neq u_k$ )  $v_i = v_k$ 
 $u_k = u'$ 
 $v_k = k$ 
If ( $v' \neq u'$ )  $v_k = v'$ 
End Procedure
```

Procedure (Exchange Rows and Columns)

Do for $m = s$ to e

 exchange $a_{i,m}$ and $a_{k,m}$

 exchange $a_{m,i}$ and $a_{m,k}$

End For

Exchange e_i and e_k

$u' = u_i$

$v' = v_i$

If ($v_k \neq u_k$) Then

$u_i = u_k$

$v_i = v_k$

Else

$u_i = i$

$v_i = i$

End If

If ($v' \neq u'$) Then

$v_k = v'$

$u_k = u'$

Else

$v_k = k$

$u_k = k$

End If

End Procedure

Procedure (Get Row and Column Norms)

$\rho_{\max} = 0$
 $\rho_{\min} = \text{big}$ (big is a very large constant)

Do For $i = s$ to e

$$r_i = \sum_{m=s}^e |a_{i,m}| - |a_{i,i}|$$

$$c_i = \sum_{m=s}^e |a_{m,i}| - |a_{i,i}|$$

$$\rho = r_i c_i / a_{i,i}$$

If ($\rho > \rho_{\max}$) Then

$$\rho_{\max} = \rho$$

$$i_{\max} = i$$

End If

If ($\rho < \rho_{\min}$) Then

$$\rho_{\min} = \rho$$

$$i_{\min} = i$$

End If

End For

End Procedure

Procedure (Update Row and Column Norms)

$$\rho_{\max} = 0$$

$$\rho_{\min} = \text{big}$$

Do For $i = s$ to e

$$r_i = r_i - |a_{i,k}|$$

$$c_i = c_i - |a_{k,i}|$$

$$\rho = r_i c_i / a_{i,i}$$

If ($\rho > \rho_{\max}$) Then

$$\rho_{\max} = \rho$$

$$i_{\max} = i$$

End If

If ($\rho < \rho_{\min}$) Then

$$\rho_{\min} = \rho$$

$$i_{\min} = i$$

End If

End For

End Procedure

Acknowledgments

I would like to thank Mac Hyman for suggesting the use of $L(U - \beta I)$ for \hat{G} , Linda Petzold for pointing out that all is not so simple if there are row permutations, Kris Stewart for getting me back in this arena, and finally my supervisor Charles Lawson for giving me a week to prepare this paper when there were things he would much rather have me doing.

References

1. Krogh, F. T. and Stewart, Kris, Asymptotic ($h \rightarrow \infty$) absolute stability for BDF's applied to stiff equations. Submitted to ACM Trans. Math. Software.
2. Klopfenstein, R. W., Numerical differentiation formulas for stiff systems of ordinary differential equations. RCA Review 32, (1971), pp. 447-462.
3. Krogh, F. T., The numerical integration of stiff differential equations. TRW Report No. 99900-6573-R000, TRW Systems, Redondo Beach, CA (March 1968). (Available from the author.)
4. Krogh, F. T., On testing a subroutine for the numerical integration of ordinary differential equations. J. ACM 20, (1973), pp. 545-562.
5. Shampine, L. F., Lipschitz constants and robust ODE codes. Computational Methods in Nonlinear Mechanics, North Holland (1980), pp. 427-449.
6. Enright, W. H. and Kamel, M. S., Automatic partitioning of stiff systems and exploiting the resulting structure. ACM Trans. Math. Software 5 (1979), pp. 374-385.
7. Enright, W. H., The use of partitioning in stiff solvers, Numerical Methods for Solving Stiff Initial Value Problems, Proceedings Oberwolfach, 28.6 - 4.7. 1981, (August 1981).

8. Enright, W. H., Improving the efficiency of matrix operations in the numerical solution of ODE's. ACM Trans. Math. Software 4, (1978), pp. 127-136.
9. Lawson, C. L., SFTRAN3, Programmer's Reference Manual. JPL Document No. 1846-98, Revision A (April 1981).
10. Milne, W. E., and Reynolds, R. R., Stability of a numerical solution of differential equations -- part II. J. ACM 7 (1960), pp. 46-56.
11. Krogh, F. T., Recurrence relations for computing with modified divided differences. Math. Comp. 33 (1979), pp. 1265-1271. Errata, Math. Comp. 35 (1980), p. 1445.
12. Krogh, F. T., Changing stepsize in the integration of differential equations using modified divided differences. Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations, October 1972, Lecture Notes in Math., vol 362, Springer-Verlag, New York (1974), pp. 22-71.
13. Houbak, N. and Thomsen, P. G., SPARKS, A FORTRAN subroutine for the solution of large systems of stiff ODE's with sparse Jacobians. Institute for Numerical Analysis, Technical University of Denmark, DK-2800 Lyngby, Denmark (1979).

STIFF ORDINARY DIFFERENTIAL EQUATIONS
IN CHEMICAL PROCESS ANALYSIS

by

Warren D. Seider
Charles W. White, III
George J. Prokopakis

Department of Chemical Engineering
University of Pennsylvania
Philadelphia, Pennsylvania 19104

April, 1982

ABSTRACT

The physical relation to stiffness is examined in systems with: chemical reactions, heat and mass transfer with reaction, diffusion, and viscous dissipation, and in distillation towers. Physical insights are illustrated to reduce computation times, as compared with integration by a generalized integrator. The advantages and disadvantages of single- and multi-step algorithms are considered.

SCOPE

Since Curtiss and Hirschfelder (1952) introduced the term "stiff" ordinary differential equation (ODE), many authors have referred to a system as stiff when it exhibits widely-spread response times and requires very small step-sizes to maintain numerical stability with some (all explicit and some implicit) integration formulas. In recent years, however, generalized algorithms have been developed to overcome the limitations of numerical stability and refinements in the definition of stiff systems have evolved.

We begin with a brief review of the origin of the term stiff and present those definitions and interpretations that we believe to be most useful. Then, stiffness in the models of several chemical processes is related to the behavior of the physical systems. Next, several approaches to reduce the computation time for integration of both stiff and non-stiff systems, mostly utilizing physical insights, are reviewed.

Finally, we focus on the generalized stiff integrators and offer our views on the advantages and disadvantages of the single- and multi-step algorithms.

CONCLUSIONS

1. A system of ordinary differential and algebraic equations is stiff if and only if numerical integrators with a stability bound must restrict their step-sizes (below that necessary to give desired accuracy) to avoid numerical instability. This restriction can be avoided with A-stable or "stiffly-stable" integra-

tion formulas, but with added computations per step, usually involving evaluation of the Jacobian matrix. When a system is not stiff, integrators with a stability bound can be used without these added computations.

2. σ quantitatively measures the additional computations due to stiffness, as compared with a reference integrator. It appears to be superior to other measures for assessing the degree of stiffness. However, we have not completed evaluating σ for a set of representative systems.
3. As suggested by Shampine and Gear (1979), there is a relation between stiffness and the inherent stability of a system. As the system stabilizes its stiffness increases, and vice versa. This is demonstrated in this paper for:
 - (a) The Belousov reaction system in a limit cycle,
 - (b) coupled heat and mass transfer with reaction in a fluidized bed,
 - (c) viscous dissipation in fluid mechanics, and
 - (d) an azeotropic distillation system.

Furthermore, as steep fronts dissipate, they stabilize and the system of ODEs stiffens. This is demonstrated for (c) and (d).

4. For partial differential equations, discretization with additional grid points gives an improved representation of the system and its eigenvalues. Additional grid points introduce larger negative eigenvalues and the system often becomes stiff.
5. In distillation, as trays are added to increase purity, the response times are increased. As demonstrated by Tyreus and coworkers (1975), $|\operatorname{Re}\{\lambda\}|_{\min}$ decreases more rapidly than $|\operatorname{Re}\{\lambda\}|_{\max}$ and the system stiffens.
6. Most stiff systems experience changes in degree of stiffness, σ , during their life cycles. Some alternate between stiff (step-size limited by a stability bound for explicit and some implicit integrators - $\sigma > 1$) and nonstiff ($\sigma = 1$).
7. Increasingly, engineers and scientists are reporting the importance of physical insights in significantly reducing integration times (compared with generalized multi- and single-step integration methods); in some cases, permitting solutions that otherwise could not be obtained with available computing resources. This is demonstrated
 - (a) for chemical reaction systems, when the fast reactions are in local equilibrium, and the remaining reactions are slow,
 - (b) using approximations that permit analytical integration, giving algebraic equations that express the exponential variation in temperature and composition - avoiding integration with small time-steps to track such fast variables,

- (c) for coupled heat and mass transfer in a fluidized bed, by eliminating the heat and mass balances for the particles as the temperatures and concentrations of the bulk and particle phases approach each other, and
 - (d) for distillation, by assuming $dL/dt = 0$ when it is not necessary to accurately track the liquid flow rates immediately following a disturbance.
8. Classification by rate of change identifies slow variables that may not require integration or may permit integration with a nonstiff integrator.
 9. For reaction systems, the suitability of the pseudo-steady-state approximation must be determined by experimentation.
 10. Steady-state algorithms that use physical insights to gain efficiency in the solution of algebraic equations can be easily coupled with implicit integration algorithms. This is demonstrated for dynamic simulation of distillation towers.
 11. The advantages of our new adaptive semi-implicit Runge-Kutta algorithm (ASIRK - Prokopakis and Seider, 1981), as compared with multi-step algorithms using backward difference formulas are:
 - (a) Larger time steps (for the problems tested),
 - (b) the Nordsieck array is not stored,
 - (c) ASIRK is strongly A-stable, not stiffly-stable,
 - (d) error estimation and step-size adjustment is fairly simple and not coupled to the adjustment of order of accuracy

(using a heuristic methodology), and

(e) ASIRK permits easy expansion and contraction of the number of ODEs in time.

12. The advantages above may not override the following advantages of the multi-step methods:

(a) The Jacobian is evaluated less often and with less accuracy,

(b) the Nordsieck array permits routine printing at even intervals, and

(c) multi-step methods are easily coupled to steady-state algorithms for solution of the algebraic equations.

INTRODUCTION

Curtiss and Hirschfelder (1952) were apparently the first to introduce the term "stiff" differential equation in connection with the mass balances for the free radicals in flames. They explain that: "Free radicals are created and destroyed so rapidly compared to the time-scale for the overall reaction that to a first approximation the rate of production is equal to the rate of depletion. This is the notion of the pseudo-stationary state". However, they continue: "In some cases such as flames and detonations, this approximation is not sufficiently accurate" and note the difficulty in integrating the radical mass balances with ordinary numerical procedures. With this, they introduce backward difference formulas for integration of the so-called stiff differential equations. However, they demonstrate the methodology for integration of a single ordinary differential equation (ODE) with slow and fast response

terms, rather than the mass balances for a reaction system.

Subsequently, many authors have adopted the adjective stiff and added their interpretations. The concise definition of Aiken and Lapidus (1974) is particularly effective: "An m-dimensional system of initial value ODEs

$$\frac{dy(t)}{dt} = f(t, y(t)) ; y(0) = y_0 \quad (1)$$

is called stiff if the local Jacobian ($J = (\partial f / \partial y)$) contains at least one eigenvalue, λ , that does not contribute significantly over most of the domain of interest". It focuses on the exponential terms that contribute to the solution:

$$\begin{aligned} y_1 &= d_{11} e^{\lambda_1 (t-t_i)} + \dots + d_{1m} e^{\lambda_m (t-t_i)} \\ &\cdot \\ &\cdot \\ &\cdot \\ y_m &= d_{m1} e^{\lambda_1 (t-t_i)} + \dots + d_{mm} e^{\lambda_m (t-t_i)} \end{aligned} \quad (2)$$

where λ is a vector of eigenvalues and d_j the eigenvectors of J evaluated at t_i . When $\text{Re}\{\lambda_j\}$ is large negative, its exponential terms decay rapidly and do not contribute significantly.

The problem, of course, with stiff systems is that, to keep the truncation errors bounded (maintain numerical stability), simple explicit (and some implicit) integration methods require a step-size that decreases inversely with the magnitude of the largest negative eigenvalue. Imprac-

tically small step-sizes are required, giving local truncation errors much smaller than necessary, with the possible accumulation of round-off errors that can invalidate the solution. Whereas in the fast transient (non-stiff) regions, when the step-size is reduced to accurately track the rapidly changing variables, further reduction to satisfy the stability bounds is unnecessary.

Then, a system is stiff if and only if numerical integrators with a stability bound (all explicit and some implicit integrators) must restrict their step-sizes to avoid numerical instability. This restriction can be avoided with A-stable or "stiffly-stable" integration formulas (permit any step-size without numerical instability) but with added computations per step, usually involving evaluation of the Jacobian matrix. When a system is not stiff, integrators with a stability bound can be used without these added computations.

The most widely used measure of stiffness is the so-called "stiffness ratio",

$$SR = \frac{|\operatorname{Re}\{\lambda\}|_{\max}}{|\operatorname{Re}\{\lambda\}|_{\min}} \quad (3)$$

Despite its simplicity, however, SR has two important limitations: (1) stiffness is not a function of $|\operatorname{Re}\{\lambda\}|_{\min}$ (although the range of integration, $t_{\text{final}} - t_0$, is often on the order of $1/|\operatorname{Re}\{\lambda\}|_{\min}$), and (2) the system is normally not stiff in rapid transient periods (regardless of $|\operatorname{Re}\{\lambda\}|_{\max}$).

Byrne (1981) suggested a superior measure

$$s = \frac{t_{\text{final}} - t_0}{\tau_{\text{min}}}, \quad (4)$$

"providing the solution is slowly varying on most of the interval", where τ_{min} is the smallest time constant, $\tau_j = -1/\text{Re}\{\lambda_j\}$. Furthermore, to paraphrase Byrne, if τ_{min} is small over a time interval many times longer and the solution is slowly varying on the interval, the system is stiff. However, "if $t_{\text{final}} - t_0 \approx \tau_{\text{min}}$ or the solution is always rapidly varying, the system is not stiff". Eqn. (4) is superior to Eqn. (3), but is weakened by the qualitative requirement that the solution must vary "slowly".

An improved measure is:

$$\sigma = \frac{h_\epsilon}{h_s} \quad (5)$$

where h_ϵ is the step-size to satisfy the local truncation error, ϵ , and h_s is the step-size to satisfy the stability bound for a reference integrator that is not A-stable; i.e.:

$$h_s = \begin{cases} h_\epsilon & h_\epsilon < \delta(r)/|\text{Re}\{\lambda\}|_{\text{max}} \\ \delta(r)/|\text{Re}\{\lambda\}|_{\text{max}} & h_\epsilon \geq \delta(r)/|\text{Re}\{\lambda\}|_{\text{max}} \end{cases}$$

where $\delta(r)$ is the real stability bound for the reference integrator of

order r and h_c is estimated using an integrator of the same order. Of course, σ is more difficult to compute than SR, but σ is a direct measure of the additional computations by the reference integrator due to stiffness and, as such, a measure of the degree of stiffness. An evaluation of σ is currently underway for representative systems.

An additional perspective, concerning the source of stiffness, is offered by Shampine and Gear (1979): "By a stiff problem, we mean one for which...at least some component is very stable (at least one eigenvalue has a real part which is large and negative)". This suggests a relation between stiffness and the inherent stability of a system, which we explore for many chemical processes in the next section.

PHYSICAL RELATION TO STIFFNESS

The coupling of slowly and rapidly changing processes is the most obvious cause of stiffness. Given a slow process, the rapid process introduces large negative eigenvalues in the exponential components of the dependent variables. When the slow process is rate-controlling (all variables respond slowly), the coupled system is stiff. However, given a fast process, the coupled system is stiff only when the dependent variables respond slowly. As examples, fast controllers do not alter the response times of slow processes, but the differential equations that describe the closed loop are stiff. In distillation, some of the vapor and liquid flow rates leaving the trays (small hold-up) continue to vary rapidly when coupled to a slow reboiler (large hold-up). After rapid changes in the flow rates, the coupled system responds slowly and the MESH (Material balance, Equilibrium, Summation of mole fractions,

and Heat balance) equations become stiff.

Chemical Reaction Systems

The dynamics of many chemical reaction systems can be adequately described with reactions involving molecular species. The intrinsic rates of forward and back reactions are relatively slow (as compared with reactions involving free radicals and ionic species) and the mass balances are usually not stiff. Hence, with integrators having a stability bound (so-called "nonstiff" integrators - all explicit and some implicit methods), the step-size is not reduced below that to give desired accuracy.

For some systems, such as in pyrolysis, reforming, and combustion, reactions involving free radicals are necessary to give an adequate representation of conversion in time. These fast reactions introduce large $|\text{Re}\{\lambda\}|$. At times when the concentrations of the molecules and radicals respond slowly, the mass balances are usually stiff; with widely-spread response times they may not be stiff.

Pyrolysis and reforming systems are endothermic and, hence, nearly isothermal in industrial furnaces. In contrast, combustion systems are highly exothermic with rapid rises of temperature and conversion in the flame front. The high sensitivity of reaction rates to temperature, usually described by the Arrhenius equation:

$$k = k_0 e^{-E/RT} \quad (4)$$

causes all reactions to be accelerated. Both $|\text{Re}\{\lambda\}|_{\text{max}}$ and the response times of the concentrations increase and stiffness of the differential

equations depends upon the largest rate of change in concentration.

Limit Cycles

Shampine and Gear (1979) refer to the Van der Pol equation as an example of a limit cycle that is intermittently stiff and nonstiff. Another example is the Belousov reaction system, studied by Prokopakis and Seider (1981) with the kinetic model of Field and Noyes (1974). For a mixture of 1.25M H_2SO_4 , 0.0125M $KBrO_3$, 0.001M $Ce(NH_4)_2(NO_3)_5$, and 0.025M $CH_2(COOH)_2$, Field and Noyes derive the mass balances for the intermediates $HBrO_2$, Br^- , and Ce^{4+} , with dimensionless concentration, y_1 , y_2 , y_3 , respectively:

$$\frac{dy_1}{dt} = 77.27(y_2 - y_1 y_2 + y_1 - 8.375 \times 10^{-6} y_1^2) \quad ; \quad y_1(0) = 4 \quad (6)$$

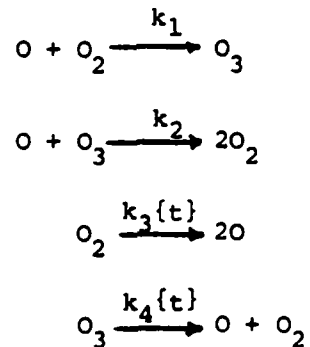
$$\frac{dy_2}{dt} = \frac{-y_2 - y_1 y_2 + y_3}{77.27} \quad ; \quad y_2(0) = 1.1 \quad (7)$$

$$\frac{dy_3}{dt} = 0.161(y_1 - y_3) \quad ; \quad y_3(0) = 4 \quad (8)$$

The period of the limit cycle is 300s as illustrated in Figure 1. Note that the bold curves in Figure 1b denote the fastest changing variable, the so-called "stiff variable". Table 1 shows the eigenvalues of the system which vary by at most four orders of magnitude (and usually far less) during the rapid transients, or steep concentration fronts. In this region, the system is not very stiff; probably a nonstiff integrator could be used without unduely small step-sizes. Then, the system gains

inherent stability, with $|\lambda|_{\max} = 1.33 \times 10^5$. Since the concentrations change more slowly, the system becomes stiff and the stiffness increases as it stabilizes. At longer times, the system slows further and $|\lambda|_{\max}$ decreases as instability sets in, just prior to the rapid transient. Consequently, stiffness decreases with destabilization.

Another limit cycle, of environmental interest, involves the reactions of atomic oxygen, O, oxygen, O₂, and ozone, O₃, in the atmosphere. In the mechanism of Chapman (Dickenson and Gelinas, 1976):



reactions (3) and (4) require photochemical energy and have rate constants that vary diurnally:

$$k_i\{t\} = \begin{cases} e^{-c_i/\sin \omega t} & \sin \omega t > 0 \\ 0 & \sin \omega t \leq 0 \end{cases} \quad i=3,4 \quad (9)$$

where half the frequency of oscillation, $\omega = \pi/43200 \text{ s}^{-1}$ ($=\pi/12\text{hr}^{-1}$) and $c_3 = 22.62$ and $c_4 = 7.601$. These rate constants rise rapidly beginning at dawn ($t = 0, 24, 48\text{hr}, \dots$), reaching a peak at noon ($t=6, 30, 54\text{hr}, \dots$),

and decreasing to zero at sunset ($t=12,36,60\text{hr},\dots$). The mass balances for O and O_3 are:

$$\frac{dy_1}{dt} = -k_1 y_1 y_3 - k_2 y_1 y_2 + 2k_3 \{t\} y_3 + k_4 \{t\} y_2 ; y_1 \{0\} = 10^6 \text{cm}^{-3} \quad (10)$$

$$\frac{dy_2}{dt} = k_1 y_1 y_3 - k_2 y_1 y_2 - k_4 \{t\} y_2 ; y_2 \{0\} = 10^{12} \text{cm}^{-3} \quad (11)$$

where y_1, y_2, y_3 are concentrations of O, O_3 , and O_2 , respectively, and $k_1 = 1.63 \times 10^{-16}$ and $k_2 = 4.66 \times 10^{-16}$. When integrated with $y_3 = 3.7 \times 10^{16} \text{cm}^{-3}$, O traverses a limit cycle while O_3 builds just prior to noon and remains constant at other times, as illustrated in Figure 2. Note that y_1 drops from 10^6cm^{-3} initially to virtually zero ($< 10^{-30} \text{cm}^{-3}$) within one minute and builds to 2.92×10^{-2} in one hour. Similarly, y_1 drops from $3.15 \times 10^{-2} \text{cm}^{-3}$ after 11 hours to virtually zero after 12 hours, where it remains until 24 hours. At 24 hours, the rapid rise begins. The rates of change and eigenvalues are given in Table 2. λ_1 is nearly constant and λ_2 reflects the rate of change of the system. $|\lambda_2|$ increases toward noon as the system accelerates and decreases toward sundown as the system decelerates. Since the rates of change are very slow at night, the system is very stiff. However, throughout most of the daytime, with rapid rates of change, the system is nonstiff.

Heat and Mass Transfer with Reaction

Models that couple heat and mass transfer between a bulk fluid phase and a solid catalyst are justified when the resistances to heat and mass

transfer are important. Such a model is presented by Luss and Amundson (1968) to represent the dynamics of a batch fluidized-bed reactor where external resistances are significant. The mass and heat balances for the bulk phase are:

$$\frac{dp}{d\tau} = p_e - p + H_g(p_p - p) \quad ; \quad p\{0\} = p_e \quad (12)$$

$$\frac{dT}{d\tau} = T_e - T + H_T(T_p - T) + H_w(T_w - T) \quad ; \quad T\{0\} = T_e \quad (13)$$

and for the reacting catalyst:

$$A \frac{dp_p}{d\tau} = -H_g K k p_p + H_g(p - p_p) \quad ; \quad p_p\{0\} = p_p^o \quad (14)$$

$$C \frac{dT_p}{d\tau} = H_T F K k p_p + H_T(T - T_p) \quad ; \quad T_p\{0\} = T_p^o \quad (15)$$

where T and p are the temperature and partial pressure of species A in the bulk phase; T_p and p_p , the particle properties; T_e , p_e , the entrance conditions; and k , the rate constant for a first-order reaction, $A \rightarrow B$. The results of Luss and Amundson (1968) are illustrated in Figure 3. Using the explicit Runge-Kutta Gill integration formula, the step-size is small initially and decreases as $T_p \rightarrow T$ and $p_p \rightarrow p$, with decreasing rates of change. Because $|\lambda|_{\max}$ increases, as shown above, stiffness increases with stabilization.

$$\lambda_j = -[2(n+1)\sin\{j\pi/[2(n+1)]\}]^2 \quad (18)$$

and

$$\lambda_1 = -\pi^2 \quad ; \quad \lambda_{n-1} = -4(n+1)^2$$

which correspond to the eigenvalues of the diffusion equation, $-(j\pi)^2$, $j = 1, 2, \dots$. They note: "The first eigenvalue of the discretized system is approximately the first eigenvalue of the differential operator, and the others are approximations to some of the larger ones". This points out that "stiffness is inherent in the problem (we prefer model), not part of the method of solution".

Viscous Dissipation

The dimensionless momentum balance with viscous dissipation in one spatial dimension:

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} + \mu \frac{\partial^2 u}{\partial x^2} \quad (19)$$

with the boundary conditions of Burger

$$u(t, 0) = \phi(t, 0)$$

$$u(t, 1) = \phi(t, 1)$$

$$u(0, x) = \phi(0, x)$$

has the analytical solution:

$$\phi(x, t) = \frac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}} \quad (20)$$

with

$$A = \frac{0.5}{\mu} (x - 0.5 + 4.95t)$$

$$B = \frac{2.5}{\mu} (x - 0.5 + 0.075t)$$

$$C = \frac{5}{\mu} (x - 0.375)$$

Equation (19) has been integrated with the method of lines, using a five-point central difference approximation for $\partial^2 u / \partial x^2$ and a third-order four-point, upwind finite difference approximation for $\partial u / \partial x$. Such an approximation is found to improve the tracking of step waves when the viscosity, $\mu = 0$ (Carver, 1976 and Hu and Schiesser, 1981)

With the grid spacing $\Delta x = 0.01$, we integrated the ODEs to give the solution in Figure 4, which compares favorably with the analytical solution. Of course, as μ increases, viscous dissipation spreads the fronts as shown.

Table 4 shows that $|\text{Re}\{\lambda\}|_{\max}$ increases as μ increases $(\frac{du_i}{dt})_{\max}$ decreases in the less steep fronts and the system gains inherent stability.

Once again, we observe that stiffness increases with stabilization.

Distillation

It is well-known that feedback of mass and energy more closely couples slow and fast process units in recycle loops. This is exemplified in the countercurrent cascade of vapor/liquid separators (or trays) of a dis-

tillation tower, as illustrated for the separation of propane from n-butane in Figure 5. The trays (small hold-up) respond rapidly to changes in flow rate, but the large reboiler slows the coupled response. After the initial rapid transients, the MESH ODEs become stiff as compositions and temperatures respond slowly. For separation of propane from n-butane, the Jacobian of the mass balances:

$$\frac{dx_j}{dt} = G_j x_j + dx_j \quad j=1, \dots, C \quad (21)$$

gives a measure of the stiffness. In Eqn. (21), $x_j = [x_{1j}, \dots, x_{Nj}]^T$, where x_{ij} is the mole fraction of compound j in the liquid on Tray i . N is the number of trays and C the number of chemical species. G_j is the tri-diagonal coefficient matrix due to the coupling of Tray i to Trays $i+1$ and $i-1$ (as illustrated in the Appendix).

It is interesting that as the product purities approach unity (e.g., $x_{N, \text{propane}} \rightarrow 1$ and $x_{1, \text{n-butane}} \rightarrow 1$), N increases and the mole fractions respond more slowly. This is demonstrated by Tyreus and co-workers (1975) who show that $|\lambda|_{\min}$ decreases more rapidly than $|\lambda|_{\max}$. Hence, for difficult separations (with large pinch zones - adjacent trays with small concentration differences), the MESH equations become stiffer as the response times increase. This effect is also demonstrated for decreasing relative volatility, $\alpha_{jr} = K_j/K_r$.

Azeotropic Distillation

Typical operating conditions for an azeotropic distillation tower to dehydrate alcohol are shown in Figure 6. The trays in the upper portion of the stripping section have very small changes in alcohol and entrainer concentration while water is removed. Then, in a few trays, the alcohol mole fraction is increased to near unity as the entrainer is eliminated. These trays respond more rapidly than others in the stripping section, as illustrated in Figure 7, which shows the profiles of liquid mole fractions after a thirty percent increase in the feed flow rate.

REDUCTION IN COMPUTATION TIME

Many models, involving ordinary differential and algebraic equations, are stiff during part of their life cycles. Consequently, modelers are turning to the generalized multi- and single-step methods for integrating systems of stiff ODEs. Although these methods are successful for many systems, increasingly engineers and scientists are reporting the importance of physical insights in significantly reducing computation times; in some cases, permitting solutions that otherwise could not be obtained with available computing resources. It is noteworthy, however, that some approaches, presented in the name of eliminating stiffness,

rather reduce the computation time for nonstiff systems. There appears to be confusion regarding the properties of stiff systems which, hopefully, the previous sections have helped to clear-up.

In this section, we consider a number of approaches for reducing computation times.

Linearization

For systems that are mildly nonlinear, linearization from time-to-time can offer an efficient solution method. At appropriate times, t_i , the linear approximation is:

$$\dot{\underline{y}} = \underline{J}\underline{y} \quad ; \quad \underline{y}(t_i) = \underline{y}_i \quad (22)$$

The Jacobian, \underline{J} , is computed, as well as its eigenvalues, $\lambda_1, \dots, \lambda_m$, and their corresponding eigenvectors, \underline{d}_j , $j=1, \dots, m$. Then, the analytical solution, Eqn. (2), is used for $t_i < t \leq t_{i+1}$, where the step-size, $h = t_{i+1} - t_i$, within which acceptable accuracy is obtained, is determined by the nonlinearities. Mah and coworkers (1961) applied this repeated linearization with some success for integration of the mass balances in distillation. However, the approach has not been widely adopted, primarily due to difficulties in adjusting the step-size and the complexity of calculations to determine \underline{J} and its eigenvalues and eigenvectors.

Of course, for linear systems, such as networks of first-order reversible reactions among isomers (Wei and Prater, 1962), Eqn. (2) is the analytical solution.

Classification by Rate of Change

For systems that exhibit widely-spread rates of change (usually not stiff, unless $|\operatorname{Re}\{\lambda\}|_{\max}$ is sufficiently large to require that a nonstiff integrator reduce its step-size below that necessary to accurately track the fastest variables), significant savings in integration time can often be achieved by classifying the variables according to rate of change. Let the m ODEs be listed in order of descending magnitude of rate of change:

$$\begin{aligned} \dot{y}_1 &= f_1(t, \underline{y}) \\ \dot{y}_2 &= f_2(t, \underline{y}) \\ &\vdots \\ &\vdots \\ \dot{y}_m &= f_m(t, \underline{y}) \end{aligned} \tag{23}$$

where $|\dot{y}_1| \gg |\dot{y}_m|$. Some authors refer to y_1 as the "stiff" variable (Aiken and Lapidus, 1974; Prokopakis and Seider, 1981) and y_1, \dots, y_n (where n is arbitrary) as the stiff variables with y_{n+1}, \dots, y_m as "nonstiff" variables. Since the system may not be stiff, we prefer to replace these with the adjectives "fast" and "slow".

In some cases, it may not be necessary to accurately track the fast variable(s). If not, their derivatives can be approximated, say, with a first-order finite difference or as zero (pseudo-steady-state assumption), resulting in algebraic equations. The resulting ODEs (with the slower rates of change) permit larger step-sizes to accurately track

the slower variables. However, with rates of change in closer proximity, they are more likely to be stiff, requiring an implicit or semi-implicit algorithm with evaluations of the Jacobian, \underline{J} . Normally, the increase in h reduces computation time far more than that added to evaluate \underline{J} .

This is demonstrated for a 12 tray azeotropic distillation tower to dehydrate isopropanol with cyclohexane, as illustrated in Figure 8. The primary feed is on Tray 9 and the reflux from the decanter is assumed to remain constant. Initially, Trays 10-12 are at the feed composition, and Trays 1-9 have a linear decrease in mole fraction of cyclohexane with the ratio of isopropanol to water equal to that in the feed. Also, initially $L_1^0 = 85$ mol/min and $V_1^0 = 566.7$ mol/min and the remaining L_i and V_i are those at constant molal overflow. The simulation begins holding the boil-up ratio, V_1/L_1 , fixed at 6.67. Cyclohexane is initially concentrated at the bottom of the tower and the dynamic response involves an inversion in concentration profiles. The results of two integrations are compared in Table 5, one with \underline{x} and \underline{L} as state-variables, (Eqns. (21) and (A10)), the other assuming $d\underline{L}/dt = 0$ (Prokopakis and Seider, 1982). As expected, very small time-steps are necessary to accurately track \underline{L} . With the pseudo-steady-state assumption, after 0.075 min \underline{L} begins small oscillations about the true solution, whereas the mole fractions are in good agreement throughout the integration. After 23 min, the liquid flow rates agree to five significant figures. In many situations, these small oscillations in flow rates in the early transient are insignificant over the 2-3 hours to achieve a steady-state. Note that a first-order finite difference approximation for $d\underline{L}/dt$ gives intermediate accuracy.

Shampine and Gear (1979) state: "Current codes for stiff differential equations are sufficiently efficient that there is no need to consider such model changes for most problems for reasons of cost, and there are excellent reasons of convenience and theoretical support for not changing the model. To be sure, there are exceptions because we are discussing general purpose codes". We believe that the MESH equations in distillation constitute such an exception.

In other cases, the slowest variables may be changing too slowly to require integration or when the rate of change is significant, it may be possible to integrate their ODEs with a nonstiff integrator (if $|\operatorname{Re}\{\lambda\}|_{\max}$ is sufficiently small). Such decompositions are recommended by Prokopakis and Seider (1982) for integration of the MESH equations in distillation and by Edsberg (1980) for integration of the mass balances in reaction systems.

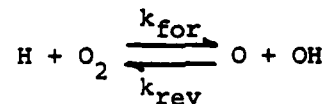
For pyrolysis, reforming, and combustion models, where the need to include the fast reactions involving free radicals is generally acknowledged, there is question concerning the applicability of the pseudo-steady-state assumption (PSSA) for radical concentrations. Blakemore and Corcoran (1969) postulate a free radical mechanism for the pyrolysis of n-butane and integrate the mass balances with and without the PSSA. After 10 msec in a batch reactor at 519°C they show negligible difference in the radical concentrations, with no changes in the radical concentrations thereafter. With the PSSA, the step-size for their nonstiff integrator is increased significantly, presumably because the resulting ODEs are nonstiff.

However, others question the suitability of the PSSA. Sundaram and Froment (1978) studied the pyrolysis of propane with a free radical mechanism at 800°C and found significant errors in the PSSA. Furthermore, Edelson and Allara (1973) report that, using the model of Herriott and coworkers (1972) for propane pyrolysis, three of the radicals do not achieve a steady-state simultaneously and H· atom does not achieve a steady-state at all.

It seems clear that the PSSA is appropriate for some reaction systems and not for others. Unfortunately, the rates of change of the fast forward and reverse reactions, or any other measure (to our knowledge), does not indicate its appropriateness. The suitability of the PSSA must be judged by experimentation. Since this involves integrating the unabridged ODEs, it is pointless to make the assumption unless repeated integrations are planned with small changes in the parameters.

Chemical Equilibrium

In chemical reaction systems, it may be possible to reduce computation times if some of the reactions are in local equilibrium (i.e., have equivalent forward and back reaction rates); for example, if the reaction:



is in local equilibrium, the forward and reverse rates of reaction are equal:

$$r_{\text{for}} = k_{\text{for}} c_{\text{H}}^c c_{\text{O}_2}^c = r_{\text{rev}} = k_{\text{rev}} c_{\text{O}}^c c_{\text{OH}}^c \quad (24)$$

and

$$K = \frac{k_{\text{for}}}{k_{\text{rev}}} = \frac{c_{\text{O}}^c c_{\text{OH}}^c}{c_{\text{H}}^c c_{\text{O}_2}^c} \quad (25)$$

where K is the chemical equilibrium constant.

Sorensen and Stewart (1980) present a method to determine an independent set of mass balances when a subset of the reactions can be taken at equilibrium. They identify R basic species, where R is both the number of independent reactions and mass balances. The R basic species are divided arbitrarily into A and B subsets. There are NB of the B species, one for each of an independent set of reactions taken at equilibrium. There are NA of the A species, one for each of R - NB reactions not taken at equilibrium that form an independent set with the reactions taken at equilibrium. Then, with row eliminations, NA mass balances are derived that do not involve the rates of reactions taken at equilibrium.

White and Seider (1981) apply this method to a system of 21 reactions for combustion of H₂ and CO in air, with 7 reactions taken at equilibrium. The 8 independent mass balances are reduced to 4, since there are 4 independent reactions at equilibrium, and have the form:

$$\frac{d\zeta}{dt} = \underline{f}(\underline{r}, \underline{n}) \quad (26)$$

where $\underline{\zeta}$ is a linear combination of concentrations (e.g., $\zeta_1 = c_{\text{NO}} + c_{\text{H}} + c_{\text{H}_2} - c_{\text{O}_2}$, $\zeta_2 = c_{\text{OH}} + 2c_{\text{O}_2} + c_{\text{H}} + 2c_{\text{O}}, \dots$) and \underline{r}_n is a vector of net rates of reaction for reactions not taken at equilibrium. White and Seider use the GEAR integrator to give $\underline{\zeta}$ across a time-step. Then, the concentrations, \underline{c} , are computed using the Rand method (Newton's method) to minimize the Gibbs free energy given $\underline{\zeta}$. For this system, computation times are not reduced primarily because the reactions not taken at equilibrium are fast as well as slow. They conclude that computation times should be reduced when all reactions not at equilibrium are relatively slow, reducing the rates of change, $d\underline{\zeta}/dt$, and probably the stiffness of Eqn. (26). This should permit large time-steps and possibly the use of a nonstiff integrator. This has been demonstrated for a very small system of marginal interest.

Near-Analytical Integration

Often the responses of process variables, such as temperatures and compositions, have a familiar form, approaching exponential variations, for example. In these cases, it may be possible to obtain algebraic equations that contain this functionality through analytical integration with the appropriate numerical approximations and, in this manner, avoid numerical integration with small time-steps in tracking these fast variables.

As an example, consider Kayihan's model for reacting polydispersed particles which he illustrates for an entrained bed reactor to devola-

tilize coal (1980). Figure 9 is a schematic of the reactor into which a slurry of pulverized coal in carrier gas (at temperature, T_{Go} , gas flow rate, G_o , and mass flow rates of particles in K discrete size ranges, m_1, \dots, m_K) is mixed with hot gas (at temperature, T_{Fo} , and flow rate, F_o). As the coal particles move through the reactor, they are heated by convection and devolatilization occurs.

The model assumes a uniform density and heat capacity in all solid particles and uniform heat capacity in the gas phase. The energy balance for particles in each size range, j , is

$$m_j c \frac{dT_j}{dt} = a_{sj} h (T_F - T_j) + \frac{dm_j}{dt} [(C - c)T_j + \Delta H_R] \quad (27)$$

where T_j does not vary with particle radius and a_{sj} is the area for convective heat transfer. It is assumed that N first-order reactions, i , remove the volatile species from each size range:

$$\frac{dv_{ij}}{dt} = k_{oi} (v_i^* - v_{ij}) e^{-E_i/RT_j} \quad \begin{matrix} i=1, \dots, N \\ j=1, \dots, K \end{matrix} \quad (28)$$

where v_i^* is the maximum volatiles produced by reaction i in grams per gram of solid initially and v_{ij} is related to m_j by the mass balances

$$m_j = m_{oj} (1 - \sum_i v_{ij}) \quad j=1, \dots, K \quad (29)$$

For devolatilization of Montana lignite, Kayihan uses kinetic parameters of Suuberg (1977) for 15 parallel reactions to produce 8 volatile species. With $K = 10$, Eqns. (27) - (29) give 160 ODEs to be solved with

the algebraic mass and energy balances in the gas phase. The particle temperatures range from 500°K (for the larger particles) to 1000°K and with a wide range of reaction rates the differential equations are expected to be stiff. Kayihan estimates a stiffness ratio ($|\lambda|_{\max}/|\lambda|_{\min}$) equal to 10^{24} , but this is a poor measure of stiffness.

This led him to use the DVOGER stiff ODE integration routine in the IMSL Library (1977), which uses the GEAR algorithm. The 160 stiff ODEs were integrated, using analytical expressions for the Jacobian elements, over one-one hundredth of the residence time for the largest particles and projected to require over 10 hours on a CDC CYBER 73 computer. This was primarily due to the excessive operations with zero elements in LU factorization of a sparse 160 x 160 Jacobian matrix and the subsequent elimination and substitution steps. A significant reduction in operations would be expected using GEARS with its routines for efficient inversion of sparse matrices. But, Kayihan did not use this generalized algorithm.

Instead, he integrated Eqns. (27) and (28) analytically with numerical approximations and obtained algebraic equations that preserve the exponential variations in temperature and conversion with time.

The energy balances for particles in size range j , Eqn. (27) are re-written:

$$\alpha_j \frac{dT_j}{dt} + T_j = T_F + \beta_j \frac{dm_j}{dt} \quad (30)$$

where

$$\alpha_j = \frac{m_j c}{a_{sj} h}, \quad \beta_j = \frac{(C - c) T_j + \Delta H_R}{a_{sj} h}$$

An integrating factor, $\exp\{t/\alpha_j\}$, is introduced:

$$d\left\{e^{t/\alpha_j} T_j\right\} = \frac{e^{t/\alpha_j}}{\alpha_j} \left(T_F + \beta_j \frac{dm_j}{dt}\right) dt \quad (31)$$

and Eqn. (31) is integrated analytically from t_{n-1} to t_n assuming

$$T_F = (T_{F_{n-1}} + T_{F_n})/2, \quad \alpha_j = (\alpha_{j_{n-1}} + \alpha_{j_n})/2, \quad \beta_j = (\beta_{j_{n-1}} + \beta_{j_n})/2, \text{ and}$$

linear variation of $m_j \exp\{t/\alpha_j\}$ with t to give:

$$T_j\{t_n\} = T_j\{t_{n-1}\} e^{-\Delta t/\alpha_j} + T_F\{t_n\} [1 - e^{-\Delta t/\alpha_j}] \\ + \frac{\beta_j}{\alpha_j} [m_j\{t_n\} (1 - \frac{\Delta t}{2\alpha_j}) - m_j\{t_{n-1}\} (1 - \frac{\Delta t}{2\alpha_j}) e^{-\Delta t/\alpha_j}] \quad (32)$$

Similarly, Eqns. (28) are integrated analytically:

$$v_{ij} = v_i^* \left[1 - e^{-k_{oi} \int_0^t e^{-E_i/RT_j} dt} \right] \quad (33)$$

or

$$v_{ij} = v_i^* [1 - e^{-k_{oi} \sum_{m=1}^n \phi_{ij}(t_m)}] \quad (34)$$

where

$$\phi_{ij}(t_m) = \int_{t_{m-1}}^{t_m} e^{-E_i/RT_j} dt \quad (35)$$

and Eqn. (35) is integrated assuming linear variation of T_j between t_{m-1} and t_m . The resulting algebraic equations are solved simultaneously with Eqns.(32), and the mass and energy balances for the gas phase to compute profiles of all unknowns in time. The equations are decoupled with guesses for the profiles of particle temperatures to avoid operations with large sparse matrices.

Using a uniform Δt ($= 0.01 t_{max}$), 4-5 iterations gave three significant figures in less than 30 seconds of CDC CYBER 73 computer time. Whereas the stiff integrator is very convenient to use, this is illustrative of the many problems for which physical insights lead to specialized algorithms that are more efficient, in some cases permitting a solution that could otherwise not be obtained with available computing resources.

Application of Steady-state Algorithms

Many models involve coupled ordinary differential and algebraic

equations, with the ODEs reducing to algebraic equations in the steady-state. Since steady performance is the usual objective for continuous processing systems, an extensive collection of algorithms has been developed for design and evaluation of operating strategies in the steady-state. These often solve large sets of algebraic equations using specialized algorithms that incorporate physical insights to gain efficiency.

However, when the GEAR multi-step algorithm is used to integrate stiff ODEs

$$\frac{dy}{dt} = \underline{f}(t, \underline{y}) ,$$

backward difference formulas:

$$\underline{y}_{n+1} = \sum_{\ell=0}^{r-1} \alpha_{\ell} \underline{y}_{n-\ell} + h \beta_{-1} \underline{f}(t_{n+1}, \underline{y}_{n+1}) \quad (36)$$

must be solved for \underline{y}_{n+1} during each time step, where r is the order of accuracy of the formulas. Of course, these nonlinear algebraic equations are very similar to the steady-state equations:

$$\underline{f}(\underline{y}) = 0 \quad (37)$$

Hence, they can be solved using the steady-state algorithm with minor modifications. The GEAR subroutine STIFF is also modified to transfer the summation term:

$$\sum_{\ell=0}^{r-1} \alpha_{\ell} \underline{y}_{n-\ell}$$

and the step-size, h , and β_{-1} to the modified steady-state algorithm, which solves the entire set of algebraic equations for y_{n+1} (and the other variables) and returns to the GEAR integrator.

This technique was introduced for distillation towers by Boston and coworkers (1981). They used the GEAR integrator to integrate Eqn. (21), the energy balance (ignoring the sidestreams):

$$\begin{aligned} \frac{dh_i^L}{dt} = \frac{1}{M_i} [& L_{i+1} (h_{i+1}^L - h_i^L) - V_i (h_i^V - h_i^L) \\ & + V_{i-1} (h_{i-1}^V - h_i^L) + F_i (h_i^F - h_i^L) + Q_i] \quad i=1, \dots, N \end{aligned} \quad (38)$$

and the overall mass balance (ignoring the sidestreams):

$$\frac{dM_i}{dt} = L_{i+1} + V_{i-1} + F_i - L_i - V_i \quad i=1, \dots, N \quad (39)$$

These equations and associated algebraic equations are derived in the Appendix. Rather than allow the GEAR algorithm to solve Eqn. (36), with the algebraic MESH equations, Boston and coworkers prefer to use their efficient RADFRAC program with modifications. RADFRAC solves the MESH equations in the steady-state using the Boston and Sullivan algorithm (1974). This algorithm calculates K-values and enthalpies accurately in an outer loop only. The MESH equations are solved in the inner loop using approximate models for K-values and enthalpies and, roughly speaking, holding relative volatilities constant. Far

fewer iterations of the outer loop are necessary, as compared with solution using a Newton-Raphson method, and hence fewer accurate evaluations of K-values and enthalpies (very time-consuming calculations in non-ideal distillation towers) are necessary.

Model Simplification

In some of the previous approaches (e.g., near-analytical integration and application of steady-state algorithms), physical attributes of the processes permit us to improve the efficiency and reliability of the numerical methods without simplification of the model. In other cases, the model is simplified; for example, using approximations for dL/dt in distillation when it is not necessary to accurately track L during the fast transients following a disturbance.

It is occasionally possible to simplify a model, to reduce its stiffness, and permit integration with a nonstiff integrator, with no loss in accuracy. The equations of Luss and Amundson (1968), Eqns. (12) - (15), comprise such a model. As noted previously, at long times $T_p \rightarrow T$ and $p_p \rightarrow p$, with decreasing rates of change, and the system stiffens because $|\lambda|_{\max}$ increases with stabilization. When the temperatures and partial pressures differ by less than 1 percent, stiffness sets in ($|\lambda|_{\max} = 2187$). To permit efficient integration with an explicit integrator, Luss and Amundson set $T_p = T$ and $p_p = p$ and add Eqns. (12) and (14) and Eqns. (13) and (15), giving two less-stiff ODEs ($|\lambda|_{\max} = 1.242$),

$$(1 + A) \frac{dp}{dt} = p_e - p - H_g Kkp \quad (40)$$

$$(1 + C) \frac{dT}{dt} = T_e - T + H_w (T_w - T) + H_T FKkp \quad (41)$$

which are integrated with a step-size increased by a factor of 100.

SINGLE-STEP AND MULTI-STEP ALGORITHMS - ADVANTAGES AND DISADVANTAGES

Much of the recent literature places emphasis on the multi-step algorithms using the backward difference formulas popularized by Gear (1971) and Hindmarsh (1974). The single-step methods, such as semi-implicit Runge-Kutta, receive less publicity, probably because the Jacobian must be evaluated accurately during each time-step. However, there are an increasing number of factors that favor usage of single-step methods, and collectively, for some problems, these may override the factors that favor the multi-step methods. These are itemized below:

1. A new adaptive semi-implicit Runge-Kutta algorithm (Prokopakis and Seider, 1981), in limited testing, appears to be competitive with the GEAR program (Hindmarsh, 1974). Larger time-steps are taken, but the computation time per step is somewhat larger, mostly due to more frequent evaluation of the Jacobian.
2. Storage of the 6xm Nordsieck array (or the equivalent Y_{i-1}, \dots, Y_{i-6}) is avoided in the single-step methods,

but interpolation for printing at even intervals is less accurate - sometimes requiring integration with a reduced time-step.

3. The semi-implicit Runge-Kutta (SIRK) methods are A-stable and the adaptive algorithm adjusts γ_∞ (the characteristic root as $h|\text{Re}\{\lambda\}|_{\max} \rightarrow \infty$) as a function of $h\bar{\lambda}_s$, where $\bar{\lambda}_s$ is the pseudo-eigenvalue given by

$$\frac{dy_s}{dt} = \bar{\lambda}_s y_s \quad (42)$$

and s is the fastest variable or "stiff" variable. γ_∞ is adjusted to give an exact solution to Eqn. (40) and to increase the accuracy of a second-order SIRK formula for the system of ODEs. When $h\bar{\lambda}_s$ is large, $\gamma_\infty \rightarrow 0$ to give strong A-stability. Whereas, the backward difference formulas of order one to five are just "stiffly-stable" and can be unstable for λ near the imaginary axis.

4. For the semi-implicit Runge-Kutta methods, step-size adjustment can be accomplished using imbedded formulas for estimation of truncation errors and an efficient extrapolation method (Prokopakis and Seider, 1981). Whereas, in the multi-step methods, the step-size and order of method are adjusted simultaneously and according to Shampine and Gear (1979):

"While the basic strategy is straightforward - the two are chosen to try and minimize the amount of work done to integrate over the interval - the implementation is not. The problem lies in the fact that there is not yet an adequate theory to tell us how to choose these parameters".

5. Finally, in some problems it is desirable to expand or contract the number of ODEs in time. This is difficult to accomplish with multi-step methods due to the need to interpolate or extrapolate the derivatives of the Nordsieck array. Whereas, in single-step algorithms no historical information need be updated.

Many of these advantages may not override two key advantages of the multistep methods:

1. The Jacobian is evaluated less frequently and need not be evaluated accurately as it is only used to solve the corrector equations.
2. With the Nordsieck array, printing at even intervals is accomplished routinely. Single-step methods require approximate interpolation or integration with a smaller time-step.

Another important advantage is:

3. Multi-step methods can be coupled to steady-state algorithms

for solution of the algebraic equations. This cannot be accomplished, to our knowledge, with single-step, non-iterative methods.

The advantages of these methods should be weighed for each system.

APPENDIX

This Appendix briefly reviews the typical model for the dynamic simulation of a distillation tower. More details are given by Prokopakis and Seider (1982).

The usual nomenclature is shown in Figure 10, a schematic of a distillation tower, where the general Tray i has a feed stream, vapor and liquid sidestreams, and heat transfer. Note that s_i^V is the fraction of vapor leaving Tray i in the sidestream and s_i^L corresponds for the liquid sidestream.

The usual model involves the following assumptions:

- (1) The vapor and liquid streams leave the trays at equilibrium,
- (2) the liquid on each tray is perfectly mixed,
- (3) the vapor hold-up on the trays is negligible,
- (4) the transportation delay of liquid and vapor between trays is negligible, and
- (5) the temperature on each tray is uniform.

These lead to the following equations, beginning with the material balances for species j on Tray i

$$\frac{d}{dt} (M_i x_{ij}) = (1 - s_{i+1}^L) L_{i+1} x_{i+1,j} + (1 - s_{i-1}^V) V_{i-1} y_{i-1,j} \quad (A1)$$

$$+ F_i z_{ij} - L_i x_{ij} - V_i y_{ij} \quad \begin{matrix} i=1, \dots, N \\ j=1, \dots, C \end{matrix}$$

The overall mass balance on Tray i is:

$$\frac{dM_i}{dt} = (1 - s_{i+1}^L) L_{i+1} + (1 - s_{i-1}^V) V_{i-1} + F_i - L_i - V_i \quad (A2)$$

$$i=1, \dots, N$$

The equations that relate the compositions of vapor and liquid phases at equilibrium are:

$$y_{ij} = K_{ij} \{x_{ij}, y_i, T_i, P_i\} x_{ij} \quad \begin{matrix} i=1, \dots, N \\ j=1, \dots, C \end{matrix} \quad (A3)$$

The energy balance on Tray i is:

$$\frac{d}{dt} (M_i h_i^L \{x_i, T_i, P_i\}) = (1 - s_{i+1}^L) L_{i+1} h_{i+1}^L \{x_{i+1}, T_{i+1}, P_{i+1}\}$$

$$+ (1 - s_{i-1}^V) V_{i-1} h_{i-1}^V \{y_{i-1}, T_{i-1}, P_{i-1}\}$$

$$+ F_i h_i^F \{z_i, T_i^F, P_i^F\} + Q_i \quad (A4)$$

$$- L_i h_i^L \{x_i, T_i, P_i\} - V_i h_i^V \{y_i, T_i, P_i\}$$

$$i=1, \dots, N$$

The liquid hold-up on Tray i is:

$$M_i = \rho_i^L \{x_i, T_i, P_i\} A_i (H_{w_i} + \Delta_{w_i}) \quad i=1, \dots, N \quad (A5)$$

where A_i is cross-sectional area of Tray i , H_{w_i} is the weir height, and Δ_{w_i} is the crest height of the liquid over the weir. Ballard and Brosilow (1978) use a form of the Francis weir formula to represent the tray hydraulics:

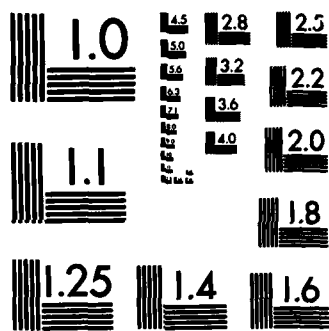
$$\Delta_{w_i} = 1.41 \left[\frac{L_i}{\sqrt{g} \rho_i^L \{x_i, T_i, P_i\} L_{w_i}} \right]^{2/3} \quad i=1, \dots, N \quad (A6)$$

where L_{w_i} is the weir length and g the acceleration due to gravity. Whereas, others assume that Δ_{w_i} is independent of L_i , ρ_i^L , and L_{w_i} ; that is, a constant volume of hold-up exists on each tray.

In addition, the mole fractions for the vapor and liquid phases on Tray i sum to unity:

$$\sum_{j=1}^C x_{ij} = \sum_{j=1}^C y_{ij} = 1 \quad i=1, \dots, N \quad (A7)$$

There are $N(C + 2)$ ODEs (Eqns. (A1), (A2) and (A4), and $N(C + 4)$ algebraic equations ((A3), (A5) - (A7)); hence, $N(C + 5)$ independent equations, since the overall mass balances depend upon the remaining equations. There are $N(3C + 15)$ variables and $N(C + 10)$ specifications.



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

with an appropriate set: $F_1, z_1, T_1^F, P_1^F, P_1^V, s_1^V, A_1, H_{w1}, L_{w1}$, $i=1, \dots, N$, and $j=1, \dots, C$, plus Q_i , $i=2, \dots, N-1$, s_i^L , $i=1, \dots, N-1$, and three of the variables, reflux ratio ($R = (1 - s_N^L)/s_N^L$), boil-up ratio ($R' = V_1/L_1$), bottoms flow rate (L_1), boil-up rate (V_1), reboiler heat duty (Q_1), or condenser heat duty (Q_N). Alternatively, tray pressures, P_i , can be computed as a function of the liquid head on the trays. Note that when constant volume hold-up is assumed, Eqn. (A6) and the variables L_{wi} do not apply; Δ_{wi} is specified in place of L_{wi} .

Eqn. (21) is derived by expanding the left-hand-side of Eqn. (A1). Eqn. (A2) is multiplied by x_{ij} and subtracted from Eqn. (A1). Finally, the vapor mole fractions, y_{ij} , are eliminated by substitution of Eqn. (A3).

Ballard and Brosilow (1978) combine Eqns. (A1), (A2), and (A6) to give:

$$\underline{M}_L \frac{dL}{dt} = \underline{D}L + \underline{E}V - \underline{d}_M \quad (A8)$$

where $\underline{M}_L = \text{diag}\{\partial M_i / \partial L_i\}$ and \underline{D} and \underline{E} are bidiagonal matrices. It is assumed that the liquid density is the molal average of the densities of the pure species. Another equation is derived using Eqns. (A4) and (A3) to give:

$$\underline{B}L + \underline{C}V = \underline{d}_E \quad (A9)$$

where \underline{B} and \underline{C} are bidiagonal matrices. Then, combining Eqns. (A8) and (A9), they obtain:

$$\frac{dL}{dt} = M_L^{-1} (D - EC^{-1}B)L + M_L^{-1} (EC^{-1}d_E - d_M) \quad (A10)$$

Eqn. (38) is derived by expanding the left-hand-side of Eqn. (A4).

Eqn. (A2) is multiplied by h_i^L and subtracted from Eqn. (A4).

ACKNOWLEDGMENT

The authors acknowledge the assistance of Joseph W. Kovach, III and the helpful comments of Stuart W. Churchill. Computing time was provided by the School of Engineering and Applied Science at the University of Pennsylvania and is appreciated.

NOMENCLATURE

| | |
|---|--|
| A, B, C | defined in Eqn. (20) |
| A, C, F, H _G H _T , H _W , K | dimensionless constants in Eqns. (14) and (15) |
| a _{sj} | surface area of particle in size range j, cm ² |
| A _i | active cross-sectional area of Tray i, m ² |
| c | concentration, mol/l; heat capacity of solid particles, cal/g°K; constant in Eqn. (9) |
| C | number of chemical species; heat capacity of gas, cal/g°K |
| d _j | jth eigenvector of Jacobian matrix |
| <u>d_{xj}</u> , <u>d_M</u> , <u>d_E</u> | vectors in Eqns. (21), (A8), and (A9) |
| E _i | activation energy for reaction i, cal/mol |
| f | function |
| F _i | flow rate of feed to Tray i, mol/s |
| g | acceleration due to gravity, m/s ² |
| <u>G_j</u> | triangular coefficient matrix for species j in Eqn. (21) |
| h | step-size; enthalpy, cal/mol; convective heat transfer coef., cal/s cm ² °K |
| h _ε | step-size to satisfy the local truncation error, ε |
| h _g | step-size to satisfy the stability bound for a reference integrator that is not A-stable |
| H _{wi} | height of weir on Tray i, m |
| <u>J</u> | Jacobian matrix |
| k | reaction rate constant |
| k ₀ | pre-exponential factor |
| K | chemical equilibrium constant; number of discrete particle size ranges |
| K _{ij} | vapor-liquid equilibrium constant for species j on Tray i |
| L _i | liquid flow rate from Tray i, mol/s |
| L _{wi} | length of weir on Tray i, mol/s |
| m | number of ODEs |
| m _j | mass flow rate of particles in size range j, g/s |
| M _i | liquid hold-up on Tray i, mol |

| | |
|-------------------|--|
| \underline{M}_L | $\text{diag}\{\partial M_i/\partial L_i\}$ |
| n | number of equally spaced intervals in x (Eqn. (17)); number of fast variables |
| N | number of trays; number of gas-solid reactions |
| NA | $R - NB$ |
| NB | number of independent reactions taken at equilibrium |
| p | Partial pressure of reacting species |
| Q_i | rate of heat transfer to Tray i , cal/s |
| r | order of accuracy; intrinsic rate of reaction, mol/l·s |
| r_n | net rate of reaction ($r_{\text{for}} - r_{\text{rev}}$), mol/l·s |
| R | universal gas constant; number of independent chemical reactions |
| s | $(t_{\text{final}} - t_0)/\tau_{\text{min}}$ |
| s_i | fraction of stream in sidedraw from Tray i |
| SR | stiffness ratio, $ \text{Re}\{\lambda\} _{\text{max}}/ \text{Re}\{\lambda\} _{\text{min}}$ |
| t | time, s |
| T | temperature, °K |
| u | dimensionless velocity in x -direction |
| v | volatiles produced in size range j by reaction i , g/g |
| v_i^* | max. volatiles produced by reaction i , g/g of solid initially |
| V_i | flow rate of vapor stream from Tray i |
| x | spatial dimension |
| x_{ij} | mole fraction of species j in liquid on Tray i |
| y | dependent variable |
| z_{ij} | mole fraction of species j in feed stream to Tray i |

Greek Symbols

| | |
|--------------------------|--|
| α_{jr} | relative volatility, K_j/K_r |
| $\alpha_{\ell, \beta-1}$ | parameters in GEAR backward difference formulas |
| γ_{∞} | characteristic root of integration formula as $h \text{Re}\{\lambda\} \rightarrow \infty$ |
| δ | real stability bound |
| Δx | grid spacing in x -direction |

| | |
|-----------------------|--|
| Δw_i | liquid height over weir on Tray i , m |
| ζ | linear combination of concentrations (Eqn. (26)), mol/l |
| $\underline{\lambda}$ | vector of m eigenvalues |
| $\bar{\lambda}_s$ | pseudo-eigenvalue for stiff variable, Eqn. (42) |
| ρ_i^L | density of liquid on Tray i |
| σ | h_e/h_s |
| τ | time constant, $1/ Re\{\lambda\} $; dimensionless time, Eqn. (13) |
| μ | dimensionless viscosity |
| ϕ | defined in Eqn. (20) |
| ω | frequency in Eqn. (9) |

Subscripts

| | |
|-----|--|
| e | entrance |
| F | feed |
| for | forward |
| i | time-step counter; tray counter |
| j | species counter; particle size-range counter |
| n | time-step counter |
| o | initial |
| p | particle |
| rev | reverse |
| s | stiff (or fast) variable |
| w | wall |

Superscripts

| | |
|---|---------|
| F | feed |
| L | liquid |
| o | initial |
| v | vapor |

LITERATURE CITED

Aiken, R.C. and L. Lapidus, "An Effective Numerical Integration Method for Typical Stiff Systems", AICHE Journal, 20, 2 (1974).

Ballard, D.M. and C.B. Brosilow, "Dynamic Simulation of Multicomponent Distillation Columns", Paper 42a presented at the 71st Annual Meeting of AIChE, Miami, November, 1978.

Blakemore, J.E. and W.H. Corcoran, "Validity of the Steady-state Approximation Applied to the Pyrolysis of n-Butane", IEC Proc. Des. Dev., 8, 2 (1969).

Boston, J.F., Britt, H.I., Jirapongphan, S., and V.B. Shah, "An Advanced System for the Simulation of Batch Distillation Operations", Foundations of Computer-Aided Chemical Process Design, Vol. II, edited by R.S.H. Mah and W.D. Seider, AIChE (1981).

Boston, J.F. and S.L. Sullivan, Jr., "A New Class of Solution Methods for Multicomponent, Multistage, Separation Processes", Can. Jour. of Chem. Eng., 52, 1 (1974).

Byrne, G.D., "Some Software for Solving Ordinary Differential Equations", Foundations of Computer-Aided Chemical Process Design, Vol. I, edited by R.S.H. Mah and W.D. Seider, AIChE (1981).

Carver, M.B., "The Choice of Algorithms in Automated Method of Lines Solution of Partial Differential Equations", Numerical Methods for Differential Systems, edited by L. Lapidus and W.E. Schiesser, Academic Press, 1976.

Curtiss, C.F. and J.O. Hirschfelder, "Integration of Stiff Equations", Proc. Nat. Acad. Sci. USA, 38 (1952).

Dickinson, Jr., R.P., and R.J. Gelinas, "SETKIN: A Chemical Kinetics Preprocessor Code", Numerical Methods for Differential Systems, edited by L. Lapidus and W.E. Schiesser, Academic Press, 1976.

Edelson, D. and D.L. Allara, "Parametization of Complex Reaction Systems: Model Fitting vs. Fundamental Kinetics", AICHE Journal, 19, 3 (1973).

Edsburg, L., "Numerical Methods for Mass Action Kinetics", Numerical Methods for Differential Systems, edited by L. Lapidus and W.E. Schiesser, Academic Press, 1976.

Field, R.J. and R.M. Noyes, "Oscillations in Chemical Systems. IV. Limit Cycle Behavior in a Model of a Real Chemical Reaction", J. Chem. Phys., 60 (1974).

Gear, C.W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, New Jersey, 1971.

Herriot, G.E., Eckert, R.E., and L.F. Albright, "Kinetics of Propane Pyrolysis", AICHE Journal, 18, 84 (1972).

Hindmarsh, A.C., "GEAR: Ordinary Differential Equation System Solver", Computer Documentation, UCID-30001, Rev. 3, Lawrence Livermore Laboratory, Dec., 1974.

Hindmarsh, A.C., "LSODE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers", ACM-SIGNUM Newsletter, 15, 4, 10 (1980).

Hu, S.S. and W.E. Schiesser, "An Adaptive Grid Method in the Numerical Method of Lines", Adv. in Comp. Meth. for Part. Diff. Eqns., IV, edited by R. Vichnevetsky and R.S. Stepleman, IMACS, 1981.

IMSL Library, 6th Ed., Int'l. Math. and Stat. Lib., Inc., July, 1977.

Kayihan, F., "An Iterative Approach for the Solution of a Class of Stiff ODE Models of Reacting Polydispersed Particles", ACS Symp. Ser., No. 124, 1980.

Luss, D. and N.R. Amundson, "Stability of Batch Catalytic Fluidized Beds", AIChE Journal, 14, 2 (1968).

Mah, R.S.H., Michaelson, S., and R.W.H. Sargent, "Dynamic Behavior of Multi-component Multi-stage Systems. Numerical Methods for the Solution", Chem. Eng. Sci., 17, 619 (1962).

Prokopakis, G.J., Ross, B.A., and W.D. Seider, "Azeotropic Distillation Towers with Two Liquid Phases", Foundations of Computer-aided Chemical Process Design, Vol. II, edited by R.S.H. Mah and W.D. Seider, AIChE (1981).

Prokopakis, G.J. and W.D. Seider, "Adaptive Semi-implicit Runge-Kutta Method for Solution of Stiff Ordinary Differential Equations", IEC Fund., 20, 255 (1981).

Prokopakis, G.J. and W.D. Seider, "Dynamic Simulation of Azeotropic Distillation Towers", submitted to AIChE Journal, 1982.

Shampine, L.F. and C.W. Gear, "A User's View of Solving Stiff Ordinary Differential Equations", SIAM Rev., 21, 1 (1979).

Sorensen, J.P., and W.E. Stewart, "Structural Analysis of Multicomponent Reaction Models: Part II, Formulation of Mass Balances and Thermodynamic Constraints", AIChE Journal, 26, 98 (1980).

Sundaram, K.M. and G.F. Froment, "Modeling of Thermal Cracking Kinetics. 3. Radical Mechanisms for the Pyrolysis of Simple Paraffins, Olefins, and Their Mixtures", IEC Fund., 17, 3 (1978).

Suuberg, E.M., ScD Thesis, Dept. of Chem. Eng., MIT, Cambridge, MA, 1977.

Tyreas, B.D., Luyben, W.L., and W.E. Schiesser, "Stiffness in Distillation Models and the Use of an Implicit Integration Method to Reduce Computation Times", IEC Proc. Des. Dev., 14, 4 (1975).

Wei, J. and C.D. Prater, "The Structure and Analysis of Complex Reaction Systems", Adv. in Catal., 13, Academic Press, NY, 1962.

White, C.W., III and W.D. Seider, "Analysis of Chemical Reaction Systems", Chem. Eng. Comm., 9, 159 (1981).

Table 1 Concentration derivatives and eigenvalues for the Belousov reaction system

| t, sec | dy_1/dt | dy_2/dt | dy_3/dt | λ_1 | λ_2 | λ_3 | Stiffness ratio |
|--------|------------------------|------------------------|-----------------------|-----------------------|-------------|-------------|----------------------|
| 0.1969 | 17.7 ³ | -0.100 | 0.935 | -84.082 | -0.0933 + | 0.11721 | 901 |
| 1.057 | 7.75x10 ⁵ | -3.19 | 55.5 ³ | -60.565 | -0.1020 + | 0.96841 | 594 |
| 1.137 | 5.53x10 ⁵ | -10.5 | 1.28x10 ³ | -7.949 | -112.65 | -1.6311 | 69 |
| 1.145 | 1.00x10 ⁶ | -6.01 | 2.17x10 ³ | -16.339 | -178.20 | -0.9162 | 195 |
| 1.152 | 1.53x10 ⁶ | -2.43 | 3.23x10 ³ | -25.381 | -261.43 | -0.6524 | 401 |
| 1.164 | 3.11x10 ⁶ | -4.84x10 ⁻² | 6.48x10 ³ | -51.985 | -523.04 | -0.4010 | 1304 |
| 1.168 | 3.85x10 ⁶ | 1.51x10 ⁻² | 8.02x10 ³ | -64.401 | -646.66 | -0.3546 | 1824 |
| 1.243 | 8.99x10 ⁶ | 0.368 | 1.87x10 ⁴ | -152.20 | -1524.8 | -0.2428 | 6280 |
| 2.990 | 4.43x10 ⁶ | -17.5 | 9.24x10 ³ | -104.70 | -1100.5 | -0.2772 | 3970 |
| 3.864 | 2.40x10 ⁵ | 11.8 | 3.53x10 ³ | -36.480 | -509.61 | -0.4521 | 1127 |
| 3.916 | -2.09x10 ⁴ | 3.99x10 ² | -5.01x10 ³ | -612.06 | -0.0934 + | 0.07591 | 6553 ⁴ |
| 3.921 | -1.15x10 ³ | 4.02x10 ² | -5.01x10 ³ | -771.13 ³ | -0.0287 | -0.1582 | 2.69x10 ⁵ |
| 3.994 | -1.39x10 ³ | 3.97x10 ² | -4.95x10 ² | -3.02x10 ⁵ | -0.0259 | -0.1610 | 1.16x10 ⁷ |
| 14.24 | -1.92x10 ² | 31.8 | -9.51x10 ² | -1.33x10 ⁴ | -0.0259 | -0.1610 | 5.14x10 ⁶ |
| 45.92 | -4.21x10 ² | -12.5 | -5.80 | -7.76x10 ⁴ | -0.0259 | -0.1610 | 3x10 ⁶ |
| 64.41 | -2.52x10 ² | -16.1 | -0.296 ⁻⁵ | -4.82x10 ⁴ | -0.0259 | -0.1610 | 1.86x10 ⁵ |
| 115.7 | -18.2 ⁻² | -4.29 | 6.44x10 ⁻³ | -1.28x10 ⁴ | -0.0259 | -0.1610 | 4.94x10 ⁴ |
| 219.6 | -2.21x10 ⁻² | -0.291 | 2.30x10 ⁻² | -868.36 | -0.0260 | -0.1610 | 3.34x10 ⁴ |
| 286.1 | 5.37x10 ⁻² | -5.40x10 ⁻² | 3.80x10 ⁻² | -152.62 | -0.0344 | -0.1525 | 4437 |
| 292.8 | 0.120 | -4.79x10 ⁻² | 7.08x10 ⁻² | -126.25 | -0.0429 | -0.1440 | 2943 |
| 298.9 | 0.431 | -4.86x10 ⁻² | 0.172 | -104.03 | -0.0825 | -0.1043 | 1261 |
| 300.0 | 0.606 | -5.13x10 ⁻² | 0.224 | -99.744 | -0.0934 + | 0.03101 | 1068 |

Table 2 Concentration derivatives and eigenvalues for the atmospheric reaction system. Computed using LSODE*.

| <u>t, hr</u> | <u>dy₁/dt</u> | <u>dy₂/dt</u> | <u>λ₁</u> | <u>λ₂</u> |
|--------------|--------------------------|--------------------------|----------------------|--------------------------|
| 0 | -6.0315x10 ⁶ | 6.0305x10 ⁶ | -6.031 | -9.319x10 ⁻¹⁰ |
| 1 | 2.324x10 ⁻⁴ | -2.595x10 ⁻⁴ | -6.031 | =0 |
| 2 | 79.331 | -117.94 | -6.031 | -7.723x10 ⁻¹¹ |
| 4 | 9.495x10 ³ | 3.022x10 ⁵ | -6.032 | -4.773x10 ⁻⁸ |
| 6 | 9.178x10 ² | 1.102x10 ⁷ | -6.032 | -1.622x10 ⁻⁷ |
| 8 | -1.021x10 ⁴ | 3.180x10 ⁵ | -6.032 | -5.140x10 ⁻⁸ |
| 10 | -85.518 | 40.671 | -6.032 | -8.324x10 ⁻¹¹ |
| 12 | =0 | =0 | -6.032 | =0 |
| 18 | =0 | =0 | -6.032 | =0 |
| 24 | =0 | =0 | -6.032 | =0 |
| 30 | 9.115x10 ² | 1.101x10 ⁷ | -6.032 | -1.741x10 ⁻⁷ |
| 42 | =0 | =0 | -6.032 | =0 |

* Relative error tolerance = 10⁻⁸
 Absolute error tolerance = 10⁻⁴⁵
 UNIVAC 1100 - Double precision
 Values =0 are <10⁻¹⁷

Table 3 Eigenvalues of ODEs from discretization of the spatial derivative in the diffusion equation.

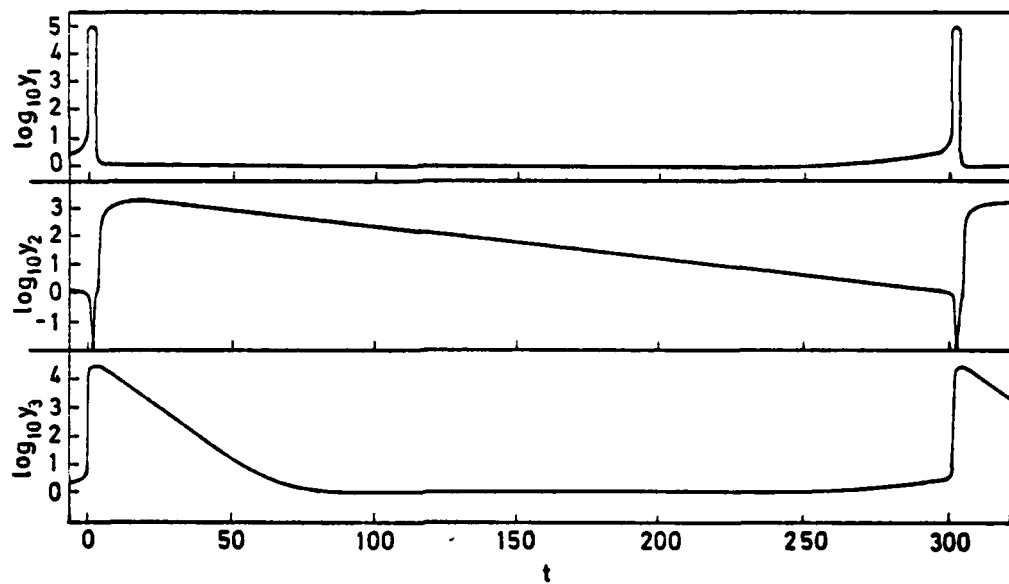
| <u>n</u> | <u>λ min</u> | <u>λ max</u> | <u>Stiffness ratio</u> |
|----------|-----------------------------------|-----------------------------------|------------------------|
| 2 | 9.00 | 27.00 | 3 |
| 3 | 9.37 | 54.62 | 5.8 |
| 5 | 9.65 | 134.34 | 13.9 |
| 7 | 9.74 | 246.23 | 25.3 |
| 9 | 9.79 | 390.18 | 39.9 |
| 11 | 9.81 | 566.14 | 57.7 |
| 13 | 9.83 | 774.11 | 78.8 |
| 15 | 9.84 | 1013.8 | 103 |

Table 4 Eigenvalues for momentum balance with viscous dissipation⁺

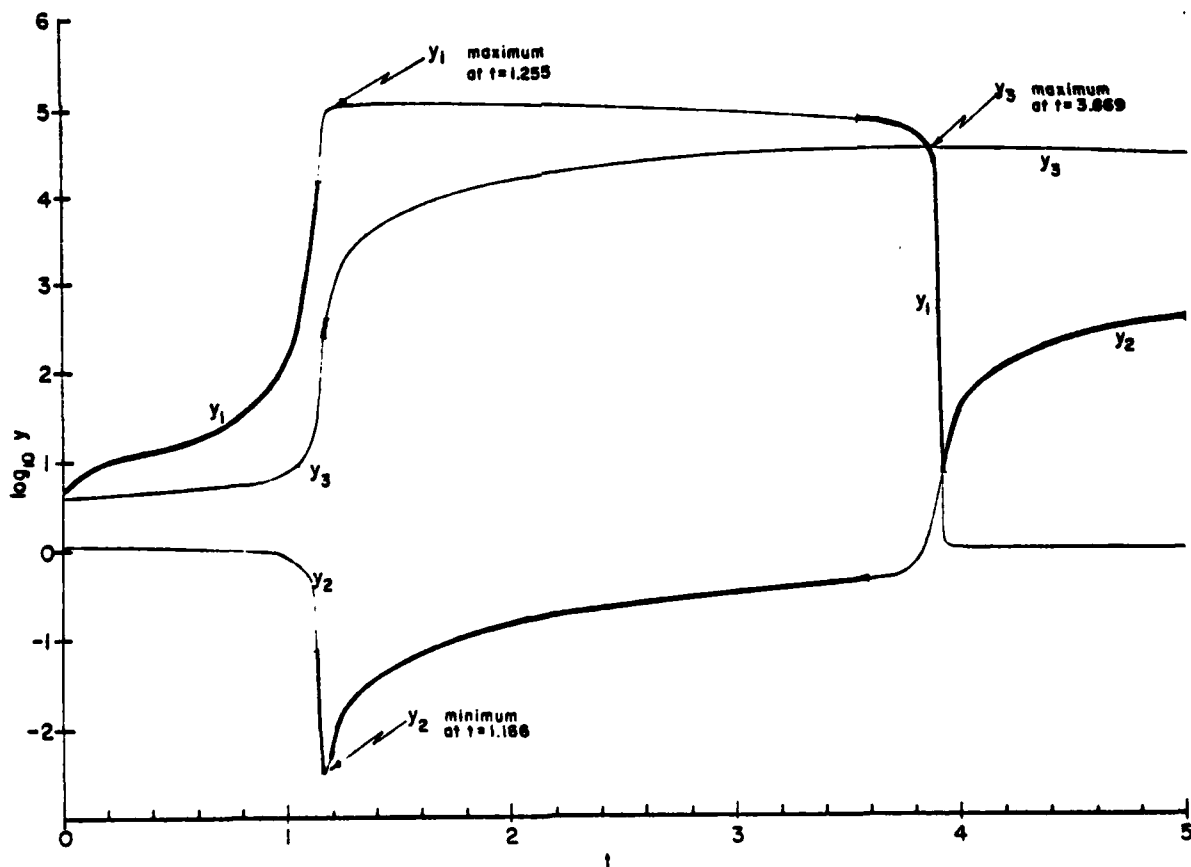
| <u>Dimensionless viscosity, μ</u> | <u>t</u> | <u>$\text{Re}\{\lambda\}$ min</u> | <u>$\text{Re}\{\lambda\}$ max</u> |
|--|----------|--|--|
| 0.003* | 0 | 0.937 | 224 |
| | 0.2 | 0.964 | 238 |
| | 0.5 | 1.04 | 254 |
| | 1.0 | 2.65 | 263 |
| 0.03 | 0 | 0.721 | 1.69×10^3 |
| | 0.2 | 0.900 | 1.71×10^3 |
| | 0.5 | 1.44 | 1.72×10^3 |
| | 1.0 | 5.39 | 1.72×10^3 |
| 0.3 | 0 | 3.07 | 1.61×10^4 |
| | 0.2 | 3.08 | 1.61×10^4 |
| | 0.5 | 3.12 | 1.61×10^4 |
| | 1.0 | 3.18 | 1.61×10^4 |

* Most eigenvalues are complex

+ All eigenvalues have a negative real part



(a)



(b)

Figure 1. Profiles of dimensionless concentration for the Belousov reaction system

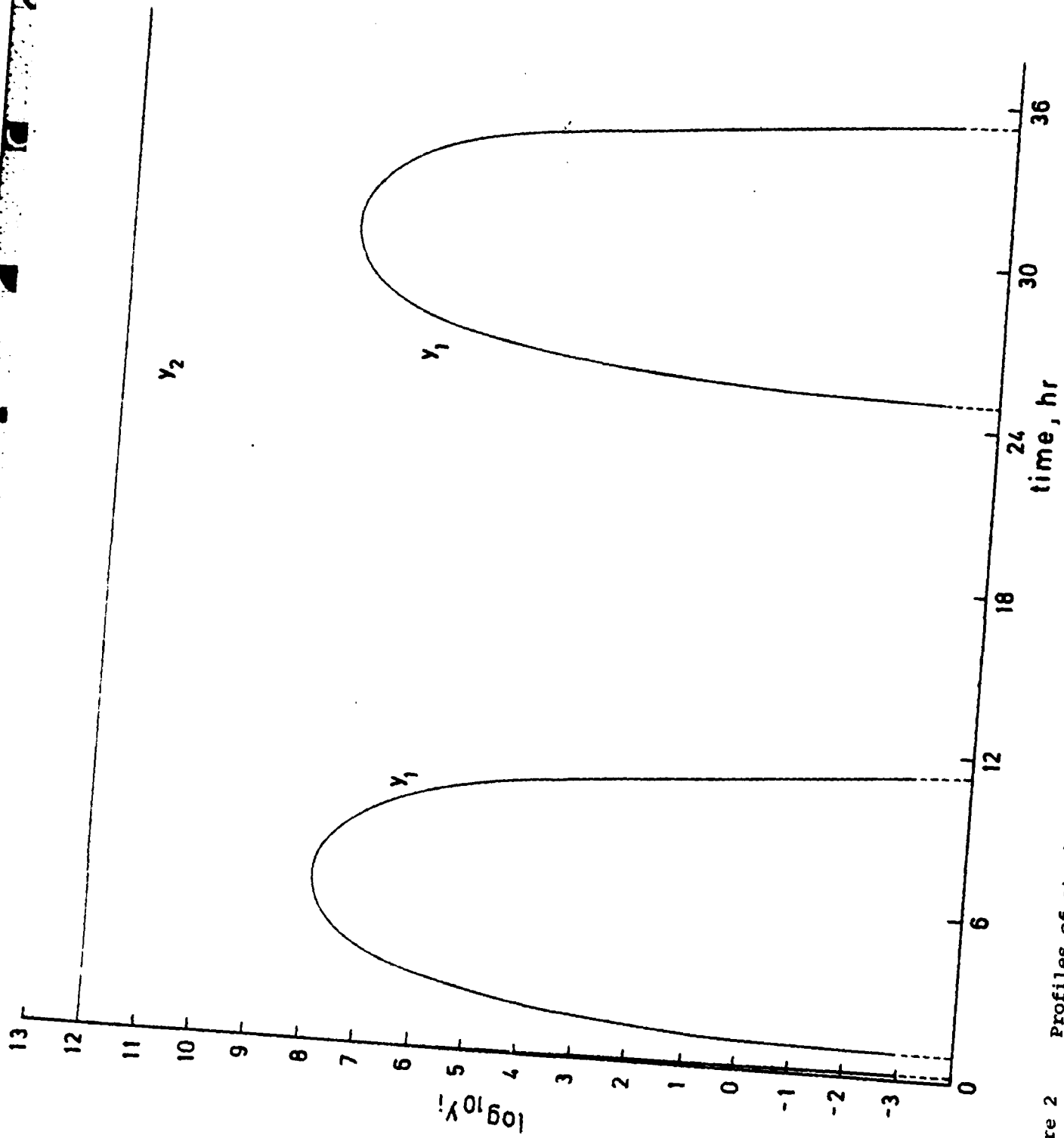


Figure 2 Profiles of atomic oxygen and ozone concentration for atmospheric reaction system (computed using LSODE (Hindmarsh, 1980)).

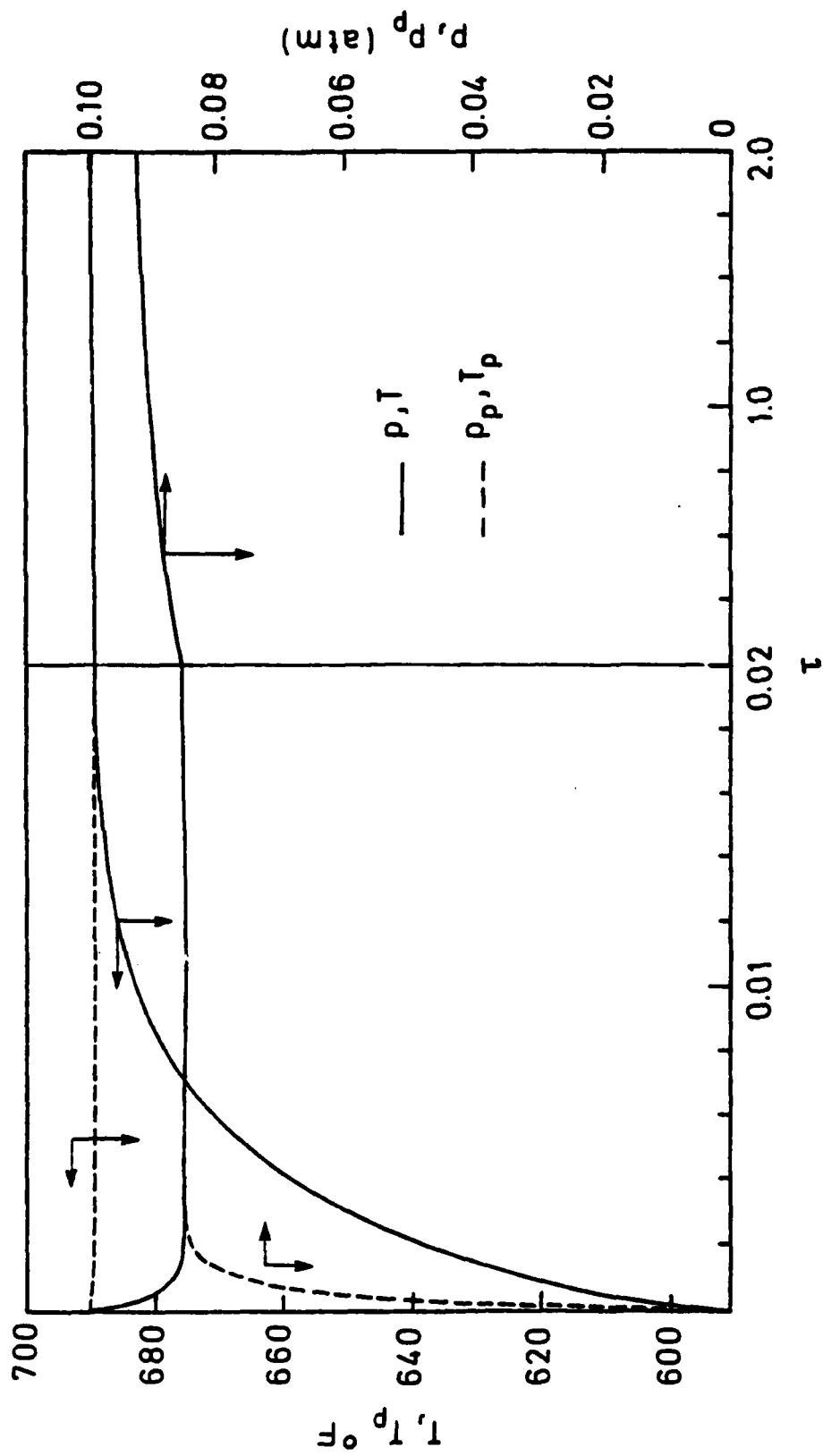


Figure 3 Profiles of bulk and particle temperature and partial pressure for the batch fluidized-bed reactor. From Luss and Amundson (1968).

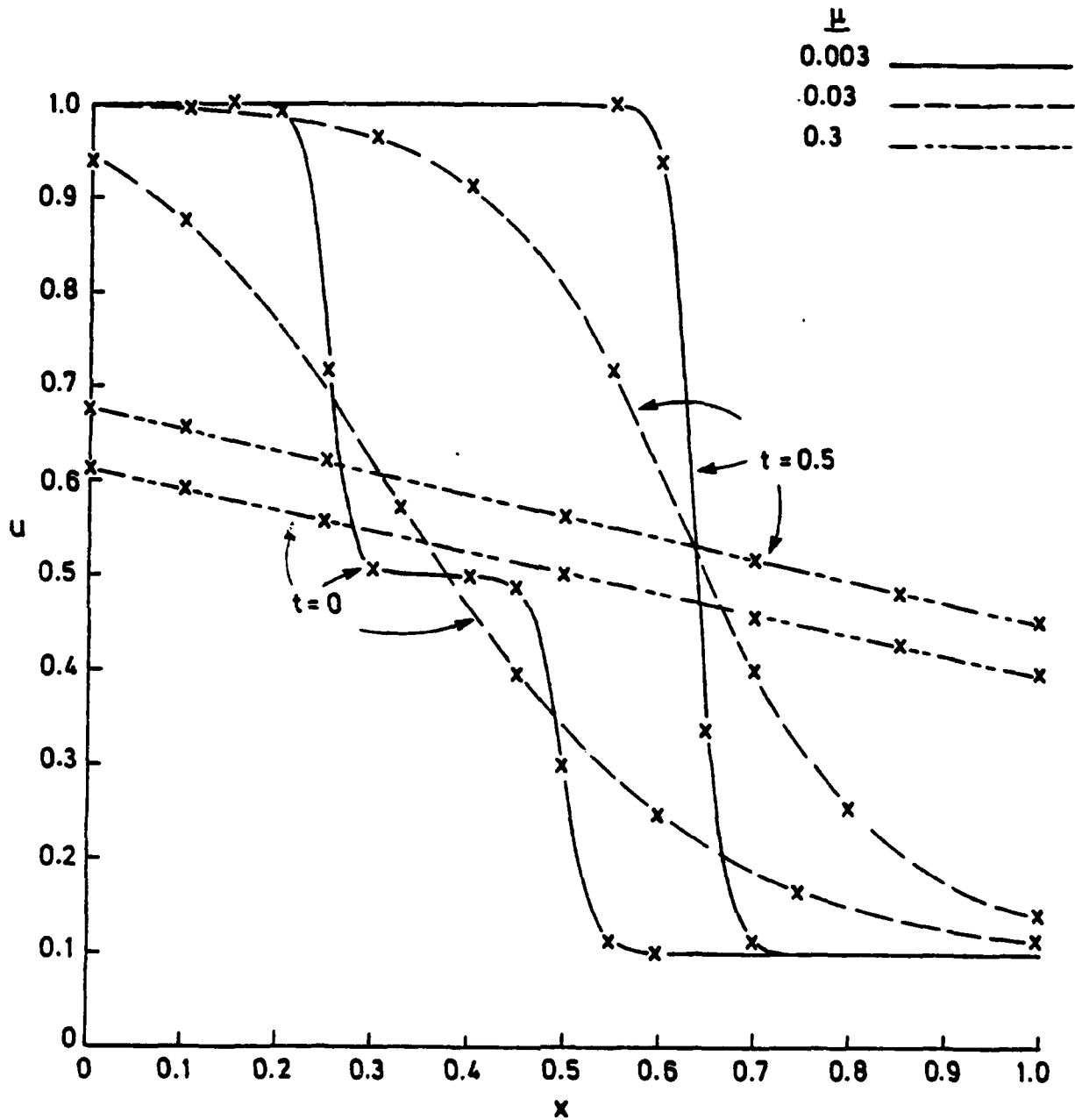


Figure 4 Velocity profiles with viscous dissipation in the x direction. Values computed using the method of lines are x . Curves are the analytical solution.

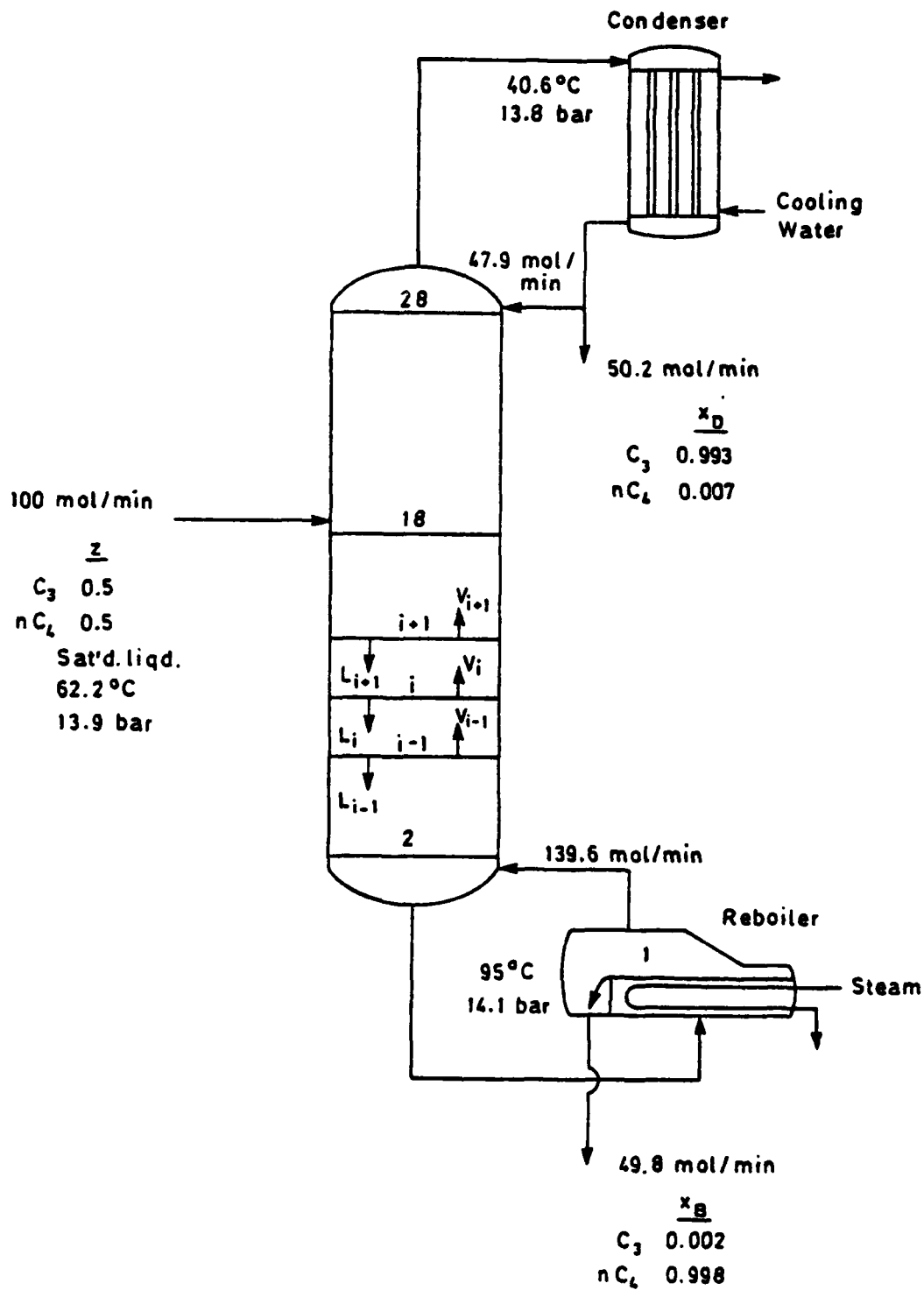


Figure 5 Distillation tower for separation of propane and n-butane.

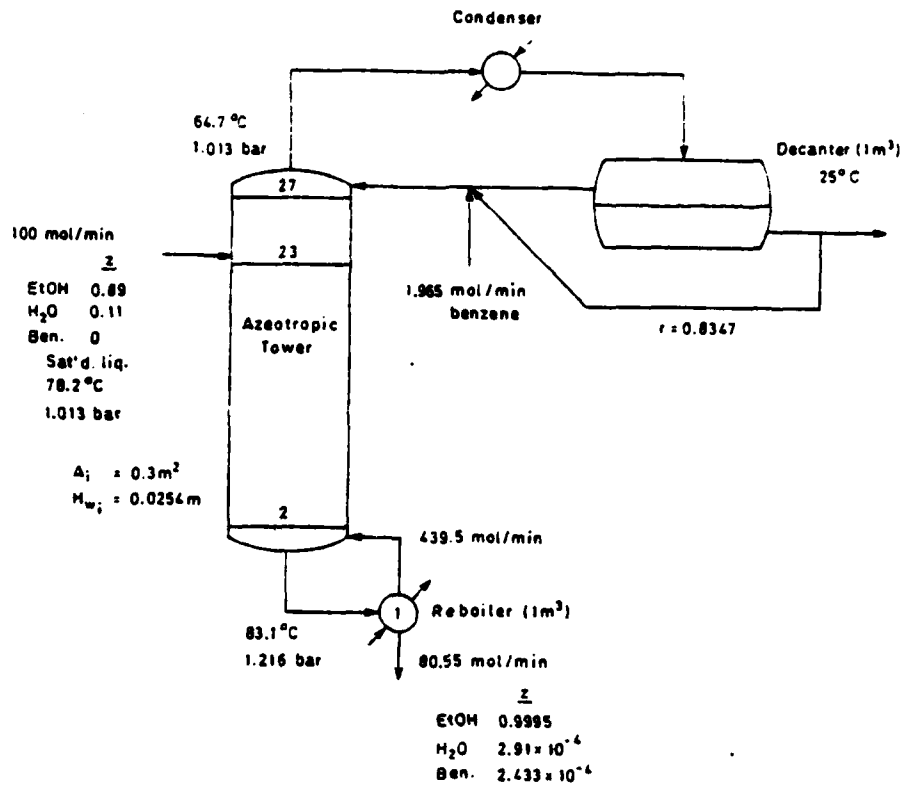


Figure 6 Azeotropic distillation tower for dehydration of ethanol with benzene.

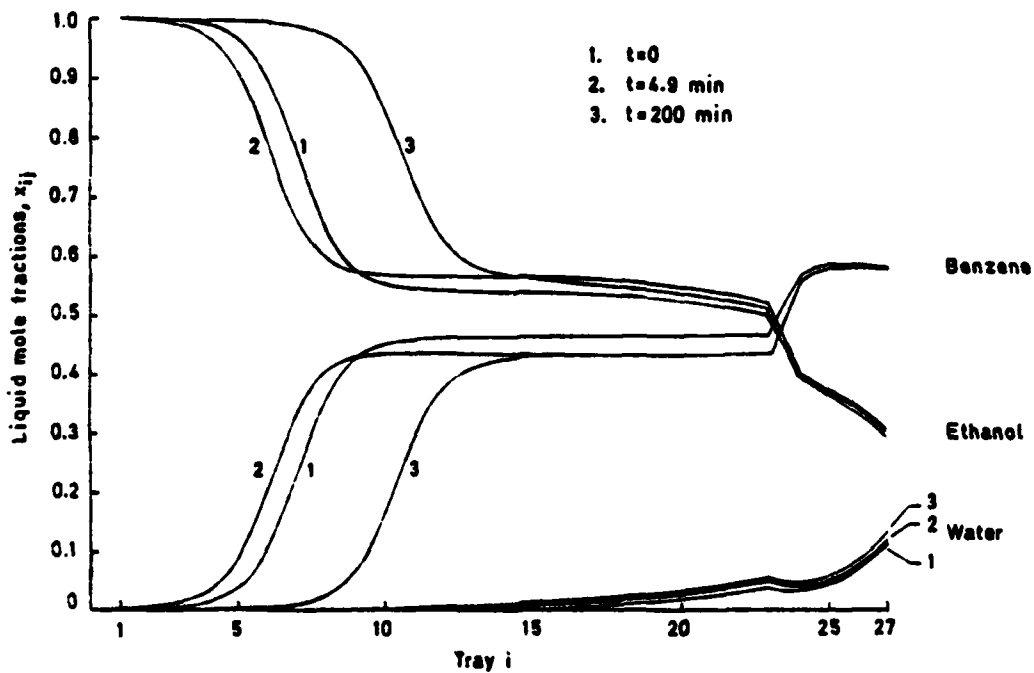


Figure 7 Profiles of liquid mole fractions after a 30 percent increase in feed flow rate.

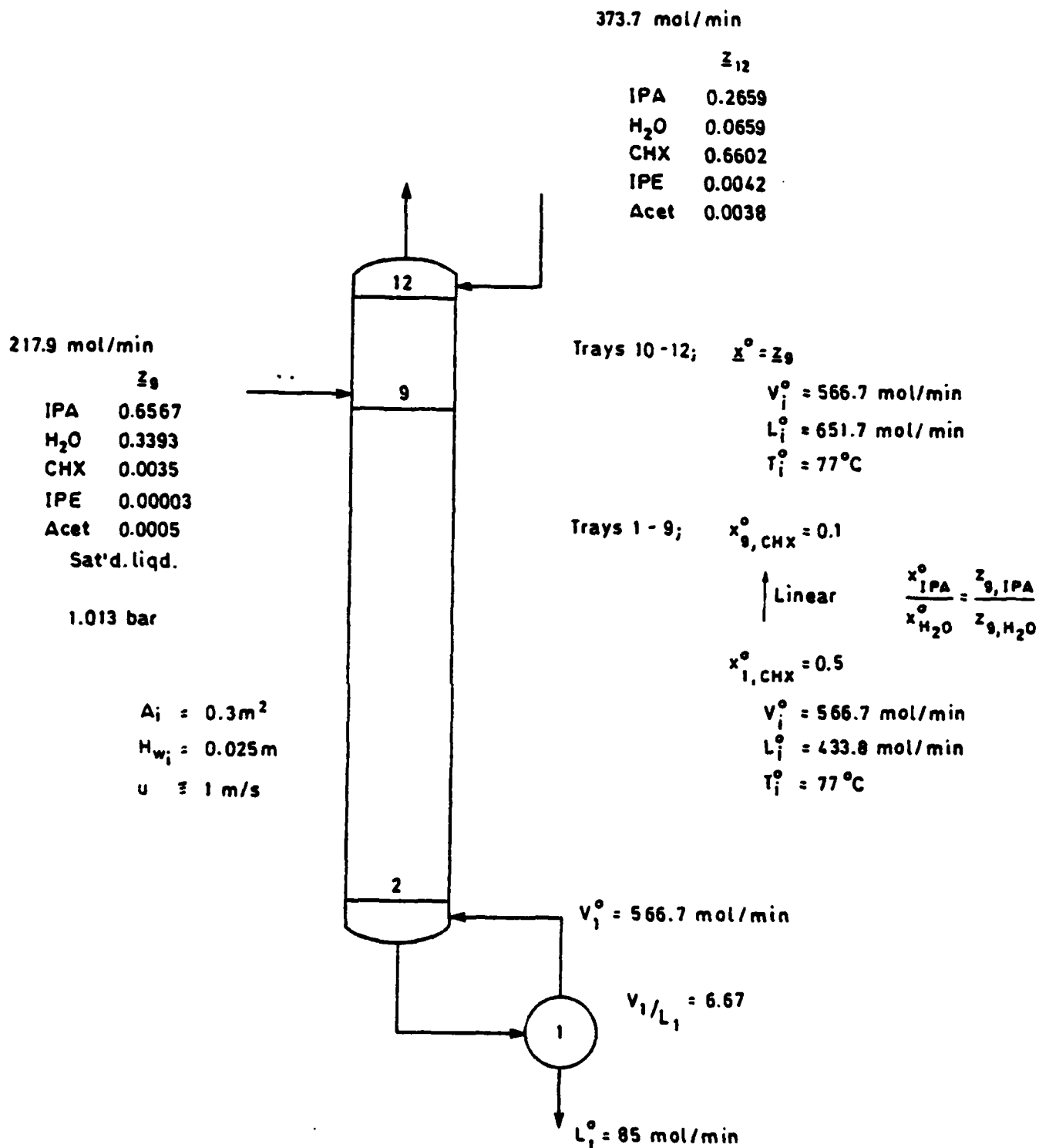


Figure 8 Azeotropic distillation tower for dehydration of isopropanol with cyclohexane.

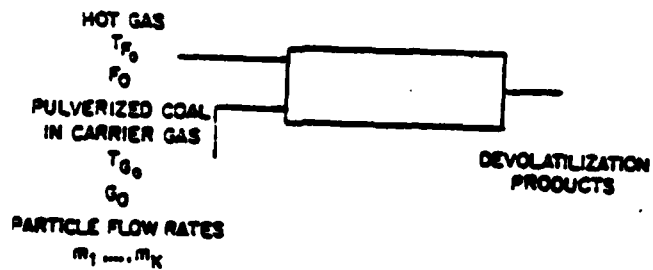


Figure 9 Schematic of entrained bed devolatilization reactor.

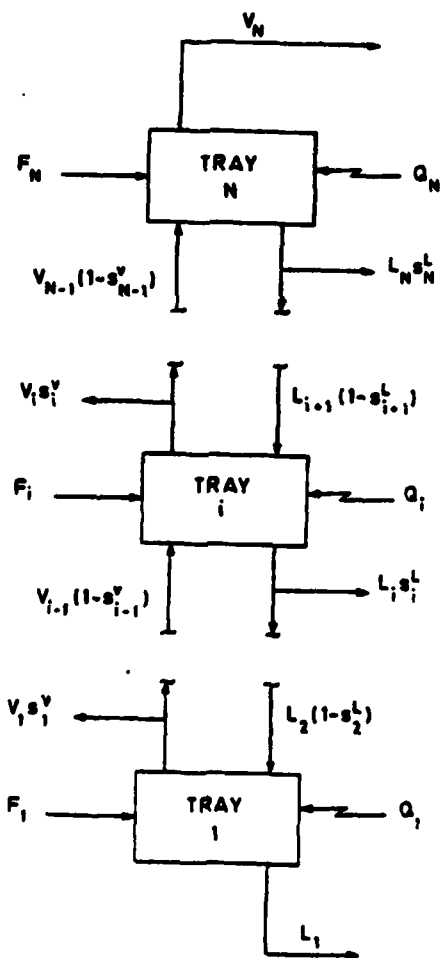


Figure 10 Schematic of a continuous distillation.

University of Leeds

Department of Computer Studies

ABSTRACT FOR THE "INTERNATIONAL CONFERENCE ON STIFF COMPUTATION

Numerical Integration of Stiff Differential/Algebraic Equations
with Severe Discontinuities

T. S. Chua
P. M. Dew
Department of Computer Studies
The University
LEEDS LS2 9JT

The simulation of industrial processes often involves the numerical solution of a system of initial valued, stiff differential/algebraic equations (DAE) with severe discontinuities in the derivative of its solution. On problems of this type a standard variable-step integrator is either very inefficient or fails to compute the solution.

The purpose of this paper is to describe a restart strategy which can be used to improve the efficiency of a variable-step stiff integrator and present a case for its inclusion in a general-purpose stiff solver. The paper will also discuss the design of an efficient variable-step solver for a DAE system based on the work of Petzold[1]. The design and use of the restart strategy for a DAE system arising from the simulation of British Gas transmission networks is given in Chua and Dew[2]; this paper put this work into a more general framework.

The general parabolic DAE system considered is

$$E \frac{dy}{dt} = \underline{f}(t, y) \quad (1)$$

where the matrix E is singular if there are algebraic equations present in the system. The general strategy employed is to use normal variable-step integration everywhere except immediately after a discontinuity has occurred when a restart phase is indicated. (The discontinuity is detected using a supplied function which changes sign over the discontinuity). During the restart phase, the local error control is suspended which enables a larger timestep to be used than would otherwise be the case. The timestep is kept constant during this phase until the estimate of the global error indicates that it is safe to return to normal variable-step integration. Extensive numerical testing on Gas transmission problems has shown that the restart strategy is extremely robust and considerably reduces the amount of computation.

The variable-step integrator described in the paper is based on a 2-stage 2nd order, L-stable Rosenbrock-type method (Scraton[3]). The numerical performances of the integrator is illustrated using a heat conduction equation with discontinuous boundary conditions and a large scale British Gas transmission network simulation.

References

- Petzold L. "Differential/algebraic equations are not ODE's" Sandio Laboratories, Numerical Maths div 5492, Albuquerque, New Mexico 1980.
- Chua T. S. and Dew P. M. "The simulation of a gas transmission network using a variable-step integrator" in the proceeding of International Conference on Numerical Methods for Coupled Problems held at University College of Swansea, Sept. 1981. Editors Hinton, Bettess and Lewis Pineridge Press.
- Scraton R. E. "Some L-stable methods for stiff differential equations" Intern. J. Computer Maths. Section B 9, 1981, pp 81-87.

EXPLOSION MODE ANALYSIS OF AN H_2-O_2 REACTION

by

F. C. Hoppensteadt and P. Alfeld

Department of Mathematics
University of Utah
Salt Lake City, UT 84112

ABSTRACT

It is shown here that the rate equations for a first order chain branched reaction modelling the combustion of hydrogen can be approximated by a solvable canonical problem. We call this the *explosion mode approximation* to the problem. We say that there is an explosion if branching reactions initially dominate termination reactions. A dimensionless parameter E is introduced which provides a quantitative criterion for an explosion to occur; for $E > 0$ there is an explosion, for $E < 0$ there is not. We construct approximations to the explosion kinetics when $E > 0$, and present numerical simulations to illustrate several cases.

1. A CANONICAL FIRST ORDER CHAIN BRANCHED REACTION

The reaction studied in this section illustrates the basic calculations needed for our explosion mode analysis of the combustion of hydrogen. The reaction is summarized in Table 1.

We denote by $[A]$ the concentration of chemical species A, and we set

$$u = [A] \quad \text{and} \quad x = [B].$$

The kinetic rate equations which describe this reaction are

$$\begin{aligned} \dot{u} &= -kux - k_0u^2, & u(0) &= u_0 \\ \dot{x} &= kux - ax + k_0u^2, & x(0) &= 0, \end{aligned}$$

where $\dot{} = d/dt$. The analysis of this problem breaks down into three parts which reflect the three stages of the reaction.

Initiation: $x \sim 0$.

In the initial stages, the kinetics are described by the equations

$$\dot{u} = -k_0u^2, \quad \dot{x} = k_0u^2$$

Thus, $u \approx u_0/(u_0k_0t+1)$ and $x \approx u_0 - u$.

Branching: $kux - ax \gg k_0u^2$.

If $ku_0 > a$, then $[B]$ will grow due to the branching reaction dominating termination. The model now becomes

$$\dot{u} = -kux, \quad \dot{x} = (ku-a)x.$$

This problem can be solved explicitly:

$$x = \left(\frac{a}{k}\right) \ln(u/\bar{u}) - u + \bar{u} + \bar{x}$$

where \bar{u} , \bar{x} are values of [A] and [B] after the initiation phase of the reaction. In fact, the initiation and branching phases are adequately approximated by taking

$$x \approx \frac{a}{k} \ln(u/u_0) + u_0 - u.$$

This approximation is valid throughout the remainder of the reaction. The initiation reaction plays no further significant role since at the maximum of x , u rapidly approaches a quasi-static state ($\dot{u} = 0$), after which the termination reaction dominates:

$$u \sim 0, \quad \dot{x} = -ax.$$

Parameters analogous to those of hydrogen combustion are

$$u_0 = 1E-8, \quad k_0 = 1E2, \quad k = 1E11, \quad a = 1E2, \quad x_0 = 0.$$

Here the notation 1E-8 denotes 1×10^{-8} , etc.

Reactions of this kind have been studied earlier [1-3]. In the next section, we show how this analysis can be applied to study the combustion of hydrogen.

2. AN H₂-O₂ REACTION

The reaction studied here is described in Table 2. This breaks down into three types of elementary reactions: initiation, branching and

propagation, and termination. In these reactions, HO_2 is taken to be an inactive particle [1,3] and W in the termination reactions indicates collision with the container's wall.

We set

$$u = [\text{H}_2], \quad v = [\text{O}_2], \quad x = [\text{H}\cdot], \quad y = [\text{OH}\cdot], \quad z = [\text{O}\cdot].$$

Then, the kinetic rate equations are

$$\begin{aligned} \dot{u} &= -(k_1 y + k_3 z)u - k_0 uv, & u(0) &= u_0, \\ \dot{v} &= -(k_2 x)v - k_0 uv, & v(0) &= v_0, \\ \dot{x} &= -(k_2 v + a_2)x + (k_1 u)y + (k_3 u)z + k_0 uv, & x(0) &= 0, \\ \dot{y} &= (k_2 v)x - (k_1 u + a_1)y + (k_3 u)z, & y(0) &= 0, \\ \dot{z} &= (k_2 v)x - (k_3 u + a_3)z, & z(0) &= 0, \end{aligned}$$

where $a_i = a_i^! w$, $i = 1, 2, 3$, denote the pseudo first order rate constants for the termination reaction (w remains constant throughout these reactions). It is convenient to rewrite this system in matrix form

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} u \\ v \end{pmatrix} &= \begin{pmatrix} -(k_1 y + k_3 z) & 0 \\ 0 & -k_2 x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} - k_0 uv \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \frac{d}{dt} \begin{pmatrix} x \\ y \\ z \end{pmatrix} &= (B-T) \begin{pmatrix} x \\ y \\ z \end{pmatrix} + k_0 uv \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

The matrix B describes the branching reactions, and it is of the form

$$B = \begin{pmatrix} -\alpha & \beta & \gamma \\ \alpha & -\beta & \gamma \\ \alpha & 0 & -\gamma \end{pmatrix}, \quad \text{where } \alpha = k_2 v, \beta = k_1 u \text{ and } \gamma = k_3 u.$$

The matrix T , which describes the termination reactions, is

$$T = \begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix} .$$

$k_0 = 0$. Ignoring for the moment the initiation reaction, we see that initially (i.e., at $t = 0$ where $u = u_0, v = v_0, x = y = z = 0$), the rate equations have the form

$$\dot{u} = \dot{v} = 0, \quad \frac{d}{dt} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = (B_0 - T) \begin{pmatrix} x \\ y \\ z \end{pmatrix} .$$

where B_0 denotes B evaluated at $u = u_0, v = v_0$. If the matrix $B_0 - T$ has a positive eigenvalue, then once initiated, the branching reactions dominate termination, and the vector of radical components (i.e., x, y, z) has a growing mode. We base our definition of an explosion on this fact. Namely, we say that an explosion will occur if branching initially dominates termination, or equivalently if $B_0 - T$ has a positive eigenvalue. We define the *explosion number* to be the dimensionless quantity

$$E = \det(B_0 - T)\tau^3$$

where τ has the dimensions of time used to measure the kinetic rates. We show next that $E > 0$ implies an explosion occurs, $E < 0$ implies no explosion occurs.

3. EXPLOSION MODE ANALYSIS:

a. Spectral Analysis of $B - T$.

We begin with the characteristic polynomial of $B - T$:

$$P(\lambda) = -\det(B - T - \lambda I_3)$$

whose roots are the eigenvalues of $B-T$. Here I_3 denotes the 3×3 -identity matrix. P has the form

$$P(\lambda) = \lambda^3 + \sigma\lambda^2 + \mu\lambda - \delta(u,v)$$

where

$$\sigma = \alpha + \beta + \gamma + a_1 + a_2 + a_3 > 0,$$

$$\mu = \alpha(a_2 + a_3) + \beta(\gamma + a_1 + a_3) + \gamma(a_1 + a_2) + a_1a_2 + a_2a_3 + a_1a_3 > 0$$

and

$$\delta(u,v) = 2\alpha\beta\gamma - \{a_1\beta\gamma + a_1a_2\gamma + a_1a_3\beta + a_2a_3\alpha + a_1a_2a_3\}. \quad (2)$$

Note that $\delta(u,v) = \det(B-T)$. The sign of δ plays an important role here.

First, P is monotone increasing for $\lambda > 0$. In fact

$$\frac{dP}{d\lambda} = 3\lambda^2 + 2\sigma\lambda + \mu > 0 \quad \text{for } \lambda > 0.$$

Therefore, if $P(0) = -\delta(u,v) < 0$, then there is a unique real, positive eigenvalue $\lambda^*(u,v)$. Moreover, if λ_1 and λ_2 denote the other two eigenvalues, then either (a) they are real and both are negative, or (b) they are imaginary (i.e., $\lambda_1 = \bar{\lambda}_2$). In the last case $\lambda^* + \lambda_1 + \bar{\lambda}_1 = -\sigma < 0$, so $2 \operatorname{Re} \lambda_1 < -\lambda^* < 0$. Thus, in either case, if $\delta(u,v) > 0$, then $B-T$ has one positive, real eigenvalue and two other eigenvalues having negative real parts.

We denote by $\lambda^*(u,v)$ the eigenvalue of $B-T$ that has largest (i.e., right most) real part. We have just seen that if $\delta(u,v) > 0$, then $\lambda^*(u,v) > 0$. Let ϕ^* denote the corresponding eigenvector; i.e.,

$$(B-T)\phi^* = \lambda^*\phi^*,$$

and let ψ^* denote the corresponding adjoint eigenvector; i.e.,

$$(B-T)^{tr} \psi^* = \lambda^* \psi^*.$$

These are normalized so that

$$\psi^* \cdot \phi^* = 1 \quad \text{and} \quad \psi^* \cdot \psi^* = 1.$$

The components of ϕ^* are denoted by $\phi_1^*, \phi_2^*, \phi_3^*$, etc., and the notation $\phi^* \cdot \phi^*$ denotes the usual dot product of vectors: $\phi^* \cdot \phi^* = \phi_1^* \phi_1^* + \phi_2^* \phi_2^* + \phi_3^* \phi_3^*$, etc.

Thus, $\lambda^*(u_0, v_0) > 0$ if and only if $\delta(u_0, v_0) > 0$ (i.e., $E > 0$), so an explosion occurs according to our definition. If $E < 0$, then $\lambda^*(u_0, v_0) < 0$, and no explosion occurs since termination reactions dominate.

b. Explosion Mode Decomposition

When $\delta(u_0, v_0) > 0$, we refer to ϕ^* as the *explosion* (or *branching*) *mode* since it gives the combination of radical concentrations which will grow initially with amplification rate $\lambda^*(u_0, v_0)$.

Any combination of radical concentrations can be written as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = c\phi^* + \Omega \quad \begin{pmatrix} \text{Explosion} \\ \text{Mode} \\ \text{Decomposition} \end{pmatrix}$$

where the scalar c , called the explosion mode amplitude, is defined by

$$c = \psi^* \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix};$$

and $\Omega = \begin{pmatrix} x \\ y \\ z \end{pmatrix} - c\phi^*$ satisfies $\psi^* \cdot \Omega = 0$. Moreover, in the absence of initiation, the components of Ω are (strongly) damped since Ω is carried

by modes corresponding to eigenvalues of B-T having negative real parts. Therefore, we ignore Ω , substitute $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = c\phi^*$ into the equations (1) and project the radical equations onto the explosion mode ϕ^* (i.e., we apply ψ^* to both sides of the radical equations). The result is

$$\dot{u} = -(k_1\phi_2^* + k_3\phi_3^*)cu - k_0uv$$

$$\dot{v} = -(k_2\phi_1^*)cv - k_0uv$$

$$\dot{c} = \lambda^*(u,v)c + k_0uv\psi_1^* - c\psi^* \cdot \frac{d}{dt}(\phi^*(u,v)),$$

which is a system of equations for the fuel, oxidant and explosion mode amplitude.

c. Some remarks on the analysis of this system.

Several remarks must be made at this point. First, if $\psi_1^* = 0$, then the initiation reaction will not excite the branching mode of the reaction. Thus, no explosion will occur even though $\delta(u_0, v_0) > 0$. Therefore, the size of ψ_1^* must be accounted for in calculating the initiation phase of the reaction. In most of the reactions we have considered in detail, the initiation reaction has been mostly in the direction of ψ^* (i.e., $\psi_1^* \approx 1$). Next, the final term in the \dot{c} equation must be evaluated. Note that if some stoichiometric relation between u and v exists, say $v \propto u$, and $a_1 = a_2 = a_3 = a$, then $\lambda^*(u,v) \propto u$ and ϕ^* is constant (independent of u and v). This term has the form $\frac{a}{\lambda_1 - \lambda^*} + \frac{b}{\lambda_2 - \lambda^*}$ where a and b are proportional to \dot{u} and \dot{v} . Thus, if $|\lambda_1| \gg 1$ or $|\dot{u}|, |\dot{v}| \ll 1$, then this term is negligible. In general, if $\psi^* \cdot \frac{d\phi^*}{dt} = 0$, then the last term in this equation for \dot{c} vanishes, and a completely solvable system results to describe the branching phase of the reaction. In particular,

if $\psi^* \cdot \frac{d}{dt} \phi^* \approx 0$, and the initiation reaction is ignored, then

$$\dot{u} = -cu(k_1\phi_2^* + k_3\phi_3^*)$$

$$\dot{v} = -cv(k_2\phi_1^*)$$

$$\dot{c} = \lambda^*(u,v)c.$$

This system can be solved in the following way: The first two equations show that

$$\xi \frac{du}{u} = \frac{dv}{v} \quad \text{where} \quad \xi = \frac{k_2\phi_1^*}{(k_1\phi_2^* + k_3\phi_3^*)}.$$

Therefore, $v = ku^\xi$ where k is a constant. Next

$$\frac{dc}{du} = \frac{\lambda^*(u, ku^\xi)}{(k_1\phi_2^* + k_3\phi_3^*)u}.$$

Therefore

$$c = \bar{c} + \int_{\bar{u}}^u \frac{\lambda^*(u', ku'^\xi) du'}{(k_1\phi_2^* + k_3\phi_3^*)u'}$$

which describes the explosion mode's amplitude starting from post initiation concentrations (\bar{u}, \bar{c}) .

We describe a composite approximation to the reaction kinetics in the next section.

4. EXPLOSION KINETICS

Let us suppose that the initial concentrations of fuel (u_0) and oxidant (v_0) are sufficient so that $\delta(u_0, v_0) > 0$. As in the canonical problem, the reaction breaks down into three phases, initiation, explosion and termination.

Initiation: $x, y, z \ll u, v$.

At the start of the reaction, the radical concentrations are small, and the rate equations become

$$\dot{u} = \dot{v} = -k_0 uv, \quad u(0) = u_0, \quad v(0) = v_0$$

$$\dot{c} = k_0 uv \psi_1^*$$

These equations can be solved explicitly: Since $\dot{u} = \dot{v}$,

$$u = v - v_0 + u_0.$$

Therefore,

$$u(t) = u_0 e^{k_0(u_0 - v_0)t} \left\{ 1 + \left(\frac{1 - e^{k_0(u_0 - v_0)t}}{v_0 - u_0} \right) u_0 \right\}^{-1}$$

$$c(t) = \psi_1^*(u_0 - u(t)).$$

This approximation is valid as long as

$$\lambda^*(u, v)c \ll k_0 uv.$$

Explosion: $k_0 = 0, \psi^* \cdot \frac{d}{dt} \phi^* = 0$.

This phase describes the branching of the radicals. Since Ω is small (at most order $k_0 u_0 v_0$), and it does not grow in time, we ignore Ω terms, and arrive at the canonical problem

$$\dot{u} = -(k_1 \phi_2^* + k_3 \phi_3^*)cu$$

$$\dot{v} = -k_2 cv \phi_1^*$$

$$\dot{c} = \lambda^*(u, v)c$$

whose solution was described in the last section. This approximation remains valid as long as $\psi^* \cdot \frac{d}{dt} \phi^* \sim 0$. However, when c is large, u and v can change rapidly, and so ϕ^* is expected to change.

Termination: $k_1 c \gg 1$.

In this $\lambda^*(u,v) \sim 0$, but the fuel (u) and oxidant (v) are driven rapidly to the static state: $u = 0$, $v = 0$. As a consequence, c decays through the termination reactions; thus

$$\dot{c} = \lambda^*(0,0)c.$$

Composite solution.

A detailed matching procedure can be formulated to combine these approximations of various phases of the reaction into a composite approximation to the kinetics. However, the portion of the solution which carries the most important information is in the initiation and explosion phases, and for typical parameter values these two phases can be approximated by

$$\begin{aligned} v &\propto u \\ c &= - \int_{u_0}^u \frac{\lambda^*(\bar{u}, k\bar{u})}{(k_1 \bar{\phi}_2^* + k_3 \bar{\phi}_3^*)} \frac{d\bar{u}}{\bar{u}}. \end{aligned} \quad (4)$$

This approximation is (roughly) valid until the turning point is reached (i.e., $\lambda^*(\bar{u}, k\bar{u}) = 0$). It is illustrated in several numerical examples presented in the next section.

5. NUMERICAL SIMULATION OF THE H_2-O_2 REACTION.

We present here several illustrative calculations using typical reaction rates listed by Semenov (1959). These are based on a version of Gear's

package [4] and the EISPACK package [5] for finding eigenvectors and eigenvalues of matrices. (See Table 4)

6. DISCUSSION

The explosion mode method of analysis presented here clarifies the mechanisms of H_2-O_2 explosions. In particular, it gives a quantitative criterion ($\delta = 0$) for determining when initial concentrations of fuel and oxidant are sufficient to sustain an explosion. These remarks are based on our definition of explosion: An explosion occurs when branching initially dominates termination.

The formula (4) is taken here as a good approximation to the solution up to the turning point. In fact, if we take λ^* to be constant in (4), then we have

$$\bar{c}_{\text{turning point}} = \frac{\lambda^*}{k_1\phi_2^* + k_3\phi_3^*} \ln\left(\frac{\bar{u}}{u_0}\right) \quad (5)$$

where \bar{c} and \bar{u} are the values of c and u at the turning point. Table 3 shows a comparison of these values with those obtained from the simulations. The numerical evidence presented here supports this approximation technique. We have also found this approach to be useful in studying the H_2 -Air combustion reaction. In that case there are for certain parameter values, two explosion and two damped modes, so that the analysis becomes more intricate. Note that in applying the method to other problems, the branching termination matrix $B-T$ can be easily identified by writing the rate equations in the linear form indicated in (1) and then evaluating the coefficient matrix at the initial concentration levels. In the case of the H_2-O_2 reaction, this matrix is $\begin{pmatrix} 0 & 0 \\ 0 & B_0-T \end{pmatrix}$.

There are no significant numerical problems in solving these equations on a computer. However, there is an interesting numerical analysis aspect to the explosion mode analysis. What stiffness there is in the problem is due to the eigenvalues of $B-T$ which have negative real parts. These are summarized for the five simulations in Table . As stated, earlier Gear's package had no trouble solving these problems. However, the explosion mode projection is actually a projection onto a subdominant mode, and so avoids what stiffness there is. To implement this scheme numerically, computational effort must be invested in evaluating the spectrum of $B-T$, and then solving the resulting canonical problem.

*Research supported in part by the Department of Energy Grant #29284.
Computations were carried out in the Applied Mathematics Computing Laboratory
at the University of Utah.*

TABLE 1

CANONICAL REACTION DESCRIPTION

| <u>REACTION</u> | <u>RATE CONSTANT</u> | <u>TYPE OF REACTION</u> |
|---------------------------|----------------------|-------------------------|
| $A + A \rightarrow B + P$ | k_0 | Initiation |
| $A + B \rightarrow 2B$ | k | Branching |
| $B \rightarrow C$ | a | Termination |

TABLE 2

 H_2-O_2 ELEMENTARY REACTIONS

| <u>REACTION</u> | <u>RATE</u> | <u>TYPE OF REACTION</u> |
|---|-------------|-------------------------|
| $H_2 + O_2 \rightarrow H\cdot + HO_2$ | k_0 | Initiation |
| $H_2 + OH\cdot \rightarrow H\cdot + H_2O$ | k_1 | |
| $O_2 + H\cdot \rightarrow OH\cdot + O\cdot$ | k_2 | |
| $H_2 + O\cdot \rightarrow H\cdot + OH\cdot$ | k_3 | Propagation |
| $OH\cdot + W \rightarrow$ | a_1' | |
| $H\cdot + W \rightarrow$ | a_2' | |
| $O\cdot + W \rightarrow$ | a_3' | Termination |

TABLE 3

Estimated and Calculated Values of c

| <u>SIMULATION</u> | <u>$c_{\max}([H\cdot]_{\max})$</u> | <u>$\bar{c}$ (from (5))</u> |
|-------------------|---|---|
| 1 | .3E-8 | |
| 2 | - | |
| 3 | .6E-7 | .36E-7 |
| 4 | .3E-7 | |
| 5 | .6E-7 | 3.9E3 $\ln(\frac{.6E-9}{.1E-5}) \sim 1.E-6$ |

TABLE 5

EIGENVALUES OF B-T (MAXIMUM)

SIMULATIONS*Eigenvalues*

| | |
|---|------------------------|
| 1 | 3.0E1, -2.4E7, -2.8E4 |
| 2 | -5.2E1, -3.2E3, -2.9E4 |
| 3 | 3.8E3, -2.3E4, -2.8E5 |
| 4 | 3.E2, -2.4E4, -2.8E5 |
| 5 | 3.9E3, -2.3E5, -2.8E6 |

Simulation 1. Low initial concentrations, slow quenching.

$$k_0 = 60, \quad k_1 = 2.3E11, \quad k_2 = 4.02E9, \quad k_3 = 2.82E12$$

$$a_1 = 92, \quad a_2 = 8, \quad a_3 = 92$$

$$u_0 = 1.E-8, \quad v_0 = .5E-8.$$

| <u>T(SeCS)</u> | <u>H₂</u> | <u>O₂</u> | <u>H.</u> | <u>OH.</u> | <u>O.</u> | <u>λ*</u> | <u>φ*</u> |
|----------------|----------------------|----------------------|-----------|------------|-----------|-----------|------------------|
| 0 | 1.E-8 | .5E-8 | 0 | 0 | 0 | 30 | (1., 2E-2, 7E-4) |
| .01 | 1.E-8 | .5E-8 | 3E-17 | 5E-19 | 2E-20 | 30 | " |
| .1 | 1E-8 | .5E-8 | 1.5E-15 | 2.5E-17 | 1E-18 | 30 | " |
| .2 | 1E-8 | .5E-8 | 3E-14 | 5E-16 | 2E-17 | 30 | " |
| .3 | 1E-8 | .5E-8 | 1E-12 | 2E-14 | 8E-16 | 30 | " |
| .4 | 1E-8 | .5E-8 | 1.3E-11 | 2.1E-13 | 9E-15 | 30 | " |
| .5 | .96E-8 | .48E-8 | 2.3E-10 | 3.8E-12 | 1.6E-13 | 29 | " |
| .6 | .35E-8 | .27E-8 | .29E-8 | .7E-10 | .3E-11 | 11 | " |
| .65 | .4E-9 | .14E-8 | .35E-8 | .2E-9 | .2E-10 | -3.1 | (1., 5E-2, 3E-3) |
| .7 | .3E-14 | .8E-9 | .2E-8 | .1E-9 | .9E-10 | -11 | (1, 4E-2, 4E-2) |
| .8 | .3E-20 | .5E-9 | .73-9 | .2E-10 | .2E-10 | -10 | (1, .02, .02) |

Simulation 2. Low initial concentrations, fast quenching (E 0)

k_i as in Simulation 1.

$$a_1 = 920, a_2 = 80, a_3 = 920$$

| <u>I</u> | <u>H₂</u> | <u>O₂</u> | <u>H·</u> | <u>OH·</u> | <u>O·</u> | <u>λ*</u> | <u>φ*</u> |
|----------|----------------------|----------------------|-----------|------------|-----------|-----------|-----------------|
| 0 | 1E-8 | .5E-8 | 0 | 0 | 0 | -52 | (1, .01, 7.E-4) |
| 1E5 | .9E-8 | .46E-8 | .5E-16 | .6E-18 | .3E-19 | -54 | (1, .01, .7E-4) |

Changes in all components are monotonic.

Simulation 3. Moderate initial conditions, slow quenching

$$(\delta(u_0, v_0) > 0)$$

| <u>T</u> | <u>H₂</u> | <u>O₂</u> | <u>H·</u> | <u>OH·</u> | <u>O·</u> | <u>λ*</u> | <u>φ*</u> |
|----------|----------------------|----------------------|-----------|------------|-----------|-----------|------------------|
| 0 | 1.E-7 | .5E-7 | 0 | 0 | 0 | 390 | (1., .02, 7.E-4) |
| .01 | 1E-7 | .5E-7 | .6E-13 | 1E-15 | .4E-16 | 390 | " |
| .02 | 1E-7 | .5E-7 | .2E-11 | .4E-13 | .2E-14 | 390 | " |
| .03 | 1E-7 | .5E-7 | .6E-10 | .1E-11 | .4E-13 | 380 | " |
| .04 | .94E-7 | .48E-7 | .4E-8 | .7E-10 | .3E-11 | 370 | " |
| .05 | .1E-7 | .2E-7 | .6E-7 | .25E-8 | .1E-9 | 140 | (1., .05, 2.E-3) |
| .051 | .9E-9 | .1E-7 | .6E-7 | .6E-8 | .7E-9 | 53 | (.95, .3, .02) |
| .52 | .2E-11 | .1E-7 | .6E-7 | .9E-8 | .3E-8 | -48 | (.6, .6, .5) |

In this case the turning point ($\lambda^* = 0$) occurs at time $T = .515$. At this value, $\phi^* = (.8, .6, .1)$. Thus, ϕ^* remaining approximately constant up to the turning point of the reaction, and then changes significantly thereafter, due to rapid changes in $[H_2]$.

Simulation 4. Moderate initial conditions, fast quenching

$$(\delta(u_0, v_0) > 0$$

$$a_1 = 920, \quad a_2 = 80, \quad a_3 = 920$$

| <u>T</u> | <u>H₂</u> | <u>O₂</u> | <u>H·</u> | <u>OH·</u> | <u>O·</u> | <u>λ*(u,v)</u> | <u>φ*</u> |
|----------|----------------------|----------------------|-----------|------------|-----------|----------------|-----------------|
| 0 | .1E-6 | .5E-7 | 0 | 0 | 0 | 300 | (1., .02, 7E-4) |
| .01 | .1E-6 | .5E-7 | .3E-13 | .4E-15 | .2E-16 | 300 | " |
| .02 | .1E-6 | .5E-7 | .6E-12 | .1E-13 | .4E-15 | 300 | " |
| .03 | .1E-6 | .5E-7 | .6E-11 | .1E-12 | .5E-14 | 300 | " |
| .05 | .1E-6 | .5E-7 | .2E-9 | .4E-11 | .2E-12 | 300 | " |
| .05 | .96E-7 | .49E-7 | .7E-9 | .4E-10 | .2E-11 | 290 | " |
| .06 | .4E-7 | .3E-7 | .3E-7 | .7E-9 | .3E-10 | 120 | (1., .02, .001) |
| .07 | .3E-13 | .8E-8 | .2E-7 | .1E-8 | .9E-9 | -110 | (1., .04, .04) |
| .08 | .3E-19 | .5E-8 | .7E-8 | .2E-9 | .2E-9 | -98 | " |

Simulation 5. High initial concentrations, slow quenching

$$(\delta \gg 0)$$

$$a_1 = 92, \quad a_2 = 8, \quad a_3 = 92$$

| <u>I</u> | <u>H₂</u> | <u>O₂</u> | <u>H·</u> | <u>OH</u> | <u>O·</u> | <u>λ*</u> | <u>φ*</u> |
|----------|----------------------|----------------------|-----------|-----------|-----------|-----------|-----------------|
| 0 | .1E-5 | .5E-6 | 0 | 0 | 0 | 3.9E3 | (1., .02, 7E-4) |
| .001 | .1E-5 | .5E-6 | .3E-12 | .6E-14 | .2E-15 | 3.9E3 | " |
| .002 | .1E-5 | .5E-6 | .3E-10 | .5E-12 | .2E-13 | 3.9E3 | " |
| .003 | .1E-5 | .5E-6 | .9E-9 | .2E-10 | .6E-12 | 3.9E3 | " |
| .004 | .9E-6 | .47E-6 | .5E-7 | .9E-9 | .4E-10 | 3.7E3 | " |
| .005 | .1E-7 | .1E-6 | .6E-6 | .6E-7 | .7E-8 | 850 | (.96, .28, .01) |
| .0051 | .6E-10 | .1E-6 | .6E-6 | .9E-7 | .3E-7 | -17 | (.1, .97, .2) |
| .00525 | E-22 | .8E-7 | .6E-6 | .1E-6 | .7E-7 | -9.2 | (0, .99, .2) |

This calculation again illustrates that ϕ^* may change rapidly near the turning point.

REFERENCES

1. Giddings, J. C., and Hirschfelder, J. O., *Sixth Symposium on Combustion*, Reinhold Publishing Corp., New York, 1975, p. 199.
2. Kermack, W. O., and McKendrick, A. G., *Proc. Royal Soc. London Sec. A*, *115*, 700-721 (1927).
3. Semenov, N. N., *Some Problems in Chemical Kinetics*, Pergamon, New York, 1959, Vol. 1, p. 47.
4. Skeel, R. D., and Kong, A. K., Report UIUCDCS-R-76-800, Department of Computer Science, University of Illinois at Urbana-Champaign, 1976.
5. Smith, B. T., Boyle, J. M., Dongarra, J. J., Garbow, B. S., Ikebe, Y., Klena, V. C., and Moler, C. B., *Matrix Eigen System Routines - ELSPACK Guide*, Springer Verlag, New York, 2nd Edition, 1976.

An Overview of the Highly Oscillatory Initial Value Problem

W. L. Miranker

**IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598**

Typed by: J. Genzano

Abstract: We review computational methods for approximating the highly oscillatory problem which utilize a functional of the solution such as its smooth part as a meaningful solution description. We review highly oscillatory recurrences by the two-time technique and give several applications. Then the highly oscillatory ordinary differential equation is treated by three methods: the two-time method, an extrapolation method and finally an averaging method. The last treatment is accompanied by illustrative computations.

1. Introduction

The highly oscillatory problem is particularly difficult since the rapidly changing behavior of the solution is an ever present feature and not a transient phase to be gotten through. Thus most approaches to the highly oscillatory problem abandon the customary procedure of approximating the solution pointwise (at least on a mesh). Instead some functional of the solution such as its "smooth part" or some "running mean" is accepted as a meaningful description and it is this functional which is approximated through computation. This overview will deal with several methods which take this approach. (For methods with the classical pointwise approach see [1] and [12].)

We begin this review with highly oscillatory recurrences since there are many applications for them, several of which are reviewed here. These recurrences are treated by the two-time technique. We then turn to the initial value problem for differential equations. We do not explicitly cite applications for the latter, but we do illustrate computations on a model problem. Applications are to be found in circuit analysis and orbiting motions, for example. We demonstrate three approaches to the differential problem. The first, which is most common, is through singular perturbation methods and in particular, multi-time techniques. The third is a method of averaging which is independent of singular perturbation methods. The second method, the extrapolation method, falls in between.

References are given to source material, but all of the work reviewed may also be found discussed in detail in my monograph on stiff equations [8].

2. Recurrences

Consider first the model

$$x_{n+1} = (A + \epsilon B)x_n, \quad x_0 \text{ given.}$$

where $x \in \mathbb{R}^p$ and A and B are $p \times p$ constant matrices. To illustrate the behavior of this recurrence it is convenient to suppose for the time being that A and B commute. In this case

$$x_n = (A + \epsilon B)^n x_0 = A^n (I + \epsilon A^{-1} B)^n x_0.$$

Then

$$x_n = A^n \exp [A^{-1} B \epsilon n] x_0 (1 + n \epsilon O(\epsilon)).$$

This demonstrates the composition of x_n as it develops on two scales: a rapidly developing part (viz, A^n) and a slowly developing part (viz, $\exp [A^{-1} B \epsilon n]$). In many cases the rapidly varying part is akin to noise from which the more meaningful part of the solution needs to be separated. How this separation may be performed in more general cases (i.e. noncommutative and nonlinear cases) has been demonstrated in [6], where a two-time formalism for recurrences is developed.

We continue by describing three applications, and we illustrate the separation of the scales achieved by use of a two-time technique.

i) Training algorithm

The so-called training algorithm [4] arises in pattern recognition and corresponds to the mathematical problem of determining a separating hyperplane. Such a hyperplane is specified by a vector $w \in \mathbb{R}^{p+1}$ to which the following recurrence may be made to

converge (in finitely many steps):

For fixed $\theta > 0$ and $w_0 \in \mathbb{R}^{p+1}$, define w_1, w_2, \dots as follows

$$w_{n+1} = w_n + x_n S\left(\frac{w_n \cdot x_n}{\theta}; x_n\right),$$

where

$$S\left(\frac{w \cdot x}{\theta}; x\right) = \begin{cases} 1, & w \cdot x \leq \theta \\ 0, & w \cdot x > \theta \end{cases} \text{ and } x \in A^*,$$

$$\begin{cases} -1, & w \cdot x \geq -\theta \\ 0, & w \cdot x < -\theta \end{cases} \text{ and } x \in B^*.$$

Here A^* and B^* are the finite point sets in \mathbb{R}^{p+1} being separated. Setting $w_n = \theta z(n)$ and $\varepsilon = \theta^{-1}$, the recurrence relation for w_n becomes

$$z(n+1) = z(n) + \varepsilon x_n S(z(n) \cdot x_n; x_n)$$

while in the definition of S , w is changed to z and θ is replaced by unity.

The two-time formalism specifies the slowly evolving aspect of the trajectory defined by this recurrence for $z(n)$ as a function $z_0(s)$ where moreover $z(n) = z_0(\varepsilon n) + O(\varepsilon)$.

Indeed z_0 is the solution of the following non-stiff problem:

$$\frac{dz_0}{ds} = \overline{xS}(z_0),$$

where the average

$$\overline{xS}(z_0) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} x_k S(z_0 \circ x_n; x_k).$$

(In applications this average is finitely computable.) Computation of z_0 is straight forward and leads to a limiting vector which is an appropriate w sought by the training algorithm.

ii) A population genetics model

In a large population of diploid organisms having discrete generations, the genotypes determined by one locus having two alleles A and a , divide the population into three groups of types AA , Aa , and aa , respectively. Suppose that the gene pool carried by this population is in proportion p_n of type A in the n th generation. It follows [2] that

$$p_{n+1} = p_n + \frac{p_n(1-p_n)[(w_{11}-w_{12})p_n + (w_{21}-w_{12})(1-p_n)]}{w_1 p_n^2 + 2w_{12} p_n(1-p_n) + w_{22}(1-p_n)^2},$$

where w_{11} , w_{12} and w_{22} are the relative fitnesses of the genotypes AA , Aa , and aa , respectively. If the selective pressures act slowly, i.e., if $w_{11} = 1 + \epsilon\alpha$, $w_{12} = 1$ and $w_{22} = 1 + \epsilon\beta$ then

$$p_{n+1} = p_n + \frac{\epsilon p_n(1-p_n)[(\alpha-\beta)p_n + \beta]}{1 + O(\epsilon)}.$$

The two time formalism tells us that

$$p_n = P(\epsilon n) + O(\epsilon),$$

where

$$\frac{dP}{ds} = P(1-P)[(\alpha-\beta)P + \beta],$$

and this simple non-stiff differential equation is easily solved numerically.

iii) Regression analysis

Let $g(w)$ be a function with a unique root \hat{w} and such that $g(w)(w-\hat{w}) > 0$, $w \neq \hat{w}$. Let z_k , $k = 0, 1, \dots$ be a sequence of identically distributed random variables with mean zero and unit variance. The Robbins-Munro algorithm for approximating \hat{w} is specified by the following recurrence:

$$w(k+1) = w(k) - \epsilon \alpha_k [g(w(k)) + \sigma z_k].$$

$g(w(k)) + \sigma z_k$ is a noisy measurement of $g(w(k))$ so that the recurrence specifies a very chaotic behavior for $w(k)$. However, the two-time methodology asserts that

$$w(k) = W_0(\epsilon k)(1 + O(\epsilon)),$$

where

$$\frac{dW_0}{ds} = -\bar{\alpha} g(W_0)$$

and

$$\bar{\alpha} = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{j=0}^k \alpha_j.$$

The differential equation for W_0 is simple to solve, and for appropriate g , its solutions converge to the equilibrium value \hat{w} , the root which is sought.

3. Two-time Methods in Stiff Differential Equations

The two-time method

The model highly oscillatory problem is taken to be

$$\epsilon \frac{d\mathbf{u}}{dt} = (A + \epsilon B)\mathbf{u}, \quad t \in (0, T],$$

$$\mathbf{u}(0) = \mathbf{u}_0,$$

where \mathbf{u} is an n -vector and A and B are $n \times n$ matrices. The solution is

$$\begin{aligned} \mathbf{u} &= e^{[(A + \epsilon B)t/\epsilon]} \mathbf{u}_0 \\ &= e^{A\tau + Bt} \mathbf{u}_0 \end{aligned}$$

where $\tau = t/\epsilon$. τ is called the fast time and t the slow time. If A and B commute the dependence of the solution on these two time scales separates, viz,

$$\mathbf{u} = e^{A\tau} e^{Bt} \mathbf{u}_0,$$

and in principle, each of the factors here could be determined separately and without computational difficulty. When A and B do not commute this separation is not so readily available, and moreover, it is not necessarily the case that the development of the solution on the t -scale is even meaningful to approximate numerically. To treat this case we employ the method of two-times. We suppose that the initial condition has the form

$$\mathbf{u}(0) = \sum_{r=0}^{\infty} a_r \epsilon^r$$

and that the solution of the initial value problem has an approximation in the form of

a general two-time expansion:

$$u = \sum_{r=0}^{\infty} u_r(t, \tau) \epsilon^r.$$

We take the leading term $u_0(t, \tau)$ of the expansion as an approximation to the solution of the initial value problem. The method two-times ([9]) specifies $u_0(t, \tau)$ as follows

$$u_0(t, \tau) = \Phi(\tau) \tilde{v}_0(t),$$

where the fundamental matrix $\Phi(\tau)$ is given by

a)
$$\Phi_\tau = A\Phi, \quad \Phi(0) = I.$$

Further

b)
$$\frac{d\tilde{v}_0}{dt} = \bar{B}\tilde{v}_0, \quad \tilde{v}_0(0) = a_0.$$

where

c)
$$\bar{B} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau \Phi^{-1}(\sigma) B \Phi(\sigma) d\sigma.$$

(For the development of this two-time methodology in a more general non-linear setting see [5].) The specification of $u_0(t, \tau)$ leads to the following numerical algorithm

Algorithm

- i) Solve (a) on a mesh of increment k in the τ -scale by some self starting numerical method, obtaining the sequence $\Phi(jk)$, $j = 0, \dots, N$.
- ii) Using the values $\Phi(jk)$ obtained in (i), approximate \bar{B} by truncating the limit of τ

integration and replacing the integral in (c) by a quadrature formula, say

$$\bar{B} = \frac{1}{N} \sum_{j=0}^N c_k \Phi^{-1}(jk) B \Phi(jk).$$

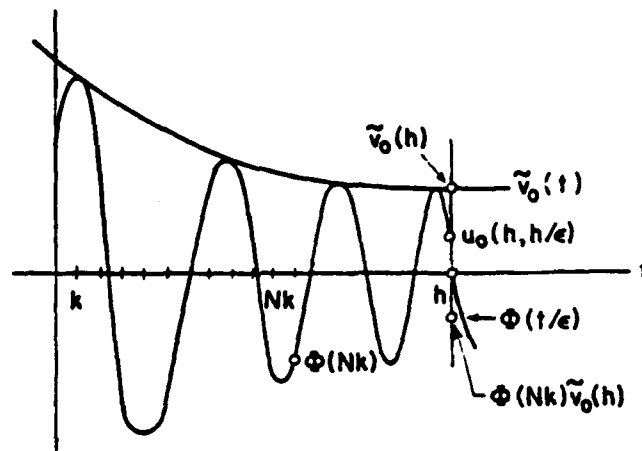
The integer N is determined by a numerical criterion which insures that the elements of the matrix \bar{B} are calculated to within some desired accuracy.

iii) With \bar{B} (approximately) determined in (ii), solve (b) for $\tilde{v}_0(h)$ by some self starting numerical method.

iv) Compute $u_0(h, Nk) = \Phi(Nk) \tilde{v}_0(h)$, and take this as the approximation for $u(h)$.

The approximation $u_0(h, Nk)$ produced by this algorithm is not an approximation to $u(h, Nk)$ (i.e., is not a pointwise approximation in the conventional sense of numerical analysis). The approach of this algorithm takes pointwise approximation as ill-conditioned (and meaningless). Indeed approximation on the fast time scale is abandoned. In fact, $u_0(h, Nk)$ is an approximation to $u(t, Nk)$ for t near h and not necessarily for $t = h$.

In the following figure we schematize the computation. Of course, in practice ϵ will be extremely small so that unlike the schematic an enormous number of oscillations of Φ will occur in the t interval $[0, h]$. Notice how far the computed answer $\Phi(Nk) \tilde{v}_0(h)$ may be from the customary approximation to the solution, $u_0(h, h/\epsilon)$.



The fundamental matrix $\Phi(\tau)$ is composed of modes corresponding to the eigenvalues of A . Since the eigenvalues of A lie in the closed left half plane, the profile for (a component of) Φ will, after some moderate number of cycles, settle down to an (almost) periodic function. Thus the set of mesh points $\{jk \mid j = 0, \dots, N\}$ may be expected to extend over just these cycles (approximately).

4. The Extrapolation Method

The exploitation of singular perturbation theory for the development of numerical techniques for stiff differential equations usually proceeds with the numerical determination of values of one or more terms in the asymptotic expansion supplied by that theory. The extrapolation method [7] which we will now review is a way to break through this limitation of approach.

Consider the following nonlinear analogue of the model problem which we have been

discussing.

$$\frac{dx}{dt} = f(t/\epsilon, x), \quad x(0) = \xi.$$

where $x, f, \xi \in \mathbb{R}^n$ and where $f(\tau, \cdot)$ is an almost periodic function of τ . Multi-time perturbation methods lead to the approximation

$$x(t, \epsilon) = x_0(t) + \epsilon x_1(t, t/\epsilon) + O(\epsilon^2),$$

where x_0 is determined from the initial value problem

$$\frac{dx_0}{dt} = \bar{f}(x_0), \quad x_0(0) = \xi.$$

\bar{f} is the average of f , defined by

$$\bar{f}(x_0) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T f(\tau, x_0) d\tau.$$

The coefficient x_1 is determined from the formula

$$x_1(t, t/\epsilon) = \tilde{x}_1(t) + \int_0^{t/\epsilon} [f(\tau, x_0) - \bar{f}(x_0)] d\tau.$$

In this formula, \tilde{x}_1 is determined at a later step in the perturbation scheme. Since it will not be needed here, it is not discussed further. Thus,

$$x(t, \epsilon) = x_0(t) + \epsilon \left\{ \tilde{x}_1(t) + \int_0^{t/\epsilon} [f(\tau, x_0) - \bar{f}(x_0)] d\tau \right\} + O(\epsilon^2).$$

We seek to determine certain larger values ϵ' of ϵ . The initial value problem is solved with these larger values of ϵ and an appropriate extrapolation determines an approximation to $x(t, \epsilon)$ itself.

First the average \bar{f} must be determined. A straightforward numerical approximation of \bar{f} can be costly, but there is the possibility of accelerating this computation by a second difference method which we now describe.

In most applications, the integral of the almost periodic function f has the form

$$V(T, x) = \bar{f}(x)T + p(T, x),$$

where p is an almost periodic function of its first argument and which has mean zero.

Thus, given a tolerance δ , there is a δ -translation number $\mathcal{F}(\delta, x)$ such that

$$\|p(T + \mathcal{F}(\delta, x), x) - p(T, x)\| < \delta$$

for all $T \geq 0$; in particular, since $p(0, x) = 0$, then $|p(\mathcal{F}(\delta, x))| < \delta$.

To find candidates for \mathcal{F} note that

$$V(2T, x) - 2V(T, x) = p(2T, x) - 2p(T, x).$$

In particular for $T = \mathcal{F}(\delta, x)$, we have that

$$\begin{aligned} V(2\mathcal{F}, x) - 2V(\mathcal{F}, x) &= p(2\mathcal{F}, x) - p(\mathcal{F}, x) - p(\mathcal{F}, x) + p(0, x) \\ &= O(\delta). \end{aligned}$$

Thus, any δ -translation number of p makes this second difference of order δ . Unfortunately, the converse does not hold; in particular, $\|V(2T, x) - 2V(T, x)\|$ may be small while $\|p(T, x)\|$ is not small.

Still, by tabulating

$$V(2T, x) - 2V(T, x),$$

candidates for $\mathcal{F}(\delta, x)$ can be found and tested by comparing the values of $V(T, x)/T$ for several of them, since these should all approximate $\bar{f}(x)$. In practice, this method is no worse than the direct calculation of $\bar{f}(x)$, and in periodic cases, it reliably gives $\bar{f}(x)$ after calculation over one period.

Thus we use

$$\frac{1}{\mathcal{F}(\delta, x)} \int_0^{\mathcal{F}(\delta, x)} f(\tau, x) d\tau$$

as an approximation to \bar{f} .

The extrapolation method proceeds by choosing an appropriate value T which represents a time at which rapid motions can be ignored. We pick T to be a δ -translation number of $p(\tau, x)$. Then, in particular,

$$\frac{p(2T, x) - 2p(T, x)}{T} = \frac{1}{T} \int_T^{2T} [f(\tau, x) - \bar{f}(x)] d\tau = O(h^p),$$

for $x = \xi + O(h)$. The existence of such a value of T follows from viewing this equation as the statement that T is an approximation to a δ -translation number. Such a T value must be found, perhaps using the second difference method just described or additional knowledge about a specific problem being studied.

Once a T value is found we define

$$\epsilon' = h/T,$$

and then we calculate $x(h, \epsilon'/2)$ and $x(h, \epsilon')$ from the initial value problem by a p th-order numerical method. It follows from the form of the asymptotic expansion for

$x(t, \varepsilon)$ above that

$$\begin{aligned} 2x(h, \varepsilon'/2) - x(h, \varepsilon) &= x_0(h) + \varepsilon' \int_T^{2T} [f(\tau, x_0(h)) - \bar{f}(x_0(h))] d\tau + O((\varepsilon')^2) \\ &= x_0(h) + O(h^{p+1}) + O\left(\left(\frac{h}{T}\right)^2\right). \end{aligned}$$

On the other hand,

$$\begin{aligned} x(h, \varepsilon) &= x_0(h) + \varepsilon \tilde{x}_1 + \varepsilon \int_0^{h/\varepsilon} [f(\tau, x_0(h)) - \bar{f}(x_0(h))] d\tau + O(\varepsilon^2) \\ &= x_0(h) + O(\varepsilon), \end{aligned}$$

since

$$\int_0^{h/\varepsilon} [f(\tau, x_0(h)) - \bar{f}(x_0(h))] d\tau = O(1).$$

Therefore,

$$x(h, \varepsilon) = 2x(h, \varepsilon'/2) - x(h, \varepsilon') + O(\varepsilon) + O(h^{p+1}) + O\left(\left(\frac{h}{T}\right)^2\right).$$

This formula gives the extrapolation method for calculating $x(h, \varepsilon)$.

5. A Method of Averaging

We now describe an example of a numerical method which is independent of singular perturbation methodology (see [10]). For other examples of methods which are likewise independent see [1] and [2] the latter employing aliasing.

Consider the following model problem

$$\ddot{x} + \lambda^2 x = f(t), \quad t \in (0, T).$$

When $f(t) = \lambda^2 \sin t$, this problem has the following family of solutions

$$x(t) = a \sin \lambda t + \frac{\sin t}{1 - 1/\lambda^2}.$$

For λ large, this solution family consists of a high frequency carrier wave, $a \sin \lambda t$, modulated by a slow wave, $\sin t / (1 - 1/\lambda^2)$. The specification of the value at a point of such a function is all ill-conditioned problem.

The linear multistep class of methods is highly desirable for numerical analysis since these methods are easy to use and easy to analyze. However these methods consist of a linear combination of unstable functionals namely, solution values and values of solution derivatives at points. The method of averaging replaces these unstable functionals by stable ones, thereby producing a class of linear multistep methods suitable for the highly oscillatory problem. We suppose that the stable functionals provide information about the solution being sought, and (subject to a process like mesh refinement) that the stable functionals furnish as adequate a description of the solution as is needed.

Let r, s and N be positive integers, let $h = T/N$ and let $t_i = ih, i = 0, \pm 1, \dots$ be the points of a mesh. Let $z(t)$ be a functional of x which can be calculated at each mesh point. Then we seek to determine $y_n = y(t_n)$, in terms of $y_{n-i}, i = 1, \dots, r$ and $z_{n-i} = z(t_{n-i}), i = 0, 1, \dots, s$ by means of the linear multistep formula

$$\sum_{i=0}^r a_i y_{n-i} + \sum_{i=0}^s b_i z_{n-i} = 0, \quad n = 0, 1, \dots, N.$$

We choose $y(t)$ to be

$$y(t) = \int_{-\infty}^{\infty} k(t-s)x(s)ds,$$

where

$$k(z) = \frac{1}{\Delta} \begin{cases} 1, & -\Delta < z < 0, \\ 0, & \text{otherwise.} \end{cases}$$

Thus $y(t)$ represents the average of $x(t)$ over the interval $[t-\Delta, t]$.

The functional $z(t)$ is chosen to be $[d^2/dt^2 + \lambda^2]x(t)$, i.e., $f(t)$, which we suppose can be stably calculated at each mesh point. Thus with a change in normalization, the linear multistep formula may be written as

$$y_n = \sum_{i=1}^r c_i y_{n-i} + h^2 \sum_{i=0}^s d_i f_{n-i}.$$

Notions of local truncation error and of order of accuracy of such generalized linear multistep method (for functionals) are introduced. They are analogous to and generalizations of the traditional ones. For example, let

$$s_0 = 1 \text{ and } s_j = -c_j, j = 1, \dots, r,$$

and let

$$L = \Delta/h.$$

Then

$$m_\ell = \frac{1}{(\ell+1)!} \sum_{k=1}^{\ell+1} \binom{\ell+1}{k} L^{k-1} \sum_{j=0}^r j^{1+\ell-k} s_j - \frac{h^2 \lambda^2}{\ell!} \sum_{j=0}^s j^\ell d_j - \frac{1}{(\ell-2)!} \sum_{j=0}^s j^{\ell-2} d_j, \quad \ell = 0, 1, \dots$$

are so-called generalized moments of the method. The equations $m_\ell = 0$, $\ell = 0, 1, \dots$ are generalized moment conditions.

View the equations $m_\ell = 0$, $\ell = 0, \dots, r-1$ as r equations for the r unknowns s_j , $j = 1, \dots, r$. The ℓ th row of the resulting coefficient matrix which has as its j th term

$$\frac{1}{(\ell+1)!} \sum_{k=1}^{\ell} \binom{\ell+1}{k} L^{k-1} j^{1+\ell-k},$$

is a linear combination of the first ℓ rows of the Vandermonde matrix. Thus the system of r equations has a solution in this case. Indeed by choosing the d_j , $j = 0, \dots, s$ to be proportional to λ^{-2} , we obtain a solution for the s_j , $j = 1, \dots, r$ which is $O(1) + O(\lambda^{-2})$.

If the coefficients s_j , $j = 1, \dots, r$ and d_j , $j = 0, \dots, s$ are chosen as solutions of the generalized moment equations $m_\ell = 0$, $\ell = 0, \dots, p$, we may obtain an estimate of the local truncation error which is $O(h^{p+1})$.

In addition to notions of local accuracy a notion of stability for generalized linear multistep methods may be introduced. One such condition is quite analogous to the

classical one; namely, that the polynomial

$$S(z) = \sum_{j=0}^r s_j z^{r-j}$$

obeys the root condition. Then combining notions of local accuracy and of stability, a global error estimate may be obtained.

Examples

We now consider some examples of these methods in which the coefficients are determined by the generalized moment conditions. In particular, we have for $l = 0, 1$ and 2, respectively:

$$0. \quad m_0 \equiv \sum_{j=0}^r s_j - h^2 \lambda^2 \sum_{j=0}^s d_j,$$

$$1. \quad m_1 \equiv \sum_{j=0}^r j s_j + \frac{L}{2} \sum_{j=0}^r s_j - h^2 \lambda^2 \sum_{j=0}^s j d_j,$$

$$2. \quad m_2 \equiv \frac{1}{2} \sum_{j=0}^r j^2 s_j + \frac{L}{2} \sum_{j=0}^r j s_j + \frac{L^2}{6} \sum_{j=0}^r s_j - \frac{h^2 \lambda^2}{2} \sum_{j=0}^s j^2 d_j - \sum_{j=0}^s d_j.$$

Consider the following case where the first two generalized moment conditions are satisfied.

$$A. \quad m_0 = m_1 = 0.$$

For $r = s = 1$, we get

$$c_1 = 1 - \frac{2}{L} + \frac{2}{L} h^2 \lambda^2 d_0,$$

$$d_1 = \frac{2}{h^2 \lambda^2 L} - \left(\frac{2}{L} + 1 \right) d_0.$$

In the special case $d_0 = 0$, this becomes

$$I \begin{cases} c_1 = 1 - \frac{2}{L}, \\ d_1 = \frac{2}{h^2 \lambda^2 L}. \end{cases}$$

These coefficients (i.e., c_1) obey the root condition if and only if $L \geq 1$. In the special case $d_0 = d_1$, we obtain

$$II \begin{cases} c_1 = 1 - \frac{2}{L+1}, \\ d_0 = d_1 = \frac{1}{h^2 \lambda^2} \frac{1}{L+1}. \end{cases}$$

Under the restriction $L \geq 0$, the root condition is equivalent to $L \geq 0$ for these coefficients.

For $r = s = 2$,

$$c_1 = 1 - \frac{2}{L} - \left(1 + \frac{2}{L}\right)c_2 + \frac{2}{L} \lambda^2 h^2 (d_0 - d_1),$$

$$d_1 = \frac{2}{\lambda^2 h^2 L} (1 + c_2) - \left(1 + \frac{2}{L}\right)d_0 - \left(1 - \frac{2}{L}\right)d_2.$$

In the special case $d_0 = 0$, $c_1 = c_2$, $d_1 = d_2$, this becomes

$$III \begin{cases} c_1 = c_2 = \frac{L-3}{2L}, \\ d_1 = d_2 = \frac{3}{2\lambda^2 h^2 L}. \end{cases}$$

In this case,

$$S(z) = z^2 - \frac{L-3}{2L}z - \frac{L-3}{2L}.$$

and this polynomial $S(z)$ obeys the root condition for a set of values of L which includes all $L \geq 1$.

In the special case $c_1 = c_2, d_1 = d_2 = 0$, c_1 and d_1 become

$$IV \begin{cases} c_1 = c_2 = \frac{1}{2} \frac{L}{3+L}, \\ d_0 = \frac{1}{\lambda^2 h^2} \frac{3}{3+L}. \end{cases}$$

Here

$$S(z) = z^2 - \frac{1}{2} \frac{L}{3+L} z - \frac{1}{2} \frac{L}{3+L}.$$

This polynomial obeys the root condition for a set of values of L which includes all $L > 0$.

In the special case $c_1 = c_2, d_0 = d_1 = d_2$, we obtain

$$V \begin{cases} c_1 = c_2 = \frac{1}{2} \frac{L-1}{L+1}, \\ d_0 = d_1 = d_2 = \frac{2}{3\lambda^2 h^2} \frac{1}{1+L}. \end{cases}$$

In this case, the root condition is obeyed for $L > 0$.

Now we consider a case corresponding to three generalized moment conditions.

B. $m_0 = m_1 = m_2 = 0.$

For $r = s = 1$, we get

$$c_1 = 1 - L \left(\frac{L^2}{3} + \frac{L}{2} - \frac{2}{h^2 \lambda^2} \right)^{-1},$$

$$VI \quad d_0 = \frac{1}{\lambda^2 h^2} \left[1 - L + L^2 \left(\frac{2L^2}{3} + L - \frac{4}{h^2 \lambda^2} \right)^{-1} \right],$$

$$d_1 = \frac{1}{\lambda^2 h^2} \left[-1 + L + (2L - L^2) \left(\frac{2L^2}{3} + L - \frac{4}{h^2 \lambda^2} \right)^{-1} \right].$$

Notice that the root condition is obeyed for L large and positive but is violated for $h\lambda$ small compared to L .

Remark: In all of these examples as in the general case, we see that the coefficients obtained as solutions of the moment conditions depend on λ^2 . At first sight this seems to be more restrictive than the case of the classical linear multistep formulas where the coefficients of the formula do not depend on the coefficients of the differential equation. In fact there is no such distinction. In the classical case, the coefficients of the differential equation enter into the method when it is used to approximate the differential equation, e.g., when \dot{y}_{n-i} is replaced by $f(y_{n-i}, t_{n-i})$. It is essential after all that the numerical method at some point be dependent on the equation to be solved. In the present development, this dependence occurs at the outset in the determination of coefficients and in the error analysis. In the classical case it enters in the error analysis and in the use of the methods.

Computational Experiments

We now apply the six sets of methods labeled I, II,, VI above to the model

problem:

$$\ddot{x} + \lambda^2 x = \lambda^2 \sin t,$$

$$x(0) = 0, \quad \dot{x}(0) = \frac{\lambda}{2} + \frac{1}{1 - 1/\lambda^2}.$$

Computations are made over the interval $[0, T] = [0, \pi]$. In the following table we display $h^{1/2}$ times the l^2 -norm of the global error:

$$\|e\|_{l^2} \equiv \left[h \sum_{n=0}^{[\pi/h]} e_n^2 \right]^{1/2},$$

for a set of various combinations of $h = .1, .01, \lambda = 10, 10^3, 10^5$ and $L = 1, 2, 3$ and for each of the six methods cited. Here $[\pi/h]$ denotes the integer part of π/h .

TABLE

| Method | λ | L | 1 | 2 | 3 | 1 | 2 | 3 |
|--------|-----------|-----|--------|--------|--------|---------|---------|---------|
| I | 10 | | .273 | .108 | .112 | .133 | .126 | .126 |
| | 10^3 | | .113 | .00217 | .0611 | .0283 | .00683 | .0083 |
| | 10^5 | | .112 | .00209 | .0611 | .0111 | .000106 | .00627 |
| II | 10 | | .122 | .133 | .155 | .126 | .127 | .128 |
| | 10^3 | | .00125 | .0622 | .177 | .0241 | .00926 | .0136 |
| | 10^5 | | .00104 | .0621 | .177 | .000118 | .00627 | .0125 |
| III | 10 | | .242 | .111 | .0872 | .136 | .126 | .126 |
| | 10^3 | | .0032 | .00422 | .00317 | .0294 | .00684 | .00546 |
| | 10^5 | | .0034 | .00419 | .00313 | .00023 | .00112 | .89E-6 |
| IV | 10 | | .123 | .111 | .0938 | .126 | .126 | .126 |
| | 10^3 | | .00627 | .0144 | .0244 | .0241 | .00684 | .00546 |
| | 10^5 | | .00623 | .0144 | .0244 | .000133 | .000179 | .000264 |
| V | 10 | | .144 | .152 | .156 | .127 | .127 | .128 |
| | 10^3 | | .0657 | .094 | .119 | .0249 | .0116 | .0136 |
| | 10^5 | | .0657 | .0939 | .119 | .0063 | .00942 | .0125 |
| VI | 10 | | .758E4 | .66E11 | .124 | .195E1 | .471E1 | .11E2 |
| | 10^3 | | .0447 | .0639 | .244 | .0246 | .00901 | .0253 |
| | 10^5 | | .0447 | .0639 | .244 | .00421 | .00629 | .0251 |
| h | | | .1 | | | .01 | | |

$$h^{1/2} \|e\|_{r^2}$$

To illustrate both the favorable and unfavorable effects, the table contains cases for which the methods are designed to operate well along with cases to which correspond poor or nonsensical results.

For example although the cases corresponding to $\lambda = 10$ give fair results, these cases are not stiff, and we should not expect good results. When h is decreased, improvement should occur but only for the stiff cases. The cases $\lambda = 10^3$ and $h = .01$ are not stiff, and improvement with decreasing h does not always occur in these cases. Method VI is used in some unstable cases. The stiff cases for moderate L give extremely good results as we expect.

In the systems case, the model problem is replaced by the second order system

$$\ddot{x} + \Lambda^2 x = f(x, t).$$

Here x and f are q -vectors and Λ is a $q \times q$ matrix. The coefficients c_j (and s_j) and d_j of the numerical method are replaced by $q \times q$ matrices (denoted by the same symbols). Many such formal replacements of the scalar development follow. For example, the first two generalized moments become

$$m_0 = \left(\sum_{j=0}^r s_j - h^2 \Lambda^2 \sum_{j=0}^s d_j \right) \zeta_q,$$

$$m_1 = \left(\sum_{j=0}^r j s_j + \frac{L}{2} \sum_{j=0}^r s_j - \Lambda^2 \sum_{j=0}^s d_j \right) \zeta_q,$$

where ζ_q is the q -vector all of whose components are unity.

Referring to the remark above and noting the dependence of the coefficients of the numerical method on the coefficients of the differential equation, we see from m_0 and m_1 here, the way in which the dependence appears in terms of the matrix Λ^2 , for the coefficients determined by the generalized moment conditions. It is important to take note that the coefficients depend on the matrix Λ^2 and not explicitly on eigenvalues of Λ^2 . Thus, if we know that a system is stiff, with highly oscillatory components, we may use the methods described here without having to calculate the eigenvalues of Λ^2 which cause this stiffness.

References

- [1] Amdursky, V. and Ziv, A.: "On the Numerical Solution of Stiff Linear Systems of the Oscillatory Type," *SIAM J. Ap. Math.* 33, 593-606 (1977).
- [2] Crow, J. F. and Kimura, M.: *An Introduction to Population Genetics Theory*. Harper Row, New York, 1970.
- [3] Frank, L.: "On Asymptotic Behavior of High Oscillating Solutions," *Uspechi Math. Nauk*, XXV, vip.5(155), 251-252 (1970).
- [4] Greenberg, H. J. and Konheim, A. G.: "Linear and Nonlinear Methods in Pattern Classification," *IBM J. Res. Dev.*, 8, 299-307 (1964).
- [5] Hoppensteadt, F. C. and Miranker, W. L.: "Differential Equations having Rapidly Changing Solutions: Analytic Methods for Weakly Nonlinear Systems," *J. Differential Equations*, 22, 237-249 (1976).
- [6] Hoppensteadt, F. C. and Miranker, W. L.: "Multi-time Methods for Systems of Difference Equations," *Studies in Appl. Math.*, 56, 273-289 (1977).

- [7] Hoppensteadt, F. C. and Miranker, W. L.: "An Extrapolation Method for the Numerical Solution of Singular Perturbation Problems," *SIAM SISSC* to appear.
- [8] Miranker, W. L.: *Numerical Methods for Stiff Differential Equations and Singular Perturbation Problems*, D. Reidel, Dordrecht, 1980.
- [9] Miranker, W. L. and Hoppensteadt, F.: "Numerical Methods for Stiff Systems of Differential Equations Related with Transistors, Tunnel Diodes, etc.," *Lecture Notes in Computer Science*, 10, Springer-Verlag, 416-432 (1973).
- [10] Miranker, W. L. and Wahba, G.: "An Averaging Method for the Stiff Highly Oscillatory Problem," *Math. Comp.*, 30, 383-399 (1976).
- [11] Persek, S. C. and Hoppensteadt, F. C.: "Iterated Averaging Methods for Systems of Ordinary Differential Equations with a Small Parameter," *Comm. Pure Appl. Math.*, XXXI 133-156 (1978).
- [12] Snider, A. D. and Fleming, G. C.: "Approximation by Aliasing with Applications to 'Certain' Stiff Differential Equations," *Math. Comp.*, 28, 465-473 (1974).

RICCATI TYPE TRANSFORMATIONS AND DECOUPLING

OF SINGULARITY PERTURBED ODE

R.M.M. MATTHEIJ

*Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, New York, 12181*

*on leave from
Mathematisch Instituut
Katholieke Universiteit
Toernooiveld
Nijmegen, The Netherlands*

A B S T R A C T

We consider the linear ODE

$$\epsilon \frac{dx}{dt} = A^{11}x^1 + A^{12}x^2$$

$$\frac{dx}{dt} = A^{21}x^1 + A^{22}x^2$$

For various reasons one may be interested in a "similarly" transformed ODE that is decoupled such that one has a possibility to compute the fast and slow modes and (or the decreasing and increasing modes) separately. We propose a technique that employs a Riccati transformation for decoupling the two time scales and a Liapunov type of transformation to decouple increasing and decreasing

fast modes. The last technique is numerically implemented as a special predictor-corrector technique, which employs a QU-decomposition each corrector step.

1.- PROBLEMSETTING

Consider the linear ODE

$$(1.1) \quad \begin{aligned} \epsilon \frac{dx^1}{dt} &= A^{11}x^1 + A^{12}x^2 + r^1 \\ \frac{dx^2}{dt} &= A^{21}x^1 + A^{22}x^2 + r^2, \quad t \geq 0 \end{aligned}$$

where A^{11} , A^{12} , A^{21} and A^{22} are matrix functions of t and r^1, r^2 are vector functions of t . We assume that A^{22} is nonsingular for all t . In fact for our discussion only the homogeneous part of (1.1) is of importance, as will turn out later. Problems with slow and fast time scales (as expressed by the "small" parameter ϵ) arise both in IVP and BVP. We restrict ourselves to the latter class of problems. They differ from IVP in that they usually deal with ODE that have increasing and decreasing modes. Due to the instability with respect to initial values of the former, serious numerical difficulties may arise if one tries to compute them (as may be necessary to obtain a fundamental solution in solving the BVP). However, for the same reason as the increasing modes make forward

integration an unstable affair, the decreasing modes render backward integration unstable. Therefore we look for methods that decouple the differential equation into slow and fast modes on the one hand (in order to localize the stiffness) and in increasing and decreasing modes on the other hand (in order to circumvent the above mentioned instability). We shall do this by transforming the homogeneous part of (1.1) onto (block) uppertriangular form, using linear transformations $T(t)$, i.e. we determine U^{11} , U^{21} and U^{22} , such that by setting

$$(1.2) \quad \begin{pmatrix} \epsilon x^1 \\ x^2 \end{pmatrix} = T \begin{pmatrix} \epsilon y^1 \\ y^2 \end{pmatrix}$$

we have

$$(1.3) \quad \begin{aligned} \text{a) } \epsilon \frac{dy^1}{dt} &= U^{11} y^1 + U^{12} y^2 + (T^{-1}r)^1 \\ \text{b) } \frac{dy^2}{dt} &= U^{22} y^2 + (T^{-1}r)^2 \end{aligned}$$

The ODE (1.3) (b) has smooth solutions. Hence one may e.g. apply a multiple shooting technique for determining a fundamental solution and a particular solution of it. Once the component y^2 of a solution has been determined we are left with an (inhomogeneous) stiff ODE for y^1 . The U^{11} now has an upper triangular form,

$$(1.4) \quad U^{11} = \begin{pmatrix} B & C \\ \phi & E \end{pmatrix}, \quad B \text{ of order } m \text{ say,}$$

such that B belongs to a system describing the increasing modes and E similarly the decreasing modes. If we partition the vector y^1 correspondingly as $\begin{pmatrix} z^1 \\ z^2 \end{pmatrix}$, then z^2 should be computed in forward direction, for which we may use a stiff integrator, and similarly z^1 in backward direction afterwards. In Section 2 we first describe the decoupling of slow and fast time scales. In Section 3 we then show how the fast system may be decoupled further. The latter decoupling is done by a predictor-corrector method, the convergence of which is shown in Section 4 and its stability in Section 5.

2.- DECOUPLING OF SLOW AND FAST MODES

For the matrix T we choose the (generalized) Riccati transformation

$$(2.1) \quad T(t) = \begin{pmatrix} M(t) & \phi \\ P(t)M(t) & I \end{pmatrix}, \quad M(t) \text{ nonsingular.}$$

we obtain then

$$(2.2) \begin{pmatrix} U^{11} & U^{12} \\ \phi & U^{22} \end{pmatrix} = \begin{pmatrix} M^{-1} [(A^{11} + A^{12}P) - \epsilon \dot{M}] & M^{-1} A^{12} \\ \phi & -PA^{12} + A^{21} \end{pmatrix}$$

provided P satisfies the Riccati equation

$$(2.3) \quad \epsilon \dot{P} = PA^{11} + PA^{12}P - \epsilon A^{22}P - \epsilon A^{21}$$

It is of importance now to realize that we are free to choose the initial value $P(0)$ of (2.3). In particular we may try to choose $P(0)$ such that P is smooth. From power series expansions arguments (see e.g. Mattheij, O'Malley, this proceedings) we may therefore expect $P(0)=0$ to be a good choice. Rather than computing P by some stiff integrator, we suggest to compute a few terms of its power expansion in ϵ . Let

$$(2.4) \quad P(t) = \sum_{j \geq 0} \epsilon^j P_j(t)$$

then we have e.g.

$$(2.5) \quad 0 = -A^{22}P_0 - A^{21},$$

whence

$$(2.6) \quad P_0 = -[A^{22}]^{-1}A^{21}$$

And also for the first order term

$$(2.7) \quad \dot{P}_0 = P_0 A^{11} + P_0 A^{12} P_0 - A^{22} P_1$$

Since \dot{P}_0 can be found analytically, i.e.

$$(2.8) \quad \dot{P}_0 = [A^{22}]^{-1} \{ \dot{A}^{22} [A^{22}]^{-1} A^{21} - \dot{A}^{21} \},$$

P_1 follows from (2.7) and (2.8) etc.

The attractive feature of this approach is that it can be carried out without introducing discretization errors and therefore is very suited to be combined with a numerical method that requires certain stepsizes in order to compute e.g. solutions of (1.3) (a).

3.- FURTHER REDUCTION OF THE FAST SYSTEM

Thus far we didn't specify the matrix M . We now require U^{11} to have an upper triangular form and therefore we have to determine M such that

$$(3.1) \quad \epsilon \dot{M} = FM - MU, \text{ where } F = A^{11} - A^{12}P, U = U^{11}$$

If we assume that the directions of the solutions of the homogeneous part of (1.1) are only slowly varying, we may expect that (3.1) has a smooth solution M , such that $M(t)$ is orthogonal for all t . To understand this one should realize that e.g. the first column of M represents the "direction" of some solution of the ODE $\dot{y}^1 = Fy^1$, and

the first two columns "span" a two dimensional solution subspace etc. In order to find such M and U we propose a predictor-corrector technique, where the corrector utilizes a kind of QR algorithm that ultimately gives the orthogonal matrix M and the upper triangular matrix U. We shall only sketch the method. Suppose we have a corrector

$$(3.2) \quad \sum_{j=0}^n \alpha_j z^j ; \quad \sum_{j=0}^n \beta_j z^j, \quad \alpha_0 = 1, \beta_0 \neq 0,$$

then a typical corrector step has the form

$$(3.3) \quad M_i^j U_i^j =: F_i M_i^{j-1} - \frac{\epsilon}{h \beta_0} M_i^{j-1} + G_{i-1}$$

In (3.3) we have set $F_i = F(t_i)$, $M_i^j =$ the j th correction step for $M(t_i)$ and similarly U_i^j ; finally G_{i-1} contains the rest of the terms, only depending on previous time steps. In the next section we deal with the convergence of the iteration (3.3) and in Section 5 with the accuracy of the approximation M_i^∞

4.- CONVERGENCE OF THE CORRECTOR METHOD

We write more conveniently the iteration (3.3) as

$$(4.1) \quad M^j U^j = H M^{j-1} + G,$$

i.e., we omit the indices and write for short

$$(4.2) \quad H = \left(F - \frac{\epsilon}{h \beta_0} j \right)$$

We shall restrict ourselves to matrices F having a complete system of eigenvectors e_1, \dots, e_n , corresponding to the (ordered) eigenvalues $\lambda_1, \dots, \lambda_n$ ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$). Moreover we shall only investigate the convergence of the first column of M^j , m^j say. Let $G(1)$ denote the first column of G , then there exist scalars $\gamma_1, \dots, \gamma_n$ such that

$$(4.3) \quad G(1) = \sum_{j=1}^n \gamma_j e_j$$

It is not restrictive to assume that all $\gamma_j \leq 0$. We now first investigate the fixed points (if any) of the iteration (of (4.1))

$$(4.4) \quad m^j v^j = H m^{j-1} + G(1), \quad \|m^j\|_2 = \|m^{j-1}\|_2 = 1$$

Hence suppose for some vector m with $\|m\|_2 = 1$ and some scalar v there holds

$$(4.5) \quad m v = H m + G(1)$$

Note that (4.5) might be regarded as an "inhomogeneous eigenvalue problem".

Now write

$$(4.6) \quad m = \sum_{\ell=1}^n \mu_\ell e_\ell,$$

then we obviously have

$$(4.7) \quad (v - \lambda_\ell) \mu_\ell = \gamma_\ell, \quad \ell = 1, \dots, n$$

Hence

Property 4.8 If $\gamma_\ell \neq 0$ for any ℓ , then $\mu_\ell \neq 0$.

Now let k be such that $\gamma_k \neq 0$, then we obtain from (4.7)

$$(4.9) \quad \mu_\ell = \mu_k \frac{\gamma_\ell}{\gamma_k + (\lambda_k - \lambda_\ell) \mu_k}$$

The actual values for μ_k (and hence for the other μ_ℓ) follow from the requirement $\|m\|_2 = 1$ if we write for short

$$(4.10) \quad e_{ij} = (e_i, e_j)$$

(the natural inner product), then we must require that

$$(4.11) \quad f_k(\mu_k) = 1,$$

where

$$(4.12) \quad f_k(x) = x^2 \sum_{i,j=1}^n \frac{e_{ij} \gamma_i \gamma_j}{(\gamma_k + (\lambda_k - \lambda_j)x)(\gamma_k + (\lambda_k - \lambda_i)x)}$$

It is not restrictive to take k minimal. Since by definition f_k is positive definite ($x \neq 0$) we have a graph like in fig 4.1

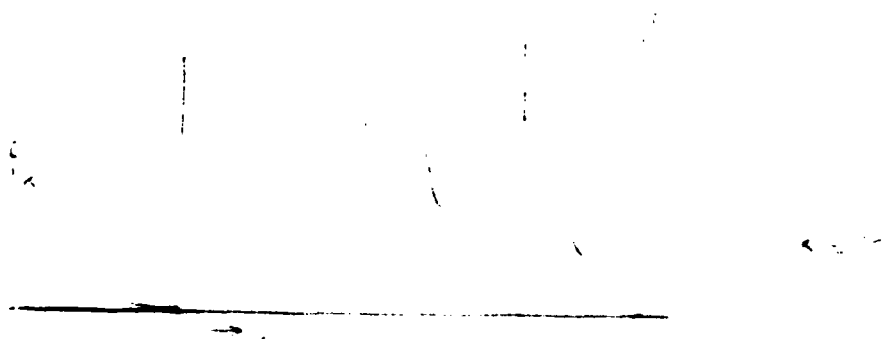


Fig. 4.1

Hence there are at least two values of x such that $f_k(x)=1$. Denote the negative root of $f_k(x)=1$ by α_k . We have then

Theorem 4.1 The choice $\mu_k = \alpha_k (< 0)$ is the only stable one and gives a fixed point m of (4.5) with the largest possible $v = \lambda_k + \frac{\gamma_k}{\mu_k}$

That we should choose $\mu_k < 0$ for obtaining the largest possible v is trivial (n.b. $\gamma_k < 0!$). The stability follows from a perturbation argument.

CONCLUSION

The sequence $\{m^j\}$ converges to a unique vector m and the sequence $\{v^j\}$ converges at the same time to a scalar v , and they moreover form the largest possible pair (m, v) .

Now if $m(t_{i-1})$ was the approximate direction of the most increasing solution at time $t=t_{i-1}$, then it can be expected that $m(t_i)$ computed this way, will approximately have the direction of this solution at $t=t_i$. Indeed, to see this, let for simplicity F be a constant matrix, and $m(t_i) = e_1 / \|e_1\|_2$. Then apparently $G_{i-1}(1)$ has the direction

of e_1 , whereas $\gamma_1 \approx \epsilon/h\beta_0$ and $\gamma_2, \dots, \gamma_n$ are small.

From (4.11), with $k=1$, we thus see

$$(4.14) \quad 1 \approx \mu_1^2,$$

$$\text{whence } v \approx \gamma_1 + \frac{\epsilon}{h\beta_0}$$

Since λ_1 is the largest eigenvalue of $H(=F - \frac{\epsilon}{h\beta_0} I)$ we therefore see that v is approximately the largest eigenvalue of F !

By similar arguments as are used to show convergence of the QR algorithm one can expect that the general process (4.1) will converge in a stable way to some M and U .

In order to find initial values for M and U , one may use an appropriately ordered Schur normal form of $F(0)$.

5.- ACCURACY OF THE DECOUPLED DIFFERENCE EQUATION

Since the above described method necessarily introduces discretization errors, the actually found upper triangular matrices U_i will, in general, not correspond to an exactly transformed ODE. The following qualitative discussion is concerned with this problem.

Let $M(t)$ be a piecewise polynomial that is sufficiently differentiable (at least as many times as the order of the corrector) and moreover such that

$$(5.1) \quad M(t_i) := M_i$$

For this M , which is a certain transformation function now, define a matrix function $\hat{U}(t)$ by

$$(5.2) \quad \hat{U} = M^{-1}(AM - \epsilon \dot{M})$$

If we (theoretically) discretize (5.2) we make a discretization error $h\Delta_i$ say. Apparently, using the notation $\hat{U}_i := \hat{U}(t_i)$, we must then have

$$(5.3) \quad 0 = h \sum_{j=0}^n \beta_j M_i (U_i - \hat{U}_i) + h\Delta_i$$

(for simplicity we assume constant step sizes). Let the polynomial $\sum_{j=0}^n \beta_j z^j$ have stable roots only, then it follows from a standard error analysis that

$$(5.4) \quad \|U_i - \hat{U}_i\| = O\left(\sum_{\ell=1}^i (\|\Delta_\ell\|/h)\right),$$

i.e. of the order (h^{p-1}) for a p th order method. (Note that M_i is orthogonal).

Whether or not these errors in U_i are small clearly depends on the values of the magnitudes of the elements of U_i , which can be monitored. Moreover, we can monitor the local errors quite conveniently by taking $(p+1)$ th divided differences of the matrices M_i .

DECOUPLING OF BOUNDARY VALUE PROBLEMS
FOR TWO-TIME-SCALE SYSTEMS

by

R. M. M. Mattheij¹

and

R. E. O'Malley, Jr.²

Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, New York 12181

1. Supported in part by the Netherlands Organization for the Advancement of Pure Research (Z.W.O.) and the Niels Stenson Stichting. On leave from the Mathematisch Instituut, Katholieke Universiteit, Nijmegen, The Netherlands
2. Supported in part by the Office of Naval Research under Contract Number N00014-81K-056.

1. Introduction

We wish to consider linear two-point boundary value problems of the form

$$(1) \quad \frac{dx}{dt} = a(t)x + f(t), \quad 0 \leq t \leq 1$$

$$(2) \quad M_0 x(0) + M_1 x(1) = b$$

where the homogeneous system involves coupled slow and fast dynamics. This contrasts sharply with most mathematical discussions of singularly perturbed systems where a small parameter multiplies some of the derivatives and slow and (potentially) fast components of the solution vector are immediately recognized. Special numerical methods are available for stiff initial value problems with slow and fast components identified (cf. Söderlind (1981)) and Riccati-like transformations are available to decouple such problems analytically and/or numerically (cf., e.g., O'Malley (1969), Mattheij (1979), and O'Malley and Anderson (1982)). In this paper, we propose a different type of transformation which will be useful in more general situations. We expect to numerically implement our algorithm and to pursue important generalizations in several directions. Relation to ongoing work of past and present colleagues and friends will be obvious to those familiar with earlier work (cf., especially, Ferguson (1975), Flaherty and O'Malley (1980), Mattheij and Staarink (1980), and Chow et al. (1981)). They are all thanked for their stimulation and encouragement.

Our basic assumption is that the homogeneous system has both smooth (or "slow") and rapidly-varying (or "fast") solution modes. This requires the matrix $a(t)$ to have large entries, so to make

our hypotheses more explicit, we introduce a small positive scaling parameter ϵ and write

$$(3) \quad a(t) = \frac{1}{\epsilon} A(t, \epsilon) \sim \frac{1}{\epsilon} \sum_{j=0}^{\infty} A_j(t) \epsilon^j \quad .$$

We do not actually intend to use a full power series expansion here (or below), but would in practice use a truncated expansion (approximation) involving only a few terms. Thus, our n -vector system takes the form

$$(4) \quad \epsilon \frac{dx}{dt} = A(t, \epsilon)x + \epsilon f(t)$$

where the presence of slow modes implies that $A(t, 0) = A_0(t)$ is a singular matrix. Since the nonhomogeneity can be dealt with by variation of parameters, most of our attention will be restricted to the homogeneous problem.

We shall seek a change of variables

$$(5) \quad x = T(t, \epsilon)y$$

where T and T^{-1} are both smooth (slowly-varying) throughout the interval $0 \leq t \leq 1$. (Appropriate assumptions will be introduced as our discussion proceeds.) The transformed problem will take the form

$$(6) \quad \epsilon \frac{dy}{dt} = U(t, \epsilon)y + \epsilon T^{-1}f$$

where

$$(7) \quad U = T^{-1} \left(AT - \epsilon \frac{dT}{dt} \right) \quad .$$

We shall determine T and U simultaneously so that U has the block triangular form

$$(8) \quad U = \begin{pmatrix} U^{11} & U^{12} & U^{13} \\ 0 & U^{22} & U^{23} \\ 0 & 0 & \epsilon U^{33} \end{pmatrix}$$

where the eigenvalues of U^{11} and $-U^{22}$ all have strictly positive real parts (at least for ϵ small). Thus, the rapidly-varying solution modes of the transformed homogeneous problem will be either fast-growing or fast-decaying, but not (indefinitely) rapidly oscillatory. We shall assume that the partitioning holds throughout the interval $0 \leq t \leq 1$, thereby implicitly eliminating possible "turning points" within the interval (cf. Wasow (1965)). The matrix T , then, corresponds to a decoupling transformation which separates rapidly growing, rapidly decaying, and slow modes. The time scales for the rapid modes will be roughly $O(1/\epsilon)$ times those for the slow modes. The reason we don't allow purely imaginary eigenvalues of $U(t,0)$ is that we wish to base our analysis on boundary layer methods (cf. O'Malley (1974)). We shall further presume that we can obtain smooth dependence of the U^{ij} 's on ϵ , although dependence on some root $\epsilon^{1/r}$, for some integer $r > 1$, would be necessary in some degenerate situations involving defective eigenvalues. These neglected possibilities should be considered in later work.

In order to proceed, let us fix the dimensions of U^{11} and U^{22} to be $k \times k$ and $l \times l$, respectively, and write T as

$$(9) \quad T(t, \epsilon) = T(t, 0) [I_n + \epsilon G(t, \epsilon)]$$

where G will ultimately have the special, compatibly partitioned form

$$(10) \quad G(t, \epsilon) = \begin{pmatrix} 0 & G^{12} & 0 \\ G^{21} & 0 & 0 \\ G^{31} & G^{32} & 0 \end{pmatrix}$$

$$\sim \sum_{j=0}^{\infty} (G_j^{\alpha\beta}(t)) \epsilon^j .$$

At several stages, we make somewhat arbitrary choices for the form of our transformations. Though these cause us no complications, others might find different choices more appropriate. Our decoupling will be done in two steps. First, we shall separate fast and slow time scales. Then, we split fast-growing and fast-decaying modes to higher orders in ϵ . We shall let $m = n - k - l$ designate the number of slow modes with bounded derivatives as $\epsilon \rightarrow 0$.

2. Time-scale Decoupling

At our first decoupling stage, it is convenient to temporarily combine notation somewhat. Splitting T after its first $k + l$ columns, we set

$$(11) \quad T(t, \epsilon) = (R(t, \epsilon) \quad Z(t, \epsilon)) .$$

Introducing

$$(12) \quad B(t, \epsilon) = \begin{pmatrix} U^{11} & U^{12} \\ 0 & U^{22} \end{pmatrix}$$

and

$$(13) \quad C(t, \epsilon) = \begin{pmatrix} U^{13} \\ U^{23} \end{pmatrix} ,$$

the equation (7) defining U implies that the fast decoupling transformation $R(t, \epsilon)$ must satisfy the linear matrix (Liapunov) equation

$$(14) \quad \epsilon \dot{R} = AR - RB .$$

Since A and B have power series expansions in ϵ , it is natural to seek a smooth matrix solution R in the form

$$(15) \quad R(t, \epsilon) \sim \sum_{j=0}^{\infty} R_j(t) \epsilon^j .$$

We shall, indeed, simultaneously determine a formal series solution for this differential equation and for the block-triangular system matrix $B(t, \epsilon)$ for the transformed fast subsystem. Specifically, by equating coefficients we obtain

$$(16) \quad A_0 R_0 - R_0 B_0 = 0$$

and, for each $j \geq 1$,

$$(17) \quad A_0 R_j - R_j B_0 = R_0 B_j + \alpha_{j-1}$$

where the α_{j-1} 's are known in terms of preceding coefficients.

To solve our problem, we introduce the Schur canonical form

$$(18) \quad A_0 = Q_0 V_0 Q_0^T$$

for the principal part $A_0(t)$ of our system matrix $\epsilon a(t)$. Here, $Q_0(t)$ is orthogonal and $V_0(t)$ is upper triangular, so

$$(19) \quad A_0 Q_0 = Q_0 V_0 .$$

We observe that these matrices are convenient to obtain numerically through the QR algorithm (cf. Golub and Wilkinson (1976)). This provides a full triangularization for A_0 , though a block-triangularization would suffice for much of our paper. We shall set

$$(20) \quad v_0 = \begin{pmatrix} B_0 & C_0 \\ 0 & 0 \end{pmatrix}$$

and, without loss of generality, we suppose that the first k (and remaining l) eigenvalues of the $(k + l) \times (k + l)$ matrix B_0 will have strictly positive (negative) real parts throughout the interval $0 \leq t \leq 1$. This clearly, but definitely, restricts the matrices $A_0(t)$ under consideration. The matrices $B_0(t)$ and $C_0(t)$ so specified determine our limiting transformed system matrix $U(t, 0)$. Moreover, with such an ordering of the eigenvalues of B_0 , we separate the fast growing (first k) and fast decaying (next l) modes at this first (and critical) order (with respect to ϵ). If we further let R_0 coincide with the first $k + l$ columns of Q_0 , we satisfy the desired linear system $A_0 R_0 - R_0 B_0 = 0$ and specify the fast part $R_0(t)$ of our limiting transformation matrix $T(t, 0)$. Its rank, $k + l$, coincides with that of $A_0(t)$.

3. Higher-order Terms

Since $Q_0^T R_0 = \begin{pmatrix} I_{k+l} \\ 0 \end{pmatrix}$, multiplying the linear system (17) through by Q_0^T and setting

$$(21) \quad R_j = Q_0 \begin{pmatrix} \tilde{R}_j^1 \\ \tilde{R}_j^2 \end{pmatrix} \quad \text{for } j \geq 1$$

implies the pair of equations

$$(22) \quad B_0 \tilde{R}_j^1 + C_0 \tilde{R}_j^2 - \tilde{R}_j^1 B_0 = B_j + \alpha_{j-1}^1$$

and

$$(23) \quad -\tilde{R}_j^2 B_0 = \alpha_{j-1}^2$$

where the α_{j-1}^i 's are known successively.

Since the triangular matrix B_0 is simply-inverted by back substitution, \tilde{R}_j^2 is uniquely determined and a linear system

$$(24) \quad B_0 \tilde{R}_j^1 - \tilde{R}_j^1 B_0 = B_j + \beta_{j-1}$$

with $\beta_{j-1} = \alpha_{j-1}^1 + C_0 \tilde{R}_j^2$ remains. We now seek to specify \tilde{R}_j^1 so that the resulting fast-growing and -decaying modes are separated (to the appropriate order) by making B_j upper triangular. Substituting for B_0 and B_j , we seek a solution of the form

$$(25) \quad \tilde{R}_j^1(t) = \begin{pmatrix} 0 & G_j^{12} \\ G_j^{21} & 0 \end{pmatrix}.$$

This provides the four linear matrix equations

$$(26) \quad \begin{cases} U_0^{11} G_j^{12} - G_j^{12} U_0^{22} = \beta_{j-1}^{12} \\ U_0^{22} G_j^{21} - G_j^{21} U_0^{11} = \beta_{j-1}^{21} \\ U_0^{12} G_j^{21} = U_j^{11} + \beta_{j-1}^{11} \\ -G_j^{21} U_0^{12} = U_j^{22} + \beta_{j-1}^{22} \end{cases}$$

presuming we can take $U_j^{12} = 0$ for $j > 0$. The first two systems

are uniquely solvable for G_j^{12} and G_j^{21} , respectively, since U_0^{11} and U_0^{22} have no eigenvalues in common. Indeed, the triangular form of U_0^{11} and U_0^{22} allows a simple solution for these nonzero blocks of \tilde{R}_j^1 by forward and backward substitution. Then, the last two equations directly specify the block-diagonal matrix B_j . The procedure can be continued to any desired order.

To determine the full transformation $T(t, \epsilon)$, we must select a matrix $Z(t, \epsilon)$ whose span is complementary to that of $R(t, \epsilon)$. It is natural to attempt to take $T(t, 0) = Q_0(t)$. Then

$$(27) \quad \tilde{R}^1(t, \epsilon) = \begin{pmatrix} 0 & G^{12} \\ G^{21} & 0 \end{pmatrix} \text{ and } \tilde{R}^2(t, \epsilon) = (G^{31} \ G^{32}).$$

The T matrix so obtained is readily inverted since

$$(28) \quad T = Q_0 \begin{pmatrix} I_{k+l} + \epsilon \tilde{R}^1 & 0 \\ \epsilon \tilde{R}^2 & I_m \end{pmatrix}$$

implies that

$$(29) \quad T^{-1} = \begin{pmatrix} (I_{k+l} + \epsilon \tilde{R}^1)^{-1} & 0 \\ -\epsilon \tilde{R}^2 & I_m \end{pmatrix} Q_0^T$$

and

$$(30) \quad (I_{k+l} + \epsilon \tilde{R}^1)^{-1} = \begin{pmatrix} I_k & -\epsilon G_{12} \\ -\epsilon G_{21} & I_l \end{pmatrix} \begin{pmatrix} (I_k - \epsilon^2 G_{12} G_{21})^{-1} & 0 \\ 0 & (I_l - \epsilon^2 G_{21} G_{12}) \end{pmatrix}$$

at least for ϵ sufficiently small.

The matrices T and U are, of course, determined simultaneously. We should check that our transformed state matrix U attains the desired block structure. From (7), it follows that Z , the last m columns of T , must satisfy

$$(31) \quad RC + \epsilon ZU^{33} = AZ - \epsilon \frac{dZ}{dt} .$$

Since $Z = Q_0 \begin{pmatrix} 0 \\ I_m \end{pmatrix}$, multiplication by Q_0^T implies that

$$(32) \quad \epsilon Q_0^T \frac{dQ_0}{dt} \begin{pmatrix} 0 \\ I_m \end{pmatrix} + \epsilon \begin{pmatrix} 0 \\ U^{33} \end{pmatrix} = Q_0^T A Q_0 \begin{pmatrix} 0 \\ I_m \end{pmatrix} - Q_0^T R C .$$

Since $Q_0^T A Q_0 = \begin{pmatrix} B_0 & C_0 \\ 0 & 0 \end{pmatrix}$ and $Q_0^T R = \begin{pmatrix} I_{k+l} \\ 0 \end{pmatrix}$, we have

equality when $\epsilon = 0$. From the coefficient of each ϵ^j with $j > 0$, we directly obtain

$$(33) \quad C_j = \begin{pmatrix} U_j^{13} \\ U_j^{23} \end{pmatrix}$$

(in terms of preceding terms) from the first $k + l$ rows of (32) and

$$(34) \quad U_j^{33}$$

from the last m rows. This completes the specification of $T_j(t)$ and $U_j(t)$.

We should verify that the truncated series actually provides decoupling to a corresponding order and that analytic transformations exist with asymptotic expansions coinciding with the formal series generated. It should, however, be realized that our approach is not necessarily limited to asymptotic validity as $\epsilon \rightarrow 0$, but that the decoupling achieved should be numerically valid for any realistic small ϵ for which our algorithm is defined.

4. Boundary Value Problems

The principal value to our approach is the simplicity it provides for solving two-point boundary value problems. In y -coordinates, our problem (1) - (2) becomes

$$(35) \quad \epsilon \frac{dy}{dt} = U(t, \epsilon)y + \epsilon T^{-1}(t, \epsilon)f(t)$$

$$(36) \quad M_0 T(0, \epsilon)y(0) + M_1 T(1, \epsilon)y(1) = b .$$

If we first find a particular bounded solution of (35), using boundary conditions appropriate to the conditional stability of our fast subsystem, we need after only consider the homogeneous system.

All solutions of the homogeneous system have the form

$$(37) \quad y(t) = \Omega(t, \epsilon)c$$

where Ω is any fundamental matrix for $\epsilon \frac{d\Omega}{dt} = U(t, \epsilon)\Omega$. We shall

use a fundamental matrix

$$(38) \quad \Omega(t, \epsilon) = \begin{pmatrix} \Omega^{11} & \Omega^{12} & \Omega^{13} \\ 0 & \Omega^{22} & \Omega^{23} \\ 0 & 0 & \Omega^{33} \end{pmatrix}$$

subject to special boundary conditions at the endpoints $t = 0$ and $t = 1$ which will assure us boundedness of the $\Omega^{\alpha\beta}$'s. Specifically, we'll define Ω on $0 \leq t \leq 1$ by asking that

$$(39) \quad \left\{ \begin{array}{l} \frac{d\Omega^{33}}{dt} = U^{33}\Omega^{33}, \quad \Omega^{33}(0) = I_m \\ \epsilon \frac{d\Omega^{22}}{dt} = U^{22}\Omega^{22}, \quad \Omega^{22}(0) = I_k \\ \epsilon \frac{d\Omega^{23}}{dt} = U^{22}\Omega^{23} + U^{23}\Omega^{33}, \quad \Omega^{23}(0) = 0 \\ \epsilon \frac{d\Omega^{11}}{dt} = U^{11}\Omega^{11}, \quad \Omega^{11}(1) = I_k \\ \epsilon \frac{d\Omega^{12}}{dt} = U^{11}\Omega^{12} + U^{12}\Omega^{22}, \quad \Omega^{12}(1) = 0 \\ \epsilon \frac{d\Omega^{13}}{dt} = U^{11}\Omega^{13} + U^{12}\Omega^{23} + U^{13}\Omega^{33}, \quad \Omega^{13}(1) = 0 \end{array} \right.$$

As $\epsilon \rightarrow 0$, we note that Ω^{22} decays rapidly to zero for $t > 0$ (i.e., away from an initial boundary layer), while Ω^{11} rapidly decays to zero away from a terminal boundary layer near $t = 1$. Away from $t = 0$, Ω^{23} tends to an outer limit $(U^{22})^{-1}U^{23}\Omega^{33}$ as $\epsilon \rightarrow 0$. Ω_{12} and Ω_{13} involve boundary layer behavior at $t = 1$. Such asymptotic results readily follow from singular perturbations theory (cf. O'Malley (1974)).

Since any solution of our homogeneous problem (1) - (2) has the form (37), we obtain a unique solution if and only if the matrix

$$(40) \quad D \equiv M_0 T(0, \epsilon) \Omega(0, \epsilon) + M_1 T(1, \epsilon) \Omega(1, \epsilon)$$

is nonsingular. Moreover, the special structure of $T(t, \epsilon)$ and of

$\Omega(0,\epsilon)$ and $\Omega(1,\epsilon)$ can be utilized to simplify the checking of this invertibility condition. If $D = O(\epsilon^p)$ as $\epsilon \rightarrow 0$, we can anticipate having solutions which become algebraically unbounded like ϵ^{-p} . We can even identify the limiting behavior as $\epsilon \rightarrow 0$ within $(0,1)$ with the solution of a reduced boundary value problem of order m (cf. O'Malley (1969) and Harris (1973)). Special numerical interest, however, concerns problems where ϵ is small, but nonvanishing.

References

1. J. H. Chow, B. Avramovic, P. V. Kokotovic, and J. R. Winkelman (1981), "Singular Perturbations, Coherency and Aggregation of Dynamic Systems," Report, Electric Utility Systems Engineering Department, General Electric Company, Schenectady.
2. W. E. Ferguson, Jr. (1975), "A Singularly Perturbed Linear Two-Point Boundary Value Problem," doctoral dissertation, California Institute of Technology, Pasadena.
3. J. E. Flaherty and R. E. O'Malley, Jr. (1980), "On the numerical integration of two-point boundary value problems for stiff systems of ordinary differential equations," Boundary and Interior Layers - Computational and Asymptotic Methods, J. J. H. Miller, editor, Boole Press, Dublin, 93-102.
4. G. H. Golub and J. H. Wilkinson (1976), "Ill-conditioned eigensystems and the computation of the Jordan canonical form," SIAM Review 18, 578-619.
5. W. A. Harris, Jr. (1973), "Singularly perturbed boundary value problems revisited," Lecture Notes in Math. 312, Springer-Verlag, Berlin, 54-64.
6. R. M. M. Mattheij (1979), "On approximating smooth solutions of linear singularly perturbed ODE," Numerical Analysis of Singular Perturbation Problems, P. W. Hemker and J. J. H. Miller, editors, Academic Press, London, 457-465.
7. R. M. M. Mattheij and G. W. M. Staavink (1980), "A Method for Solving General Linear Boundary Value Problems," Report 8029, Mathematisch Instituut, Katholieke Universiteit, Nijmegen.

subject to special boundary conditions at the endpoints $t = 0$ and $t = 1$ which will assure us boundedness of the $\Omega^{\alpha\beta}$'s. Specifically, we'll define Ω on $0 \leq t \leq 1$ by asking that

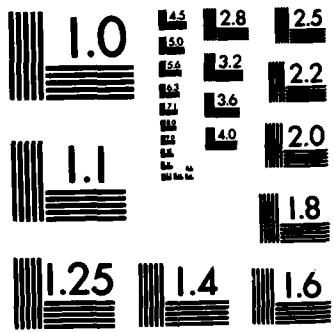
$$(39) \quad \left\{ \begin{array}{l} \frac{d\Omega^{33}}{dt} = U^{33}\Omega^{33}, \quad \Omega^{33}(0) = I_m \\ \epsilon \frac{d\Omega^{22}}{dt} = U^{22}\Omega^{22}, \quad \Omega^{22}(0) = I_\ell \\ \epsilon \frac{d\Omega^{23}}{dt} = U^{22}\Omega^{23} + U^{23}\Omega^{33}, \quad \Omega^{23}(0) = 0 \\ \epsilon \frac{d\Omega^{11}}{dt} = U^{11}\Omega^{11}, \quad \Omega^{11}(1) = I_k \\ \epsilon \frac{d\Omega^{12}}{dt} = U^{11}\Omega^{12} + U^{12}\Omega^{22}, \quad \Omega^{12}(1) = 0 \\ \epsilon \frac{d\Omega^{13}}{dt} = U^{11}\Omega^{13} + U^{12}\Omega^{23} + U^{13}\Omega^{33}, \quad \Omega^{13}(1) = 0 \end{array} \right.$$

As $\epsilon \rightarrow 0$, we note that Ω^{22} decays rapidly to zero for $t > 0$ (i.e., away from an initial boundary layer), while Ω^{11} rapidly decays to zero away from a terminal boundary layer near $t = 1$. Away from $t = 0$, Ω^{23} tends to an outer limit $(U^{22})^{-1}U^{23}\Omega^{33}$ as $\epsilon \rightarrow 0$. Ω_{12} and Ω_{13} involve boundary layer behavior at $t = 1$. Such asymptotic results readily follow from singular perturbations theory (cf. O'Malley (1974)).

Since any solution of our homogeneous problem (1) - (2) has the form (37), we obtain a unique solution if and only if the matrix

$$(40) \quad D \equiv M_0 T(0, \epsilon) \Omega(0, \epsilon) + M_1 T(1, \epsilon) \Omega(1, \epsilon)$$

is nonsingular. Moreover, the special structure of $T(t, \epsilon)$ and



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

$\Omega(0,\epsilon)$ and $\Omega(1,\epsilon)$ can be utilized to simplify the checking of this invertibility condition. If $D = O(\epsilon^P)$ as $\epsilon \rightarrow 0$, we can anticipate having solutions which become algebraically unbounded like ϵ^{-P} . We can even identify the limiting behavior as $\epsilon \rightarrow 0$ within $(0,1)$ with the solution of a reduced boundary value problem of order m (cf. O'Malley (1969) and Harris (1973)). Special numerical interest, however, concerns problems where ϵ is small, but nonvanishing.

References

1. J. H. Chow, B. Avramovic, P. V. Kokotovic, and J. R. Winkelman (1981), "Singular Perturbations, Coherency and Aggregation of Dynamic Systems," Report, Electric Utility Systems Engineering Department, General Electric Company, Schenectady.
2. W. E. Ferguson, Jr. (1975), "A Singularly Perturbed Linear Two-Point Boundary Value Problem," doctoral dissertation, California Institute of Technology, Pasadena.
3. J. E. Flaherty and R. E. O'Malley, Jr. (1980), "On the numerical integration of two-point boundary value problems for stiff systems of ordinary differential equations," Boundary and Interior Layers - Computational and Asymptotic Methods, J. J. H. Miller, editor, Boole Press, Dublin, 93-102.
4. G. H. Golub and J. H. Wilkinson (1976), "Ill-conditioned eigensystems and the computation of the Jordan canonical form," SIAM Review 18, 578-619.
5. W. A. Harris, Jr. (1973), "Singularly perturbed boundary value problems revisited," Lecture Notes in Math. 312, Springer-Verlag, Berlin, 54-64.
6. R. M. M. Mattheij (1979), "On approximating smooth solutions of linear singularly perturbed ODE," Numerical Analysis of Singular Perturbation Problems, P. W. Hemker and J. J. H. Miller, editors, Academic Press, London, 457-465.
7. R. M. M. Mattheij and G. W. M. Staarink (1980), "A Method for Solving General Linear Boundary Value Problems," Report 8029, Mathematisch Instituut, Katholieke Universiteit, Nijmegen.

8. R. E. O'Malley, Jr. (1969), "Boundary value problems for linear systems of ordinary differential equations involving many small parameters," J. Math. Mech. 18, 835-856.
9. R. E. O'Malley, Jr. (1974), Introduction to Singular Perturbations, Academic Press, New York.
10. R. E. O'Malley, Jr. and L. R. Anderson (1982), "Time-scale decoupling and order reduction for linear time-varying systems," Optimal Control Applications and Methods 3.
11. G. Söderlind (1981), "Theoretical and Computational Aspects of Partitioning in the Numerical Integration of Stiff Differential Systems," Report 8115, Department of Numerical Analysis and Computing Science, the Royal Institute of Technology, Stockholm.
12. W. Wasow (1965), Asymptotic Expansions for Ordinary Differential Equations, Wiley-Interscience, New York.

P-STABLE AND $P[\alpha, \beta]$ -STABLE INTEGRATION/INTERPOLATION METHODS
IN THE SOLUTION OF
RETARDED DIFFERENTIAL-DIFFERENCE EQUATIONS*

Theodore A. Bickart[†]

ABSTRACT The equation $\dot{u}(t) = pu(t) + qu(t-\tau)$ is presented as an archetype (scalar) equation for assessing the quality of integrator/interpolator pairs used to solve retarded differential-difference equations. P-stability and $P[\alpha, \beta]$ -stability, defined with respect to this archetype equation, are proposed as desirable properties of integrator/interpolator pairs. Relationships of these properties to passivity and order of multistep integrators and Lagrange interpolators are developed. Composite multistep integrators and composite Lagrange interpolators are considered as a means of obtaining high order pairs stable for all step-sizes over a large portion, if not all, of the (p, q) -domain on which the archetype equation is stable.

*The research reported herein was supported by grant MCS-7815396 from the National Science Foundation.

[†]Electrical and Computer Engineering Department, Syracuse University, Syracuse, New York 13210, USA.

INTRODUCTION

Consider the autonomous retarded differential-difference equation

$$\dot{y}(t) = f[y(t), y(t-\tau_1), \dots, y(t-\tau_m)] \quad (1a)$$

with the initial function

$$y(t) = \phi(t) \quad (t \in [-\tau, 0]), \quad (1b)$$

where $\tau = \max_{i \in \{1, \dots, m\}} \{\tau_i\}$. Suppose the function f is sufficiently differentiable; then we have at $(0, 0, \dots, 0)$ — chosen for notational convenience —

$$f(y_0, y_1, \dots, y_m) = f(0, 0, \dots, 0) + \sum_{i=1}^m F^i y_i + g(y_0, y_1, \dots, y_m), \quad (2)$$

where

$$F^i = \frac{f}{y_i}(0, 0, \dots, 0) \quad (3)$$

and

$$\lim_{\sum_{i=1}^m \|y_i\| \rightarrow 0} \|g(y_0, y_1, \dots, y_m)\| / \sum_{i=1}^m \|y_i\| = 0. \quad (4)$$

Assume that there exists an α such that

$$f(0, 0, \dots, 0) + [\sum_{i=1}^m F^i] \alpha = 0 \quad (5)$$

Now, set $y(t) = x(t) + \alpha$; then (1) with (2) and (5) yields the autonomous retarded differential-difference equation

$$\begin{aligned} \dot{x}(t) = & F^0 x(t) + \sum_{i=1}^m F^i x(t-\tau_i) \\ & + g[x(t)+\alpha, x(t-\tau_1)+\alpha, \dots, x(t-\tau_m)+\alpha] \end{aligned} \quad (6a)$$

with the initial function

$$\underline{x}(t) = \underline{\phi}(t) - \underline{a} \quad (t \in [-\tau, 0]) \quad (6b)$$

It is out of consideration of (6a) that we were led to consider

$$\underline{\dot{u}}(t) = \underline{F}^0 \underline{u}(t) + \sum_{i=1}^m \underline{F}^i \underline{u}(t-\tau_i) \quad (7a)$$

with the initial function

$$\underline{u}(t) = \underline{\psi}(t) \quad (t \in [-\tau, 0]) \quad (7b)$$

as an archetype equation against which to assess the stability properties of numerical methods for the solution of (1). In [1] we defined F-stability and $F[\alpha, \beta]$ -stability with respect to (7) for a multistep-method/Lagrange-method pair and displayed pairs exhibiting such stability types.

In this paper we turn to a special case, that when (7) is a scalar equation with just one delay. For this distinctive case we replace (7) by

$$\dot{u}(t) = pu(t) + qu(t-\tau) \quad (8a)$$

with

$$u(t) = \psi(t) \quad (t \in [-\tau, 0]). \quad (8b)$$

In [2], Barwell, extending results reported by Cryer in [3], proposed this as an archetype equation. He there introduced the P-stability and GP-stability types. These types differed only in a restriction on τ in the former type. Since we will not distinguish two types by such a restriction, we will need only one type designator. For simplicity, we shall adopt the former designator (P-stability), but with a definition to suit our exposition.

BACKGROUND

We shall say that the archetype equation (8a) is p-dominant if

$$\operatorname{Re}(p) < -|q|. \quad (9)$$

The asymptotic behavior of the solution of (8), when (8a) is p-dominant, is described in

Theorem 1: For all bounded, measurable $\psi(t)$, the solution $u(t)$ of (8) is asymptotic to zero ($\lim_{t \rightarrow \infty} u(t) = 0$) for all $\tau \geq 0$ if (8a) is p-dominant.

The validation of this result given by Barwell in [2] is more tedious than that found next in our

Proof: Let a variable subscripted with R [with I] denote the real part [the imaginary part] of that variable. Then the complex scalar-valued differential-difference equation (8a) is equivalent to the real vector-valued differential-difference equation

$$\begin{bmatrix} \dot{u}_R(t) \\ u_I(t) \end{bmatrix} = \begin{bmatrix} p_R & -p_I \\ p_I & p_R \end{bmatrix} \begin{bmatrix} u_R(t) \\ u_I(t) \end{bmatrix} + \begin{bmatrix} q_R & -q_I \\ q_I & q_R \end{bmatrix} \begin{bmatrix} u_R(t-\tau) \\ u_I(t-\tau) \end{bmatrix} \quad (10)$$

The solution of this equation will be asymptotic to zero for all $\tau \geq 0$ if and only if its characteristic function

$$\begin{aligned}
 P(s, e^{-\tau s}) &= \det \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} s - \begin{bmatrix} p_R & -p_I \\ p_I & p_R \end{bmatrix} - \begin{bmatrix} q_R & -q_I \\ q_I & q_R \end{bmatrix} e^{-\tau s} \right\} \\
 &= (s - p - q e^{-\tau s})(s - p^* - q^* e^{-\tau s})
 \end{aligned} \tag{11}$$

is nonzero in the closed right half-plane for all $\tau \geq 0$. [4,5]

Note: A superscript * denotes (complex) conjugate. As shown by Kamen in [6], this is equivalent to

$$P(s, z) \neq 0 \quad (\operatorname{Re}\{s\} \geq 0, |z| \leq 1), \tag{12a}$$

where

$$P(s, z) = (s - p - qz)(s - p^* - q^*z). \tag{12b}$$

Now, for $|z| \leq 1$,

$$\operatorname{Re}\{p + qz\} = \operatorname{Re}\{p\} + \operatorname{Re}\{qz\} \leq \operatorname{Re}\{p\} + |q| < 0$$

and

$$\operatorname{Re}\{p^* + q^*z\} = \operatorname{Re}\{p^*\} + \operatorname{Re}\{q^*z\} \leq \operatorname{Re}\{p^*\} + |q^*| = \operatorname{Re}\{p\} + |q| < 0.$$

These two relations imply, respectively, that $(s - p - qz)$ and $(s - p^* - q^*z)$ cannot become zero for $\operatorname{Re}\{s\} \geq 0$ when $|z| \leq 1$.

Therefore, $P(s, z)$ cannot become zero for $\operatorname{Re}\{s\} \geq 0$ when $|z| \leq 1$. ■

Next, let u_n , \dot{u}_n , and u_{n-v} denote approximations of $u(t_n)$, $\dot{u}(t_n)$ and $u(t_n - \tau)$, where $t_n = t_{n-1} + h$ and $v = \tau/h$ for some h . Clearly, (8a) with the substitution of these approximations provides just one relation among the three variables u_n , \dot{u}_n , and u_{n-v} . An additional two independent relations are needed.

As the first of these two remaining relations we shall use the multi-step (integration) relation

$$\sum_{i=-\kappa}^0 [\alpha_i u_{n+1} - h\beta_i \dot{u}_{n+1}] = 0, \quad (13)$$

which is of order \bar{p} if the coefficients α_i and β_i satisfy

$$\sum_{i=-\kappa}^0 \alpha_i = 0 \quad (14a)$$

and

$$\sum_{i=-\kappa}^0 [i^j \alpha_i - j i^{j-1} \beta_i] = 0 \quad (j=1, \dots, \bar{p}). \quad (14b)$$

Being of order \bar{p} implies, when $u(t)$ is sufficiently differentiable, that

$$\sum_{i=-\kappa}^0 [\alpha_i u(t_{n+1}) - h\beta_i \dot{u}(t_{n+1})] = \gamma_{\bar{p}+1}^{(\bar{p}+1)}(\mu) h^{\bar{p}+1} + O[h^{\bar{p}+2}], \quad (15)$$

where $\mu \in [t_{n-\kappa}, t_n]$ and

$$\gamma_{\bar{p}+1} = \sum_{i=-\kappa}^0 [i^{\bar{p}+1} \alpha_i - (\bar{p}+1) i^{\bar{p}} \beta_i]. \quad (16)$$

As the second of these two remaining relations we shall use the order \bar{q} Lagrange (interpolation) relation

$$u_{n-\nu} - \sum_{j=0}^{\bar{q}} \delta_j(\nu) u_{n+\sigma_j} = 0, \quad (17)$$

where the indices σ_j ($j=0, \dots, \bar{q}$) satisfy $0 \geq \sigma_{\bar{q}} > \dots > \sigma_0$ and the coefficients $\delta_j(\nu)$ satisfy

$$(-\nu)^k - \sum_{j=0}^{\bar{q}} \delta_j(\nu) \sigma_j^k = 0 \quad (k=0, \dots, \bar{q}). \quad (18)$$

The well known solution of (18) is

$$\delta_j(\nu) = \prod_{\substack{k=0 \\ k \neq j}}^{\bar{q}} \frac{\sigma_k + \nu}{\sigma_k - \sigma_j}. \quad (19)$$

Being of order \bar{q} implies, when $u(t)$ is sufficiently differentiable, that

$$\begin{aligned} u(t_n - \tau) - \sum_{j=0}^{\bar{q}} \delta_j(v) u(t_{n+\sigma_j}) \\ = \xi_{\bar{q}+1}^-(v) u^{(\bar{q}+1)}(\mu) h^{\bar{q}+1} + O[h^{\bar{q}+2}], \end{aligned} \quad (20)$$

where $\mu \in [t_{n+\sigma_0}, t_{n+\sigma_{\bar{q}}}]$ and

$$\xi_{\bar{q}+1}^-(v) = (-v)^{\bar{q}+1} - \sum_{j=0}^{\bar{q}} \delta_j(v) \sigma_j^{\bar{q}+1}. \quad (21)$$

We turn our attention now to the stability versus order properties of the numerical methods.

Firstly, with respect to the multistep-method, herein assumed to be such that $\sum_{i=-k}^0 \alpha_i \zeta^{k+1}$ and $\sum_{i=-k}^0 \beta_i \zeta^{k+1}$ are relatively prime, we have

Definition 1: The multistep-method is said to be passive if

$$\operatorname{Re}\left\{\frac{\sum_{i=-k}^0 \alpha_i \zeta^{k+1}}{\sum_{i=-k}^0 \beta_i \zeta^{k+1}}\right\} \geq 0 \quad \forall \zeta \in \{|\zeta| \geq 1 \wedge \sum_{i=-k}^0 \beta_i \zeta^{k+1} \neq 0\}. \quad (22)$$

and

Definition 2: The multistep-method is said to be α -passive

$(0 < \alpha \leq \pi/2)$ if

$$\left| \arg\left\{\frac{\sum_{i=-k}^0 \alpha_i \zeta^{k+1}}{\sum_{i=-k}^0 \beta_i \zeta^{k+1}}\right\} \right| \leq \pi - \alpha \quad \forall \zeta \in \{|\zeta| \geq 1 \wedge \sum_{i=-k}^0 \beta_i \zeta^{k+1} \neq 0\}. \quad (23)$$

Obviously, a $\pi/2$ -passive method is passive. Note: Passivity of a method is equivalent to it being A-stable and α -passivity of a method is equivalent to it being $A[\alpha]$ -stable. The research of Nevanlinna and Sipila [7] makes it evident that α -passive methods are necessarily implicit. Furthermore, by the now classic Dahlquist Theorem [8], we know that the order of passive methods cannot exceed two. Fortunately, α -passive methods (with $\alpha < \pi/2$) of order exceeding two exist; for example, the backward differentiation methods

of Gear [9, chapter 11] are a set of α -passive methods—orders one through six.

Secondly, with respect to the Lagrange method we have

Definition 3: The Lagrange-method is said to be passive if

$$\left| \sum_{j=0}^{\bar{q}} \delta_j(v) \zeta^{\sigma_j} \right| \leq 1 \quad \forall |\zeta| \geq 1. \quad (24)$$

and

Definition 4: The Lagrange-method is said to be β -passive

($1 \leq \beta$) if

$$\left| \sum_{j=0}^{\bar{q}} \delta_j(v) \zeta^{\sigma_j} \right| \leq \beta \quad \forall |\zeta| \geq 1. \quad (25)$$

Obviously, a 1-passive method is passive. The research of Strang [10] (See also, [11].) discloses that Lagrange-methods with a uniform mesh ($\sigma_j = -\lambda + j$) are passive when $-v \in [-\lambda + (\bar{q}-1)/2, -\lambda + (\bar{q}+1)/2]$ for q odd and $-v \in [-\lambda + \bar{q}/2 - 1, -\lambda + \bar{q}/2 + 1]$ for q even. These intervals are center in the larger interval $[-\lambda, -\lambda + \bar{q}]$ spanned by the interpolation points. A Lagrange-method with a uniform mesh will be called a uniform Lagrange-method and, when λ is chosen such that $-v$ is restricted to the central interpolation interval, it will be called a centered uniform Lagrange-method.* It is evident that, if a centered uniform Lagrange-method is to be passive for all $v \geq 0$, then the order cannot exceed two. (This limit is the counter-

*In general, a mesh is uniform when $\sigma_j = -\lambda + j\ell$. Obviously, Strang's result continues to be correct, but with respect to the central interpolation interval within a larger mesh, a mesh with a mesh-interval of ℓ . It was convenient for the results to be presented in this section to implicitly use $\ell=1$.

part of the order limit on a passive multistep-method.) Fortunately, because $\sum_{j=0}^{\bar{q}} \delta_j(v) \zeta^{\sigma_j}$ is a polynomial in v and ζ^{-1} , it is bounded for all $v \geq 0$, provided the interpolation points σ_j ($j=0, \dots, \bar{q}$) are suitably chosen and for $|\zeta| \geq 1$. Thus, a Lagrange-method of any order is β -passive.

P-STABILITY

Our initial result is with respect to

Definition 5: The multistep-method/Lagrange-method pair (13) and (17) is said to be P-stable if the (numerical) solution of (8) by (13) and (17) for all $\tau \geq 0$ and for any fixed positive h is asymptotic to zero when (8a) is p-dominant.

We now offer

Theorem 2: The multistep-method/Lagrange-method pair (13) and (17) is P-stable if the multistep-method is passive and if the Lagrange-method is passive.

and its

Proof: By Definition 5, (8a) is p-dominant. It therefore follows, by Theorem 1, that the solution of (8) is asymptotic to zero for all $\tau \geq 0$. As noted in the proof of that theorem, this is equivalent to

$$P(s, z) \neq 0 \quad (\operatorname{Re}(s) \geq 0, |z| \leq 1), \quad (26a)$$

where

$$P(s, z) = (s-p-qz)(s-\bar{p}-\bar{q}z). \quad (26b)$$

In that proof we validated (26a) by showing that

$$R(s, z) \neq 0 \quad (\operatorname{Re}\{s\} \geq 0, |z| \leq 1), \quad (27a)$$

where

$$R(s, z) = s-p-qz. \quad (27b)$$

Now consider the set of discrete time equations (8), in which the variables have been replaced by their previously noted approximations, with (13) and (17). If the solution is to be asymptotic to zero, then the characteristic function (easily shown to be) $h\sigma(\zeta)R[\rho(\zeta)/h\sigma(\zeta), \theta(\zeta)]$ must be nonzero for $|\zeta| \geq 1$, where $\rho(\zeta) = \sum_{i=-\kappa}^0 \alpha_i \zeta^{i+1}$, $\sigma(\zeta) = \sum_{i=-\kappa}^0 \beta_i \zeta^{i+1}$, and $\theta(\zeta) = \sum_{j=0}^{\bar{q}} \delta_j(\nu) \zeta^{\sigma_j}$. That this is true for ζ such that $|\zeta| \geq 1$ and $\sigma(\zeta) \neq 0$ follows from (27a) with (22) and (24). For the case of ζ such that $|\zeta| \geq 1$ and $\sigma(\zeta) = 0$, the characteristic function equals $\rho(\zeta)$, which is nonzero. ■

The orders of the multistep-method and of the Lagrange-method are related to the discretization error as follows: If $\bar{q} \geq \bar{p}-1$, if $u_{n+l} = u(t_{n+l}) + O[h^{\bar{p}+1}]$ for $l \in \{-\kappa, \dots, -1\} \cup \{\sigma_0, \dots, \sigma_{\bar{q}}\} \setminus \{0\}$, and if $u(t)$ is $\bar{p}+1$ times continuously differentiable for $t \in [t_{n-\kappa}, t_n] \cup [t_{n+\sigma_0}, t_{n+\sigma_{\bar{q}}}]$, then $u_n = u(t_n) + O[h^{\bar{p}+1}]$. We hereafter assume that $\bar{q} \geq \bar{p}-1$ and shall, as a consequence of this observation, refer to \bar{p} as the order of the multistep-method/Lagrange-method pair. As a consequence of Theorem 2 and of previous observations, we have

Theorem 3: The order of a, necessarily P-stable, passive multistep-method/passive Lagrange-method pair cannot exceed two.

If we should assume that $\bar{q} \geq \bar{p}$, then, additionally, the principal error term would be dominated by the principal error term of the multistep-method. This being the situation, a solution process for differential-difference equations based on a set of passive (hence, P-stable) multistep-method/Lagrange-method pairs could be created by augmenting a solution process for differential equations, based on the set of multistep-methods, by interpolators from the set of Lagrange-methods. The error-control—hence, also the step-control—procedure would not need to be changed. And, the order-control procedure could remain unchanged, but with only an upper limit of two on the order.

P[α, β]-STABILITY

The order constraint imposed by Theorem 3 can be circumvented by invoking other methods and/or by considering alternate stability types. We consider the latter in this section, beginning with

Definition 6: The multistep-method/Lagrange-method pair (13) and (17) is said to be P[α, β]-stable if the (numerical) solution of (8) by (13) and (17) for all $\tau \geq 0$ and for any fixed positive h is asymptotic to zero when

$$\operatorname{Re}\left\{\frac{1}{\beta} e^{\pm j(\pi/2-\alpha)P}\right\} < -|q|, \quad (28)$$

where $j = \sqrt{-1}$.

With respect to this definition we offer

Theorem 4: The multistep-method/Lagrange-method pair (13) and (17) is $P[\alpha, \beta]$ -stable if the multistep-method is α -passive and if the Lagrange-method is β -passive.

and its

Proof: The validity of (28) by Definition 6 implies the differential-difference equation

$$\dot{v}(\tau) = \frac{1}{\beta} e^{\pm j(\pi/2-\alpha)} p v(\tau) + q v(\tau-\tau) \quad (29)$$

is $\{\frac{1}{\beta} e^{\pm j(\pi/2-\alpha)} p\}$ -dominant. Therefore, by Theorem 1, the solution of (29) for any bounded, measurable initial function is asymptotic to zero for all $\tau \geq 0$. This implies and is implied by its characteristic function

$$Q(\sigma, e^{-\tau s}) = [\sigma - \frac{1}{\beta} e^{\pm j(\pi/2-\alpha)} p - q e^{-\tau s}] [\sigma - \frac{1}{\beta} e^{\pm j(\pi/2-\alpha)} p^* - q^* e^{-\tau s}] \quad (30)$$

being nonzero for $\text{Re}\{\sigma\} \geq 0$ and for all $\tau \geq 0$. As previously noted, this is equivalent to

$$Q(\sigma, \zeta) \neq 0 \quad (\text{Re}\{\sigma\} \geq 0, |\zeta| \leq 1) \quad (31a)$$

where

$$Q(\sigma, \zeta) = [\sigma - \frac{1}{\beta} e^{\pm j(\pi/2-\alpha)} p - q \zeta] [\sigma - \frac{1}{\beta} e^{\pm j(\pi/2-\alpha)} p^* - q^* \zeta]. \quad (31b)$$

This implies that both factors must be nonzero for $\text{Re}\{\sigma\} \geq 0$ and $|\zeta| \leq 1$. Since multiplication of either factor by an entire function does not alter this property, we have by implication (upon multiplying the left factor by $\beta e^{\mp j(\pi/2-\alpha)}$ and setting $s = \beta e^{\mp j(\pi/2-\alpha)} \sigma$ and $z = \beta e^{\mp j(\pi/2-\alpha)} \zeta$) that

$$R(s, z) \neq 0 \quad (|\arg\{s\}| \leq \pi - \alpha, |z| \leq \beta), \quad (32a)$$

where

$$R(s, z) = s - p - qz. \quad (32b)$$

This same function appeared in the proof of Theorem 2, but was, there, nonzero on a smaller domain. (See (26).) Now, as in the proof of Theorem 2, $h\sigma(\zeta)R[\rho(\zeta)/h\sigma(\zeta), \theta(\zeta)]$, the characteristic function in the solution of (8) with (13) and (17), must be nonzero for $|\zeta| \geq 1$. (The functions ρ , σ , and θ are as defined in the proof of Theorem 2.*) By definition 2, $|\arg\{\rho(\zeta)/h\sigma(\zeta)\}| \leq \pi - \alpha$ over the region of ζ such that $|\zeta| \geq 1$ and $\sigma(\zeta) \neq 0$ and, by Definition 4, the modulus of $\theta(\zeta)$ is bounded by β over the region $|\zeta| \geq 1$. Our result, for ζ such that $|\zeta| \geq 0$ and $\sigma(\zeta) \neq 0$, now follows from property (32a). For the case of ζ such that $|\zeta| \geq 1$ and $\sigma(\zeta) = 0$, the characteristic function equals $\rho(\zeta)$, which is nonzero.

Since there exist α -passive multistep-methods (such as the backward differentiation multistep-methods) of order exceeding two and since every

*In the first half of this proof σ and ζ were used as transition variables, with no relationship to the meaning ascribed to them hereon and in the proof of Theorem 2. We trust that no confusion has ensued or will ensue from doing this.

Lagrange-method is bounded, it follows that $P[\alpha, \beta]$ -stable multistep-method/Lagrange-method pairs exist. For uniform Lagrange-methods a β valid for all $v \geq 0$ is easily obtained for $q > 2$ as the maximum of $\sum_{j=0}^{\bar{q}} |\delta_j(v)|$ with respect to $-v \in [-\lambda + (\bar{q}+1)/2, -\lambda + \bar{q}]$ for \bar{q} odd and $-v \in [-\lambda + \bar{q}/2 + 1, -\lambda + \bar{q}]$ for \bar{q} even. In Table 1 we display order-by-order, for orders 1 through 6, the α and β values for the paired backward differentiation multistep-methods and uniform Lagrange-methods.

| P | α | β |
|---|----------|---------|
| 1 | 90.0° | 1.00 |
| 2 | 90.0° | 1.00 |
| 3 | 86.1° | 1.64 |
| 4 | 73.4° | 2.21 |
| 5 | 51.9° | 3.11 |
| 6 | 17.9° | 4.55 |

Table 1 $P[\alpha, \beta]$ -stable backward differentiation multistep-method/uniform Lagrange-method pairs.

The constraint (28) on p and q associated with a $P[\alpha, \beta]$ -stable method is illustrated in Figure 1. The value of p must lie within the sector when that of q lies within the disk of radius ρ .

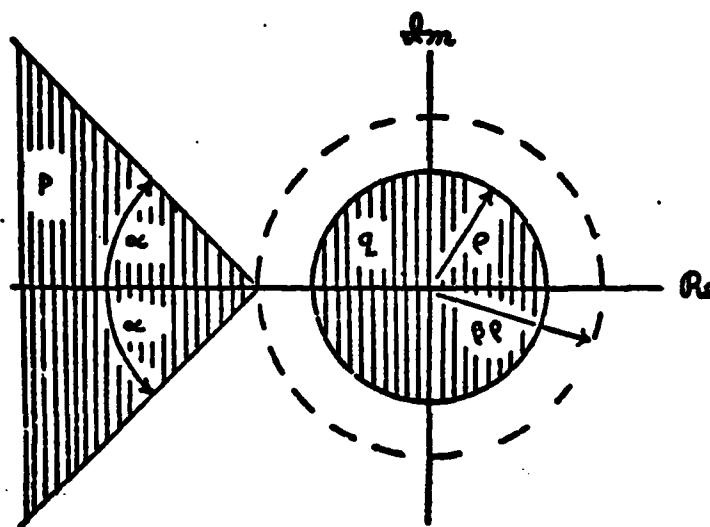


Figure 1 p and q regions for a $P[\alpha, \beta]$ -stable pair.

COMPOSITE MULTISTEP-METHODS

In this section we turn our attention to other methods; in particular, we focus on composite multistep-methods [12,13] as an alternative to multistep-methods. Such methods are characterized by the relation

$$\sum_{I=-K}^0 [A_I U_{N+I} - h B_I \dot{U}_{N+I}] = 0, \quad (33)$$

where

$$U_J = \begin{bmatrix} u_{J\ell-1} \\ \vdots \\ u_{(J-1)\ell} \end{bmatrix} \quad \text{and} \quad \dot{U}_J = \begin{bmatrix} \dot{u}_{J\ell-1} \\ \vdots \\ \dot{u}_{(J-1)\ell} \end{bmatrix}$$

are l -vectors of consecutive blocks of l contiguous approximants to the solution of (8) and where A_I ($I=-K, \dots, 0$) and B_I ($I=-K, \dots, 0$) are $(l \times l)$ -matrices which satisfy

$$\sum_{I=-K}^0 A_I \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = 0 \quad (34a)$$

and

$$\sum_{I=-K}^0 \left\{ A_I \begin{bmatrix} [(I-1)l]^j \\ \vdots \\ [(I-1)l]^j \end{bmatrix} - B_I \begin{bmatrix} j[(I-1)l]^{j-1} \\ \vdots \\ j[(I-1)l]^{j-1} \end{bmatrix} \right\} = 0 \quad (j=1, \dots, \bar{p}). \quad (34b)$$

As for multistep-methods, \bar{p} is the order. Note: We assume herein that the composite multistep-methods have zero defect; that is, that both A_0 and B_0 are of full rank, rank l . Now, with respect to the composite multistep-method we have

Definition 7: The composite multistep-method is said to be α -passive ($0 < \alpha \leq \pi/2$) if

$$\det\{[\sum_{I=-K}^0 A_I \zeta^{K+I}] - \lambda [\sum_{I=-K}^0 B_I \zeta^{K+I}]\} \neq 0 \quad (|\zeta| \geq 1, |\arg(-\lambda)| < \alpha). \quad (35)$$

This definition relates to those for passivity and α -passivity of a multistep-method (composite multistep-method with $l=1$) in the following way:

$\det\{\sum_{I=-K}^0 B_I \zeta^{K+I}\}$ is not identically zero because B_0 is of full rank. Therefore, $\det\{[\sum_{I=-K}^0 B_I \zeta^{K+I}]^{-1} [\sum_{I=-K}^0 A_I \zeta^{K+I}] - \lambda I\}$ is not zero for ζ such that $|\zeta| \geq 1$ and $\det\{\sum_{I=-K}^0 B_I \zeta^{K+I}\} \neq 0$ and for $|\arg(-\lambda)| < \alpha$.

This implies that the eigenvalues of $[\Sigma_{I=-K}^0 B_I \zeta^{K+I}]^{-1} [\Sigma_{I=-K}^0 A_I \zeta^{K+I}]$ are algebraic function of ζ whose arguments are contained in the interval $[-(\pi-\alpha), (\pi-\alpha)]$ for all ζ such that $|\zeta| \geq 1$ and $\det(\Sigma_{I=-K}^0 B_I \zeta^{K+I}) \neq 0$. This is just property (23) in the definition of α -passivity when $l=1$. It is equivalent to property (22) in the definition of passivity when $l=1$ and $\alpha = \pi/2$.

We now offer

Theorem 5: The composite multistep-method/Lagrange-method pair (33) and (17) with $\sigma_j = \xi_j \Delta$ is $P[\alpha, \beta]$ -stable if the composite multistep-method is α -passive and if the Lagrange-method is β -passive.*

and its

Proof: The first portion of the proof of this theorem is that for Theorem 4 through (32). We take-up this proof from that point. The numerical solution of (8) by (33) and by (17) with $\sigma_j = \xi_j \Delta$ engenders the difference equation

$$\Sigma_{I=-K}^0 \{A_I U_{I-N+I} - h B_I [p U_{N+I} + q \sum_{j=0}^{\bar{q}} \delta_j(\nu) U_{N+I+\xi_j}]\} = 0, \quad (36)$$

which has

$$\det\{[\Sigma_{I=-K}^0 A_I \zeta^{K+I}] - h [\Sigma_{I=-K}^0 B_I \zeta^{K+I}] [p + q \sum_{j=0}^{\bar{q}} \delta_j(\nu) \zeta^{\xi_j}]\} \quad (37)$$

*Definitions of the stability types when a composite multistep-method is used as the integrator were not given as they would be trivially different from Definitions 5 and 6 wherein just a multistep-method serves as the integrator.

as its characteristic function. If the numerical solution is to be asymptotic to zero, then (37) must be nonzero for $|\zeta| \geq 1$. Firstly, let us consider those ζ such that $|\zeta| \geq 1$ and $\det\{\Sigma_{I=K-I}^0 B_I \zeta^{K+I}\} \neq 0$. Then (37) is equivalent to

$$\det\left\{\frac{1}{h}\left[\Sigma_{I=K-I}^0 B_I \zeta^{K+I}\right]^{-1}\left[\Sigma_{I=K-I}^0 A_I \zeta^{K+I}\right] - [p+q \Sigma_{j=0}^q \delta_j(\nu) \zeta^{\epsilon_j}]\right\} \neq 0. \quad (38)$$

Now, let $\epsilon_j(\zeta)$ ($j=1, \dots, l$) denote the eigenvalues - algebraic functions of ζ - of $[\Sigma_{I=K-I}^0 B_I \zeta^{K+I}]^{-1} [\Sigma_{I=K-I}^0 A_I \zeta^{K+I}]$. Then, (38) may be replaced by

$$\frac{1}{h} \epsilon_j(\zeta) - p - q \Sigma_{j=0}^q \delta_j(\nu) \zeta^{\epsilon_j} \neq 0 \quad (j \in \{1, \dots, l\}), \quad (39a)$$

or, equivalently,

$$R[\epsilon_j(\zeta)/h, \theta(\zeta^{1/l})] \neq 0. \quad (j \in \{1, \dots, l\}). \quad (39b)$$

As previously noted, we have $|\arg \epsilon_j(\zeta)| \leq \pi - \alpha$ ($j \in \{1, \dots, l\}$), and, because $|\theta(\zeta)| \leq \beta$, we have $|\theta(\zeta^{1/l})| \leq \beta$. It therefore follows from (32) that property (39) is true for ζ such that $|\zeta| \geq 1$ and $\det\{\Sigma_{I=K-I}^0 B_I \zeta^{K+I}\} \neq 0$. Let us, now, secondly consider those ζ such that $|\zeta| \geq 1$ and $\det\{\Sigma_{I=K-I}^0 B_I \zeta^{K+I}\} = 0$. In this case (37) is equivalent to $\det\{\text{adj}\{\Sigma_{I=K-I}^0 B_I \zeta^{K+I}\} [\Sigma_{I=K-I}^0 A_I \zeta^{K+I}]\} \neq 0$. But this is an immediate consequence of (35) for such values of ζ .

If for the composite multistep-methods we were to use those reported by Bickart and Picel [14] and if for the Lagrange-methods we were to require a uniform mesh (mesh interval l , rather than the previous 1), then we would have a set of $P[\alpha, \beta]$ -stable composite multistep-method/Lagrange-method pairs with the α and β values displayed in Table 2. The Bickart-Picel

methods are one-step methods for which maximum order was sacrificed—the order is one less than the maximum possible—to achieve improved numerical solution properties for very large h .

| P | α | β |
|----|----------|---------|
| 1 | 90.0° | 1.00 |
| 2 | 90.0° | 1.00 |
| 3 | 88.9° | 1.64 |
| 4 | 87.7° | 2.21 |
| 5 | 85.5° | 3.11 |
| 6 | 82.7° | 4.55 |
| 7 | 79.5° | 6.93 |
| 8 | 76.0° | 10.95 |
| 9 | 72.5° | 17.85 |
| 10 | 69.6° | 29.91 |

Table 2 P[α, β]-stable Bickart-Picel composite multistep-method/uniform Lagrange-method pairs

If we were to use the maximum order one-step composite multistep-methods, then we would have a set of P[α, β]-stable pairs with, in particular, $\alpha=90.0^\circ$ for order 2 at $l=1$ through order 7 at $l=6$. [15-16]

COMPOSITE LAGRANGE-METHODS

Interpolation on a uniform mesh with an interval of l rather than 1 has a principal error term which is larger by a factor of $l^{\bar{q}+1}$. If l is large, then the principal error term of the integrator, which dominates that of the interpolator when $\bar{q} \geq \bar{p}$, may not be in fact significantly greater than that of the interpolator. This could have a deleterious effect on error control in a solution process using an approximation to the principal error term of

the integrator as the error measure. This is the reason we here remove the constraint $\sigma_j = \xi_j l$ and consider the resulting composite Lagrange-methods.

With respect to the block of solution values \underline{U}_J , the archetype equation (8a) establishes the relation

$$\underline{U}_J = p\underline{U}_J + q\underline{U}_{J-v} \quad (40)$$

with

$$\underline{U}_{J-v} = \begin{bmatrix} u_{Jl-1-v} \\ \vdots \\ u_{(J-1)l-v} \end{bmatrix}.$$

The interpolation relation (17) applied to each element of \underline{U}_{J-v} yields the composite interpolation relation

$$\underline{U}_{J-v} = \sum_{I=0}^{\bar{q}} \Delta_I(v) \underline{U}_{I+\xi_I}, \quad (41)$$

where each row of the (block row) matrix $[\Delta_{\bar{q}} \cdots \Delta_0]$ has $\bar{q}+1$ nonzero elements for almost all v , the coefficients $\delta_j(v)$ of (17). For example, when $l=3$, $\bar{q}=4$, and $(\sigma_0, \dots, \sigma_4) = (-8, -6, -5, -3, -2)$, then $\bar{Q}=3$, $(\xi_0, \xi_1, \xi_2, \xi_3) = (-3, -2, -1, 0)$, and the Δ_i have the nonzero structure displayed next:

$$\Delta_0 = \begin{bmatrix} 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \Delta_{-1} = \begin{bmatrix} x & 0 & x \\ x & x & 0 \\ 0 & x & x \end{bmatrix} \quad \Delta_{-2} = \begin{bmatrix} x & 0 & x \\ x & x & 0 \\ 0 & x & x \end{bmatrix} \quad \Delta_{-3} = \begin{bmatrix} 0 & 0 & 0 \\ x & 0 & 0 \\ 0 & x & 0 \end{bmatrix}$$

It is quite evident, based on (17) for each of the constituent relations, that

$$\begin{aligned} [\Delta_{\bar{Q}}^i; \dots; \Delta_0]_{ij} &= [\Delta_{\bar{Q}}^i; \dots; \Delta_0]_{(i-1), (j-1)} && (i=2, \dots, l, (j=1, \dots, (\bar{Q}+1)l)) \\ &= 0 && (i=2, \dots, l, (j=1, \dots, (i-1))). \end{aligned}$$

If each of the constituent relations were based on a different interpolation relation of the form of (17), then we could say only that $[\Delta_{\bar{Q}}^i; \dots; \Delta_0]$ would have $\bar{q}+1$ nonzero elements per row for almost all v and that

$$[\Delta_{\bar{Q}}^i; \dots; \Delta_0]_{ij} = 0 \quad (i=2, \dots, l, (j=1, \dots, (i-1))). \quad (42)$$

Note: \bar{q} could of course be different for each row, a case we will not develop in this paper. Composite interpolation relations (or composite Lagrange-methods) for which (42) is true are said to be cyclic.*

Since we are solving for the block of solution values \underline{U}_N , the i -th relation could depend on all of the values of \underline{U}_N , not just the i -th and those preceding it. So, for the i -th constituent relation the constraint on the σ_j can be relaxed, becoming $i-1 \geq \sigma_q > \dots > \sigma_0$ ($i=1, \dots, l$). The numerical solution of (8) by (33) and, in this general setting, (41) engenders the difference equation

$$\sum_{I=-K}^0 \{ A_I U_{N+I} - h B_I [p U_{N+I} + q \sum_{J=0}^{\bar{Q}} \Delta_J(v) U_{N+I+\Sigma J}] \} = Q, \quad (43)$$

which has

$$\det \{ [\sum_{I=-K}^0 A_I \zeta^{K+I}] - h [\sum_{I=-K}^0 B_I \zeta^{K+I}] [p I + q \sum_{J=0}^{\bar{Q}} \Delta_J(v) \zeta^{\Sigma J}] \} \quad (44)$$

*Such methods are said to be cyclic because, when one is paired with a cyclic composite multistep-method [17], the constituent equations of the resulting solution process can be solved cyclicly for the successive solution values in a block of solution values.

as its characteristic function. If the numerical solution is to be asymptotic to zero, then (44) must be nonzero for $|\zeta| \geq 1$.

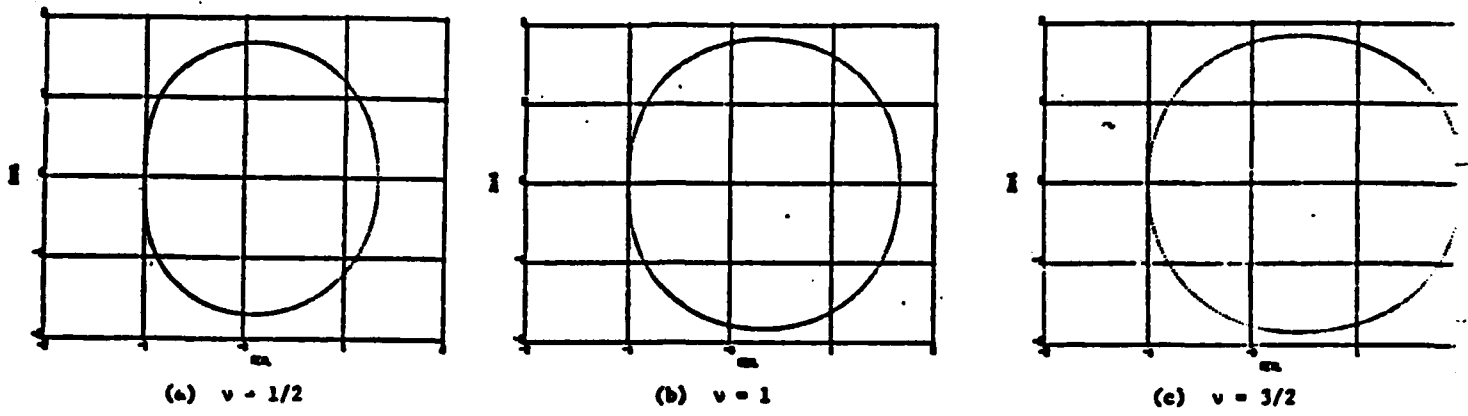
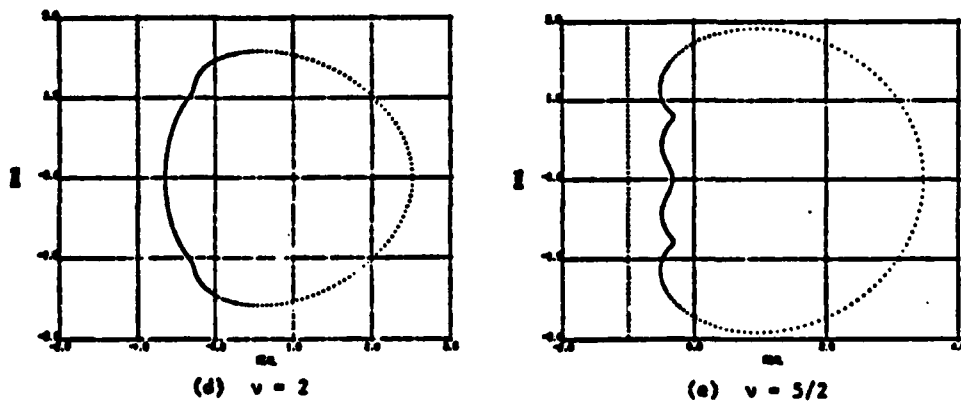
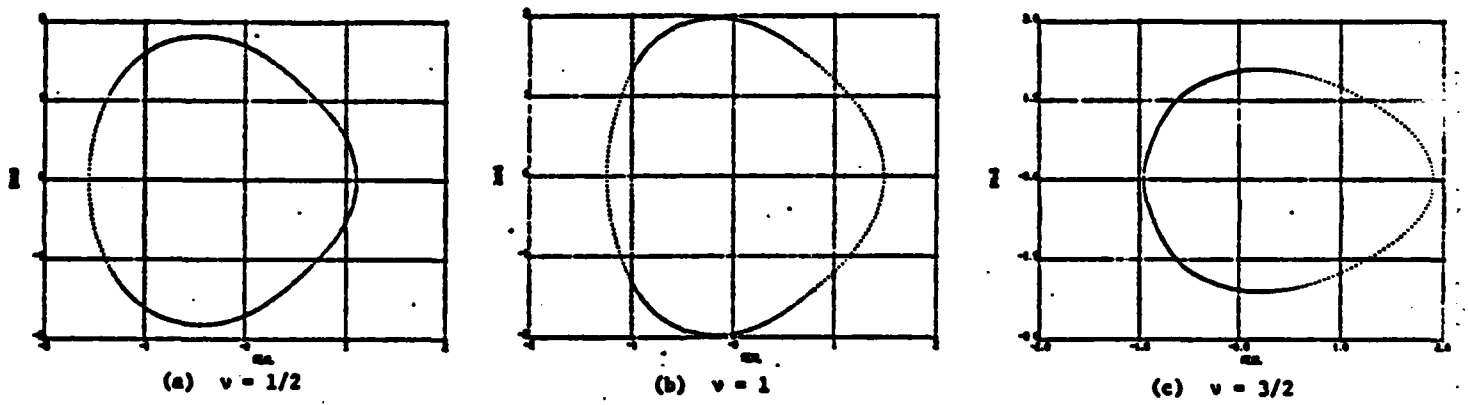
We have no analytic results on the stability properties of composite multistep-method/composite Lagrange-method pairs to report in this paper. However, for a few cases, we have obtained stability boundary plots when $v \in (0,1)$ to illustrate what might be expected of such pairs. We used the Bickart-Picel composite multistep-methods in conjunction with composite Lagrange-methods having Δ_{-l} with the nonzero structure displayed next:

$$l=2 \quad \Delta_{-0} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \quad \Delta_{-1} = \begin{bmatrix} 0 & X \\ 0 & X \end{bmatrix}$$

$$l=3 \quad \Delta_{-0} = \begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix} \quad \Delta_{-1} = \begin{bmatrix} 0 & X & 0 \\ 0 & X & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \Delta_{-2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ X & 0 & 0 \end{bmatrix}$$

$$l=4 \quad \Delta_{-0} = \begin{bmatrix} X & X & X & 0 \\ X & X & 0 & 0 \\ 0 & X & X & 0 \\ 0 & X & X & X \end{bmatrix} \quad \Delta_{-1} = \begin{bmatrix} X & X & 0 & 0 \\ 0 & X & 0 & X \\ 0 & 0 & X & X \\ 0 & 0 & 0 & X \end{bmatrix} \quad \Delta_{-2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 \\ X & 0 & 0 & 0 \\ X & 0 & 0 & 0 \end{bmatrix}$$

The corresponding plots, obtained by numerically solving the characteristic equation—characteristic function equated to zero—with $q=1$ and $v=1/2, 1, 3/2, \dots, l-1/2$, are displayed in Figures 2 through 4. With respect to these plots, we can at best expect these methods to be $P[90^\circ, \beta]$ -stable with $\beta=1, 1.56, 3.69$ for $l=2, 3, 4$ (respectively).

Figure 2 Stability boundary plots: $l = 2$ Figure 3 Stability boundary plots: $l = 3$

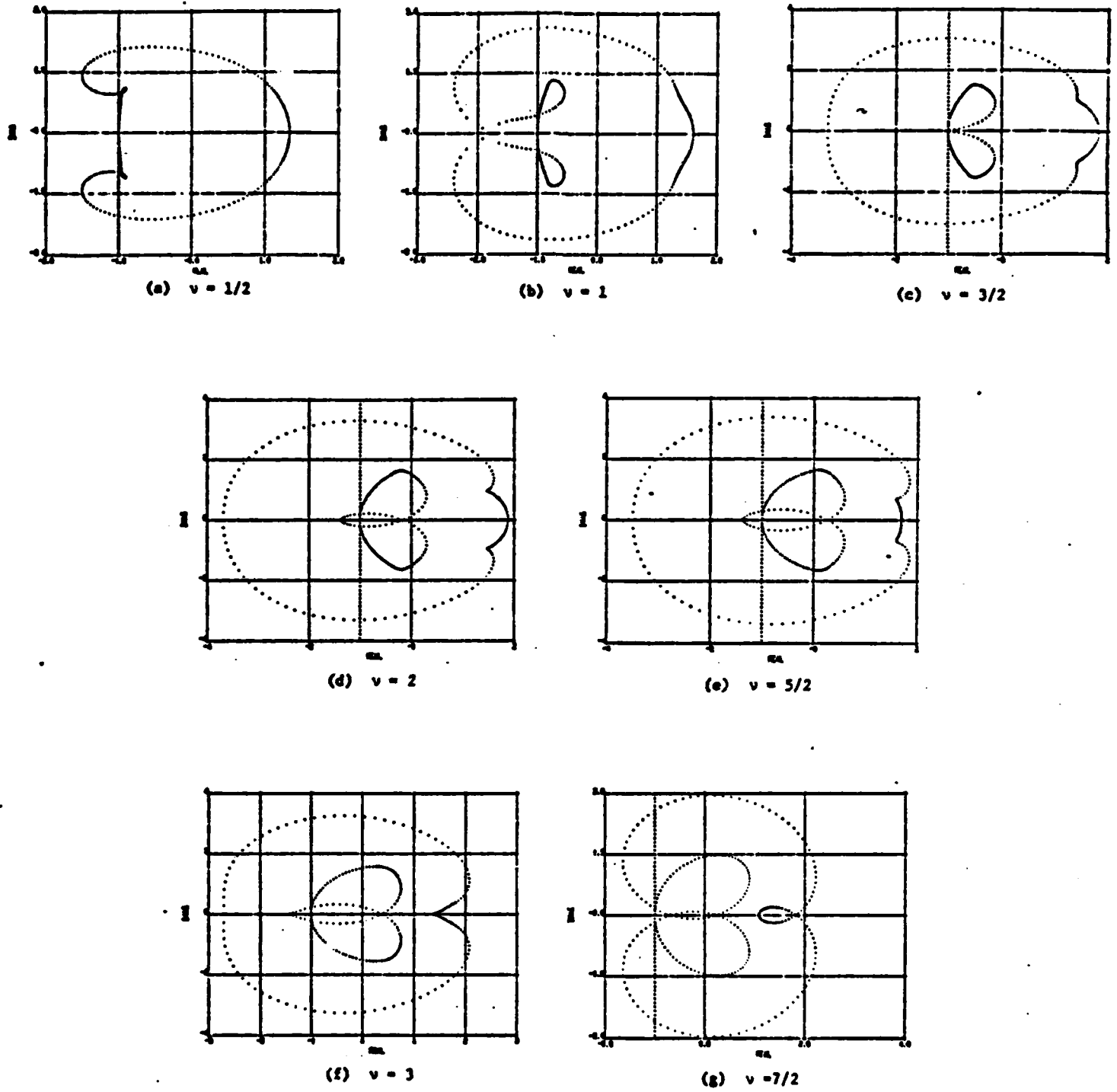


Figure 4 Stability boundary plots: $l = 4$

CONCLUDING DISCUSSION

We have herein introduced the properties of P-stability and $P[\alpha, \beta]$ -stability for integration/interpolation pairs upon which to base a process for solution of retarded differential-difference equations. We considered various pairs with respect to these properties; in particular, we introduced the pairing of a composite multistep-method with a composite Lagrange-method. In this instance we could present only limited non-analytic data on stability attributes. A larger base of such data led us to implement a variable-order, variable-step solution process based on a set of composite multistep-method/composite Lagrange-method pairs. The listing of a FORTRAN coded version, together with supporting documentation, is to be found in [18]. The listing and user manual of an APL coded version are provided in [19]. A simple illustration of the latter, drawn from [2], is provided in Plate 1.

ACKNOWLEDGEMENTS

The stability boundary plots were obtained by David Vander Staay using the symbolic algebra features of MACSYMA to determine (44) and the plotter with its supporting software at the General Electric Company Computer Center in Utica, NY to draw the curves derived from (44). MACSYMA, the symbolic manipulation system of the Massachusetts Institute of Technology Mathlab Group, is supported under Work Order 2095 from the Defense Advanced Research Projects Agency and under Contract N00014-75-C-0661 from the Office of Naval Research.

REFERENCES

- [1] T. A. Bickart, "F-stable and $F[\alpha, \beta]$ -stable integration/interpolation methods in the solution of retarded differential-difference equations," in preparation.
- [2] V. K. Barwell, Numerical solution of differential-difference equations, technical report CS-76-04, Department of Computer Science, University of Waterloo, Waterloo, Ontario, CANADA, 1976.
- [3] C. W. Cryer, "Highly stable multistep methods for retarded differential equations," SIAM Journal on Numerical Analysis, 11 (1974), 788-797.
- [4] R. Bellman and K. L. Cooke, Differential-Difference Equations, Academic Press, New York, 1963.
- [5] J. Hale, Functional Differential Equations, Springer-Verlag, New York, 1971.
- [6] E. W. Kamen, "On the relationship between zero criteria for two-variable polynomials and asymptotic stability of delay differential equations," IEEE Transactions on Automatic Control, AC-25 (1980), 983-984.
- [7] O. Nevanlinna and A. H. Sipila, "A nonexistence theorem for A-stable methods," Mathematics of Computation 28(1974), 1053-1055.
- [8] G. Dahlquist, "A special stability problem for linear multistep methods," BIT 3(1963), 27-43.
- [9] G. W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [10] G. Strang, "Trigonometric polynomials and differencing methods of maximum accuracy," Journal of Mathematics and Physics 41(1962), 147-154.
- [11] R. K. Brayton, "Numerical A-stability for difference-differential systems," in Stiff Differential Systems, R. A. Willoughby (editor), Plenum, New York, 1974.
- [12] H. M. Sloat and T. A. Bickart, "A-stable composite multistep methods," Journal ACM 20(1973), 7-26.
- [13] T. A. Bickart and W. B. Rubin, "Composite multistep methods and stiff stability," in Stiff Differential Systems, R. A. Willoughby (editor), Plenum, New York, 1974.
- [14] T. A. Bickart and Z. Picel, "High order stiffly stable composite multistep methods for numerical integration of stiff differential equations," BIT 13(1973), 272-286.

- [15] L. F. Shampine and H. A. Watts, "A-stable block implicit one-step methods," BIT 12(1972), 252-266.
- [16] T. A. Bickart, D. A. Burgess, and H. M. Sloate, "High order A-stable composite multistep methods for numerical integration of stiff differential equations," Proceedings Ninth Annual Allerton Conference on Circuit and System Theory, University of Illinois, Urbana, IL, 1971, 465-473.
- [17] J. M. Tendler, T. A. Bickart, and Z. Picel, "A stiffly stable integration process using cyclic composite methods," ACM Transactions on Mathematical Software 4(1978), 339-368.
- [18] D. L. Vander Staay, Composite integration/interpolation methods for the solution of stiff differential-difference equations, Ph.D. dissertation, Electrical and Computer Engineering Department, Syracuse University, Syracuse, NY, 1982.
- [19] T. A. Bickart, Stiff differential-difference equation solver: algorithm and program, technical report TR 81-7, Electrical and Computer Engineering Department, Syracuse University, Syracuse, NY, 1981.

$$\dot{y}(t) = -10^4 y(t) + y(t - \tau)$$

$$\tau = \ln(10^4 - 1)$$

$$y(t) = e^{-t} (te[-\tau, 0])$$

| | | ERROR TOLERANCE | | | | | | | | |
|---|---------|-----------------|--------------------|--------------------|-------|--------------------|--------------------|---------|--------------------|--------------------|
| | | 0.1 | | | 0.001 | | | 0.00001 | | |
| | | L | HF | ERROR | L | HF | ERROR | L | HF | ERROR |
| t | 0.00001 | 1 | 8.0E ⁻⁶ | 1.0E ⁻⁵ | 1 | 8.0E ⁻⁶ | 1.0E ⁻⁵ | 1 | 6.4E ⁻⁶ | 4.8E ⁻⁶ |
| | 0.0001 | 1 | 6.4E ⁻⁵ | 1.0E ⁻⁴ | 1 | 6.4E ⁻⁵ | 1.0E ⁻⁴ | 1 | 5.1E ⁻⁵ | 4.3E ⁻⁷ |
| | 0.001 | 1 | 5.1E ⁻⁴ | 1.0E ⁻⁴ | 1 | 5.1E ⁻⁴ | 3.3E ⁻⁶ | 1 | 6.1E ⁻⁴ | 1.6E ⁻⁷ |
| | 0.01 | 1 | 8.2E ⁻³ | 4.4E ⁻⁵ | 1 | 8.2E ⁻³ | 5.9E ⁻⁶ | 2 | 4.1E ⁻³ | 1.2E ⁻⁸ |
| | 0.1 | 1 | 6.6E ⁻² | 4.9E ⁻⁴ | 2 | 6.6E ⁻² | 1.6E ⁻⁵ | 3 | 3.3E ⁻² | 1.1E ⁻⁸ |
| | 1.0 | 2 | 5.2E ⁻¹ | 8.0E ⁻⁴ | 3 | 2.5E ⁻¹ | 2.4E ⁻⁵ | 5 | 2.0E ⁻¹ | 2.6E ⁻⁷ |
| | 10.0 | 1 | 3.0E0 | 4.7E ⁻⁵ | 4 | 1.3E0 | 2.9E ⁻⁷ | 5 | 5.6E ⁻¹ | 1.7E ⁻⁸ |

TRUE SOLUTION ON [0,10]

$$y(t) = e^{-t}$$

Plate 1 Example of DDE solver of [19].

DAVID J. NOLTING, Dept of Mathematics, U. S. Air Force Academy, Colorado, 80840 and DAVID J. RODABAUGH, Lockheed - Burbank, Ca 91520. Stiffly Stable Linear Multistep Methods. Abstract.

Currently popular codes for the solution of stiff ordinary differential equations employ the backwards difference formulae (BDF). Notwithstanding the computational advantage of the BDF, they suffer large regions of instability in the left half plane for orders four through six and are not A_0 -stable for orders greater than six. This roughly points out the need for more robust methods.

Desirable properties of linear multistep methods (LMM), such as size of stability regions, can be optimized by viewing those properties as functional values and the LMM possessing those properties as points in a domain space. This study conducts such an optimization numerically.

The concept of $A(\alpha, r)$ -stability is introduced for stiffly stable LMM. It recognizes the need for large regions of absolute stability in the left half plane and the need for a region of accuracy about the origin defined by the region of relative stability. An economical means of determining the region of relative stability is developed and used. Nearly-optimal $A(\alpha, r)$ -stable implicit LMM are found for orders four through six for a variety of classes determined by fixed error constants $C_{p+1}/\sigma(1)$.

STIFFLY STABLE LINEAR MULTISTEP METHODS

1. Introduction

Consider the initial value problem

$$y' = f(x,y), x \in [a,b], y(a) = y_0 \quad (1)$$

where y , y_0 , and $f(x,y)$ are in R^n and sufficient conditions are placed on f to ensure a unique solution exists. In the past fifteen years much research has been devoted to the development of numerical methods for obtaining solutions to problems (1) of the class referred to as stiff problems. Commonly used numerical methods are linear k -step methods of the form

$$\sum_{i=-1}^{k-1} \alpha_i y_{n-i} + h \sum_{i=-1}^{k-1} \beta_i y'_{n-i} = 0,$$

where $\alpha_{-1} = -1$ and $|\alpha_{k-1}| + |\beta_{k-1}| \neq 0$. Most popular codes which employ LMM for the solution of stiff problems depend exclusively on the backward difference formulas (BDF) [3, p.217]. These methods suffer large regions of instability in the left half plane and are not A_0 -stable [1] for orders greater than six. This suggests much improvement is possible.

Grigorieff and Schroll [4] and Kong [7] have given constructive proofs which show the existence of $A(\alpha)$ -stable [12] methods of arbitrarily high order with α arbitrarily close to $\pi/2$. However, searches for practical methods successful in the solution of stiff problems have made little progress [7, 11, 6, 5, 2]. None of these investigations have resulted in methods which perform significantly better than BDF.

2. Defining the problem.

There are several properties of LMM which appear decisive in determining the performance of a LMM in the solution of a stiff problem. To conduct a search for optimal methods we need to isolate these properties, determine a useful measure for each property, and define our object function in terms of these measures. The region of absolute stability is of prime interest because of the need to maintain stability for the components resulting from the eigenvalues with negative real part and large magnitude. The angle α is a measure of the absolute stability region which is widely used.

The region of accuracy about the origin is important because the components of the solution resulting from eigenvalues near the origin are dominant in the solution and their accuracy must be of concern. A natural measure for this region is the radius of relative stability. A knowledge of the relative stability characteristics is essential before we attach meaning to the numerical solution.

To illustrate the necessity of measuring relative stability we solve an example problem in 35-digit precision with the 5th order Adams-Bashforth (A-B) method and with a 7th order explicit near-optimal method [9] of better relative stability characteristics. The problem considered is

$$y' = -16y, y(0) = 1, x \in [0, 3].$$

We use a stepsize of $h = 0.01$. Thus we have $h\lambda = 0.16$ which is inside both methods' region of absolute stability. However it is well outside the A-B 5th order relative stability region and near the boundary of relative stability for the 7th order method. This problem gives us an example of what can happen when we forget about relative stability in the left half plane. The results are given below in table 1.

TABLE 1

Illustration of the importance of relative stability

| Step No. | True Solution | A-B 5th Order | | 7th Order Near-Optimal | |
|----------|---------------|---------------|--------------|------------------------|--------------|
| | | Calculated | Actual Error | Calculated | Actual Error |
| 0 | 0.1000Q+01 | 0.1000Q+01 | 0.0 | 0.1000Q+01 | 0.0 |
| 50 | 0.3355Q-03 | 0.3357Q-03 | -0.1988Q-06 | 0.3354Q-03 | 0.6280Q-07 |
| 100 | 0.1125Q-06 | 0.3001Q-06 | -0.1875Q-06 | 0.1125Q-06 | 0.8515Q-10 |
| 150 | 0.3775Q-10 | 0.1108Q-06 | -0.1108Q-06 | 0.3769Q-10 | 0.6116Q-13 |
| 200 | 0.1266Q-13 | 0.6541Q-07 | -0.6541Q-07 | 0.1263Q-13 | 0.3401Q-16 |
| 250 | 0.4248Q-17 | 0.3862Q-07 | -0.3862Q-07 | 0.4231Q-17 | 0.1686Q-19 |
| 300 | 0.1425Q-20 | 0.2280Q-07 | -0.2280Q-07 | 0.1418Q-20 | 0.7640Q-23 |

If we had solved this problem with the A-B 5th order method and desired to follow the solution to this problem (possibly a component of a larger system) until the solution fell below 10^{-10} for example, we would find it necessary to continue to step 825. Whereas in fact, and as detected by the relatively stable 7th order near-optimal method, we could terminate calculation after step 150. In this case and others like it, ignoring the relative stability characteristics of a method can be a very costly choice. The relative error for the A-B solution in Table 1 after 300 steps is $1.6000Q+13$ and for the near optimal method is $5.3610Q-03$. Note that the A-B solution is tending to zero, which evidences the methods' absolute stability at $h\lambda = -0.16$, but not nearly so fast as the solution itself. In fact it tends to zero so slowly that the calculated solution values are essentially meaningless.

These considerations led to the definition of $A(\alpha, r)$ -stability given below.

Definition: A LMM is said to be $A(\alpha, r)$ -stable if the method is $A(\alpha)$ -stable with regard to its region of absolute stability and is relatively stable

within the disk of radius r about the origin.

It is this definition which we implement in our search for stiffly stable LMM. It seems of interest then to know how far α and r can be extended for a fixed value of the error constant. We report the results of such an investigation in this paper and find methods nearly optimal with regard to these desired properties. Further, in comparison we find these near-optimal methods perform successfully as we would expect from the stability and error measurements applied. A method which is $A(\alpha, r)$ -stable will be relatively robust depending upon the size of α and r . By that we mean it should give good results on a large variety of both stiff and non-stiff problems.

Let $c(n,m)$ be the class of all n th order, m -step correctors. We choose the class $C(n, n)$ for our search of optimal methods. The corrector search is posed as a numerical optimization problem. Our object function for the optimization is defined as a linear combination of α and r .

3. Results and Comparisons

Defining the object function as a linear combination of α and r increased the scope of the optimization since then our interest extends to the effect of taking different linear combinations. In doing so we find a relationship existing between α and r , corresponding to maximal values of different linear combinations. (Different relationships exist for different values of the error constant $C_{p+1}/\sigma(1)$.) The two properties are inversely related but not linearly so. For example, the following relationships exist within $C(4,4)$.

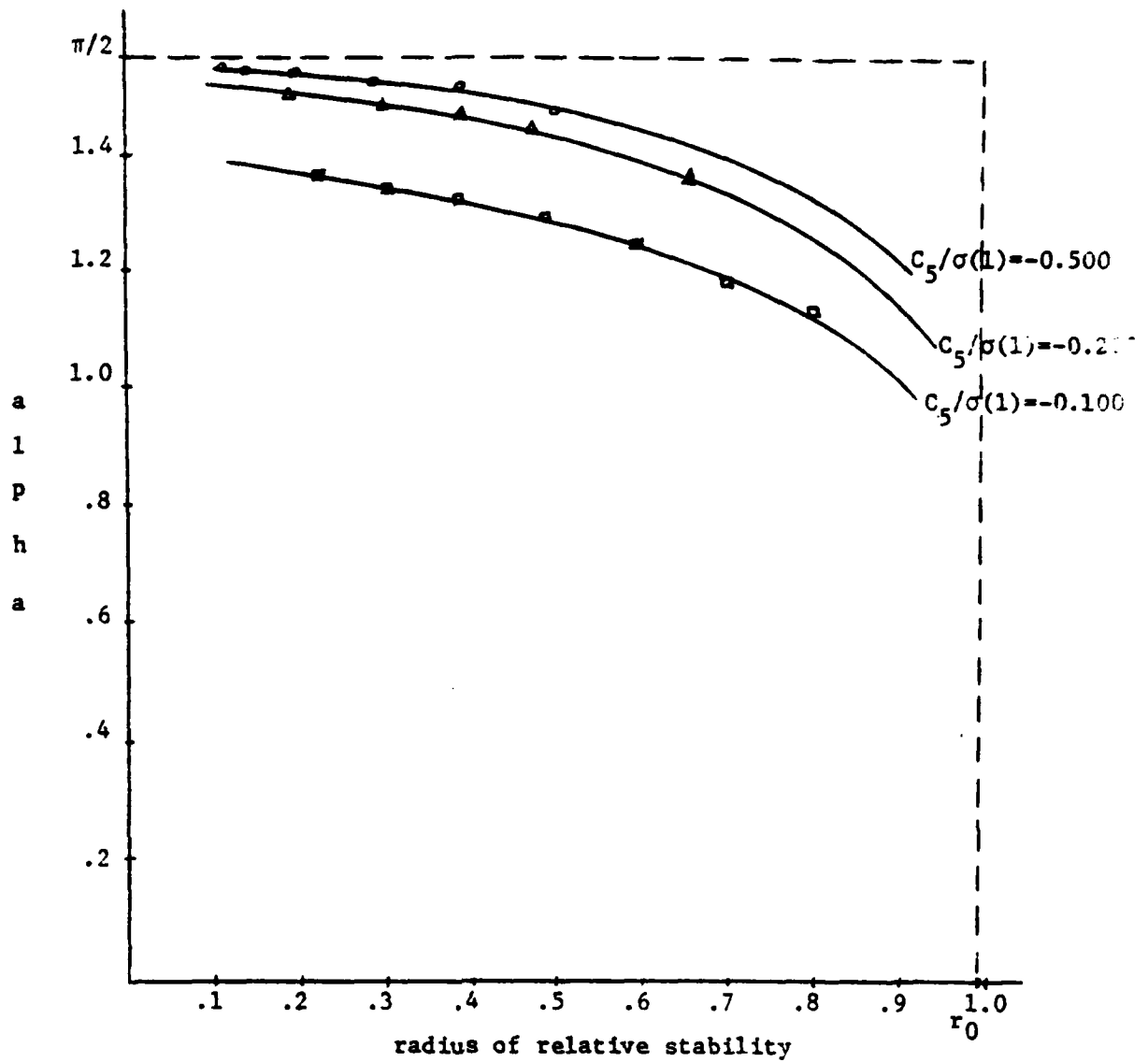


Figure 1. α vs. r .

The three curves result from fixing the error constant, $C_{p+1}/\sigma(1)$, at the values given adjacent to the curves. Notice the curves seem to approach the limiting α -value of $\pi/2$ for larger values of the radius as the error constraint is relaxed. This leads one to suspect these curves, corresponding to values of $C_{p+1}/\sigma(1)$ in $(0, \infty)$, fill in the area under the line $\alpha = \pi/2$ and to the left of

the line $r = r_0$, where r_0 is the largest radius of any method in the class $C(4,4)$. An estimate for the value of r_0 is provided by the work of Thompson and Rodabaugh [10], where they report candidates for r_0 values for the classes $C(n, n-1)$, $4 \leq n \leq 9$. They obtained these values by a numerical optimization with no constraint on the error constant. We did not investigate thoroughly for all orders and error constants the portion of the curve defined by linear combinations in which the radius was weighted much more than alpha. The methods in this portion of the curve are expensive to find, and of little value in the solution of stiff problems because of their poor absolute stability properties. Kong [7] considered only absolute stability, and therefore his results define the upper extent of the curve on the alpha axis. The optimization runs for these methods are inexpensive, however, they also are of little value to this investigation because of their poor relative stability properties. For orders greater than four we concentrate our effort in the region about and just to the left of the point where the curve starts dropping off most rapidly. This region of the curve yields methods which pair values of α and r in a way least costly to either property.

Several methods obtained from the investigation are described here for orders four through six. The properties of the second order trapezoidal rule and the third order BDF suggest little or no improvement possible below fourth order. Since the BDF are members of $C(n, n)$ and are used extensively in stiff codes, we use them for comparison. We choose a selection of near-optimal methods with error constants $C_{p+1}/\sigma(1)$ of -0.2 and -0.5 for fourth order, -0.4 and -0.8

for fifth order, and -0.9 for sixth order. Their stability properties are outlined in Table 2.

TABLE 2
Stability Properties of Near-Optimal Stiff Methods.

| Order | Alpha | Radius | $C_{p+1}/\sigma(1)$ |
|-------|--------|--------|---------------------|
| 4 | 1.481 | 0.192 | -0.200 |
| | 1.414 | 0.471 | |
| | 1.377* | 0.650 | |
| 4 | 1.535 | 0.142 | -0.500 |
| | 1.511 | 0.294 | |
| | 1.445 | 0.514 | |
| 5 | 1.431* | 0.092 | -0.400 |
| | 1.338 | 0.359 | |
| | 1.259 | 0.497 | |
| 5 | 1.485 | 0.077 | -0.800 |
| | 1.463 | 0.155 | |
| | 1.394 | 0.463 | |
| 6 | 1.321* | 0.121 | -0.900 |
| | 1.284 | 0.299 | |
| | 1.065 | 0.435 | |

* Indicates use in test runs referred to Table 4.

For comparison we give corresponding information for the BDF in Table 3 below.

TABLE 3
Stability Properties of the BDF

| Order | Alpha | Radius | $C_{p+1}/\sigma(1)$ |
|-------|-------|--------|---------------------|
| 4 | 1.280 | 0.484 | -0.200 |
| 5 | 0.905 | 0.302 | -0.167 |
| 6 | 0.311 | 0.130 | -0.143 |

The k th order BDF has an error constant of $1/(k+1)$. The severity of this restriction on the error constant no doubt causes the loss of A_0 -stability for the higher order BDF. Notice for each order, the near-optimal methods have considerably greater regions of relative and absolute stability. The effect of the larger error constants is nullified by the capability to increase the order with no accompanying restriction on α . For example, if we compare the near-optimal sixth order $A(1.284, 0.299)$ -stable method from Table 2 with the fourth order BDF, it is the case that the problem independent part of the error term for the near-optimal method will be less than that of the BDF for stepsizes less than 0.471. This particular comparison does not involve a sacrifice in the permissible values of α . In other similar comparisons we could even realize a gain in the permissible values of α .

4. Demonstration of methods obtained.

We note that an increase in the error constant accompanying methods with higher values of α is a negative aspect that can be controlled. However, an inferior value of α , such as possessed by the fifth and sixth order BDF, severely disables a method. We illustrate this fact by solving the following two stiff systems of the form $y'(t) = Ay(t)$ where A is a constant 4×4 real matrix.

Problem 1 (P1)

$$y' = Ay, t \in [0, 5], y(0) = (2.0, 0.0, -0.99, 3.73)^T$$

$$A = \begin{bmatrix} -1 & 0 & 100 & 0 \\ 0 & -1 & 0 & 100 \\ 0 & 0 & -100 & -373 \\ 0 & 0 & 373 & -100 \end{bmatrix}$$

Problem 2 (P2)

$$y' = Ay, t \in [0, 5], y(0) = (2.0, 0.0, -0.99, 2.5)^T$$

$$A = \begin{bmatrix} -1 & 0 & 100 & 0 \\ 0 & -1 & 0 & 100 \\ 0 & 0 & -100 & -250 \\ 0 & 0 & 250 & -100 \end{bmatrix}$$

Table 4 indicates the methods used on each of the problems. A constant stepsize of $h = 0.005$ was used for each of the solutions throughout the interval $[0, 5]$. Results of the calculations, as well as coefficients of the methods used, are given in the appendix. All calculations were carried out on an Amdahl 370/V7 in 35-digit precision. Numbers less than approximately 10^{-78} in magnitude are represented as 0.0.

TABLE 4

| Order of Method | Method Used | Problem |
|-----------------|------------------------|---------|
| 4 | A(1.377, 0.650)-Stable | P1 |
| 4 | BDF | P1 |
| 5 | A(1.431, 0.092)-Stable | P2 |
| 5 | BDF | P2 |
| 6 | A(1.321, 0.121)-Stable | P2 |
| 6 | BDF | P2 |

Referring to the appendix we see the sixth order near-optimal method solved problem P2 accurately whereas the fifth order EDF was unstable. Since these computations were completed we have found a seventh order method which also solved problem P2.

Lambert [8, p.479] indicates an interest exists in the use of stiff methods for non-stiff problems. With their large regions of accuracy about the origin these near-optimally $A(\alpha, r)$ -stable methods should be ideally suited for this purpose. In test runs comparing

these methods to Adams-Moulton methods, we solved non-stiff problems with a method of one order higher than the Adams-Moulton method used, and obtained a smaller actual error. The larger error constants of these methods is more than offset by using a method of one order higher. Using a method of higher order is permitted by the larger stability regions.

A large number of methods were found in this investigation, and only a few have been mentioned or used. Space has not permitted a full discussion of the procedure used to find these methods. For a further discussion of these procedures refer to Nolting [9].

APPENDIX

TABLE A-1

First Component Solution of Problem P1

| Step (h=0.005) | True Solution | Fourth Order Near-Optimal | | Fourth Order BDF | |
|-------------------|------------------|---------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.2000Q+01 | 0.2000Q+01 | 0.0 | 0.2000Q+01 | 0.0 |
| 200 | 0.3679Q+00 | 0.3679Q+00 | -0.5267Q-06 | 0.9475Q+01 | -0.9107Q+01 |
| 400 | 0.1353Q+00 | 0.1353Q+00 | 0.3970Q-10 | 0.2122Q+03 | -0.2121Q+03 |
| 600 | 0.4979Q-01 | 0.4979Q-01 | 0.1872Q-10 | -0.8181Q+03 | 0.8181Q+03 |
| 800 | 0.1832Q-01 | 0.1832Q-01 | 0.9199Q-11 | -0.1285Q+06 | 0.1285Q+06 |
| 1000 | 0.6738Q-02 | 0.6738Q-02 | 0.4234Q-11 | -0.2008Q+07 | 0.2008Q+07 |

TABLE A-2

Second Component Solution of Problem P1

| Step (h=0.005) | True Solution | Fourth Order Near-Optimal | | Fourth Order BDF | |
|-------------------|------------------|---------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 200 | 0.2794Q-43 | -0.1916Q-05 | 0.1916Q-05 | 0.5968Q+01 | -0.5968Q+01 |
| 400 | 0.0 | 0.4772Q-11 | -0.4772Q-11 | -0.1348Q+03 | 0.1348Q+03 |
| 600 | 0.0 | -0.1773Q-17 | 0.1773Q-17 | -0.5742Q+04 | 0.5742Q+04 |
| 800 | 0.0 | -0.5826Q-22 | 0.5826Q-22 | -0.3738Q+05 | 0.3738Q+05 |
| 1000 | 0.0 | 0.3869Q-27 | -0.3869Q-27 | 0.2348Q+07 | -0.2348Q+07 |

TABLE A-3

Third Component Solution of Problem P1

| Step (h=0.005) | True Solution | Fourth Order Near-Optimal | | Fourth Order BDF | |
|-------------------|------------------|---------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | -0.9900Q+00 | -0.9900Q+00 | 0.0 | -0.9900Q+00 | 0.0 |
| 200 | -0.7988Q-43 | 0.6626Q-05 | -0.6626Q-05 | -0.3128Q+02 | 0.3128Q+02 |
| 400 | 0.0 | -0.1198Q-10 | 0.1198Q-10 | 0.2928Q+03 | -0.2928Q+03 |
| 600 | 0.0 | -0.2188Q-16 | 0.2188Q-16 | 0.2223Q+05 | -0.2223Q+05 |
| 800 | 0.0 | 0.3095Q-21 | -0.3095Q-21 | 0.2669Q+06 | -0.2669Q+06 |
| 1000 | 0.0 | -0.1601Q-26 | 0.1601Q-26 | -0.6769Q+07 | 0.6769Q+07 |

TABLE A-4

Fourth Component Solution of Problem P1

| Step (h=0.005) | True Solution | Fourth Order Near-Optimal | | Fourth Order BDF | |
|-------------------|------------------|---------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.3730Q+01 | 0.3730Q+01 | 0.0 | 0.3730Q+01 | 0.0 |
| 200 | -0.1193Q-42 | 0.3862Q+05 | -0.3862Q+05 | 0.2806Q+02 | -0.2806Q+02 |
| 400 | 0.0 | -0.2664Q-10 | 0.2664Q-10 | 0.9245Q+03 | -0.9245Q+03 |
| 600 | 0.0 | 0.1091Q-15 | -0.1091Q-15 | -0.2633Q+04 | 0.2633Q+04 |
| 800 | 0.0 | -0.2697Q-21 | 0.2697Q-21 | -0.4424Q+06 | 0.4424Q+06 |
| 1000 | 0.0 | 0.2124Q-27 | -0.2124Q-27 | -0.9815Q+07 | 0.9815Q+07 |

TABLE A-5

First Component Solution of Problem P2 by Fifth Order Methods

| Step (h=0.005) | True Solution | Fifth Order Near-Optimal | | Fifth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.2000Q+01 | 0.2000Q+01 | 0.0 | 0.2000Q+01 | 0.0 |
| 200 | 0.3679Q+00 | 0.3679Q+00 | -0.2413Q-11 | 0.9366Q+03 | -0.9366Q+03 |
| 400 | 0.1353Q+00 | 0.1353Q+00 | -0.1687Q-12 | 0.1696Q+08 | -0.1696Q+08 |
| 600 | 0.4979Q-01 | 0.4979Q-01 | -0.9338Q-13 | 0.2666Q+12 | -0.2666Q+12 |
| 800 | 0.1832Q-01 | 0.1832Q-01 | -0.4588Q-13 | 0.2852Q+16 | -0.2852Q+16 |
| 1000 | 0.6738Q-02 | 0.6738Q-02 | -0.2112Q-13 | -0.2026Q+20 | 0.2026Q+20 |

TABLE A-6

Second Component Solution of Problem P2 by Fifth Order Methods

| Step (h=0.005) | True Solution | Fifth Order Near-Optimal | | Fifth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 200 | -0.3610Q-43 | -0.5067Q-12 | 0.5067Q-12 | 0.2558Q+04 | -0.2558Q+04 |
| 400 | 0.0 | -0.4956Q-23 | 0.4946Q-23 | 0.6429Q+08 | -0.6429Q+08 |
| 600 | 0.0 | -0.1757Q-31 | 0.1757Q-31 | 0.1601Q+13 | -0.1601Q+13 |
| 800 | 0.0 | -0.6335Q-32 | 0.6335Q-32 | 0.3953Q+17 | -0.3953Q+17 |
| 1000 | 0.0 | -0.1333Q-32 | 0.1333Q-32 | 0.9673Q+21 | -0.9673Q+21 |

TABLE A-7

Third Component Solution of Problem P2 by Fifth Order Methods

| Step (h=0.005) | True Solution | Fifth Order Near-Optimal | | Fifth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | -0.9900Q+00 | -0.9900Q+00 | 0.0 | -0.9900Q+00 | 0.0 |
| 200 | 0.8139Q-43 | -0.8977Q-12 | 0.8977Q-12 | -0.7321Q+04 | 0.7321Q+04 |
| 400 | 0.0 | 0.7781Q-22 | -0.7781Q-22 | -0.1775Q+09 | 0.1775Q+09 |
| 600 | 0.0 | -0.4609Q-32 | 0.4609Q-32 | -0.4267Q+13 | 0.4267Q+13 |
| 800 | 0.0 | 0.2243Q-42 | -0.2243Q-42 | -0.1016Q+18 | 0.1016Q+18 |
| 1000 | 0.0 | -0.9632Q-53 | 0.9632Q-53 | -0.2398Q+22 | 0.2398Q+22 |

TABLE A-8

Fourth Component Solution of Problem P2 by Fifth Order Methods

| Step (h=0.005) | True Solution | Fifth Order Near-Optimal | | Fifth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Value | Calculated Value | Global Error |
| 0 | 0.2500Q+01 | 0.2500Q+01 | 0.0 | 0.2500Q+01 | 0.0 |
| 200 | 0.5816Q-43 | 0.5967Q-11 | -0.5967Q-11 | -0.1914Q+03 | 0.1914Q+03 |
| 400 | 0.0 | -0.1604Q-21 | 0.1604Q-21 | -0.2124Q+08 | 0.2124Q+08 |
| 600 | 0.0 | 0.4982Q-32 | -0.4982Q-32 | -0.9187Q+12 | 0.9187Q+12 |
| 800 | 0.0 | -0.1293Q-42 | 0.1293Q-42 | -0.3200Q+17 | 0.3200Q+17 |
| 1000 | 0.0 | 0.2167Q-53 | -0.2167Q-53 | -0.1008Q+22 | 0.1008Q+22 |

TABLE A-9

First Component Solution of Problem P2 by Sixth Order Methods

| Step (h=0.005) | True Solution | Sixth Order Near-Optimal | | Sixth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.2000Q+01 | 0.2000Q+01 | 0.0 | 0.2000Q+01 | 0.0 |
| 200 | 0.3679Q+00 | 0.3679Q+00 | -0.3246Q-09 | -0.1667Q+14 | 0.1667Q+14 |
| 400 | 0.1353Q+00 | 0.1353Q+00 | 0.3810Q-14 | 0.3979Q+29 | -0.3979Q+29 |
| 600 | 0.4979Q-01 | 0.4979Q-01 | 0.2113Q-14 | -0.8016Q+44 | 0.8016Q+44 |
| 800 | 0.1832Q-01 | 0.1832Q-01 | 0.1039Q-14 | 0.1467Q+60 | -0.1467Q+60 |
| 1000 | 0.6738Q-02 | 0.6738Q-02 | 0.4786Q-15 | -0.2508Q+75 | 0.2508Q+75 |

TABLE A-10

Second Component Solution of Problem P2 by Sixth Order Methods

| Step (h=0.005) | True Solution | Sixth Order Near-Optimal | | Sixth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 200 | -0.3610Q-43 | 0.6007Q-10 | -0.6007Q-10 | -0.4319Q+14 | 0.4319Q+14 |
| 400 | 0.0 | -0.2089Q-18 | 0.2089Q-18 | 0.5945Q+29 | -0.5945Q+29 |
| 600 | 0.0 | -0.6950Q-28 | 0.6950Q-28 | -0.7611Q+44 | 0.7611Q+44 |
| 800 | 0.0 | -0.2182Q-31 | 0.2182Q-31 | 0.8753Q+59 | -0.8753Q+59 |
| 1000 | 0.0 | -0.5375Q-32 | 0.5375Q-32 | -0.8216Q+74 | 0.8216Q+74 |

TABLE A-11

Third Component Solution of Problem P2 by Sixth Order Methods

| Step (h=0.005) | True Solution | Sixth Order Near-Optimal | | Sixth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | -0.9900Q+00 | -0.9900Q+00 | 0.0 | -0.9900Q+00 | 0.0 |
| 200 | 0.8139Q-43 | -0.4715Q-09 | 0.4715Q-09 | 0.1245Q+15 | -0.1245Q+15 |
| 400 | 0.0 | 0.4503Q-18 | -0.4503Q-18 | -0.1880Q+30 | 0.1880Q+30 |
| 600 | 0.0 | 0.3034Q-27 | -0.3034Q-27 | 0.2696Q+45 | -0.2696Q+45 |
| 800 | 0.0 | -0.1408Q-36 | 0.1408Q-36 | -0.3640Q+60 | 0.3640Q+60 |
| 1000 | 0.0 | -0.1651Q-45 | 0.1651Q-45 | 0.4537Q+75 | -0.4537Q+75 |

TABLE A-12

Fourth Component Solution of Problem P2 by Sixth Order Methods

| Step (h=0.005) | True Solution | Sixth Order Near-Optimal | | Sixth Order BDF | |
|-------------------|------------------|--------------------------|--------------|------------------|--------------|
| | | Calculated Value | Global Error | Calculated Value | Global Error |
| 0 | 0.2500Q+01 | 0.2500Q+01 | 0.0 | 0.2500Q+01 | 0.0 |
| 200 | 0.5816Q-43 | 0.7519Q-09 | -0.7519Q-09 | 0.1089Q+13 | -0.1089Q+13 |
| 400 | 0.0 | 0.3885Q-18 | -0.3885Q-18 | 0.4061Q+29 | -0.4061Q+29 |
| 600 | 0.0 | -0.2590Q-27 | 0.2590Q-27 | -0.1250Q+45 | 0.1250Q+45 |
| 800 | 0.0 | -0.2275Q-36 | 0.2275Q-36 | 0.2800Q+60 | -0.2800Q+60 |
| 1000 | 0.0 | 0.7022Q-46 | -0.7022Q-46 | -0.5457Q+75 | 0.5457Q+75 |

TABLE A-13

Rational Coefficients of the Near Optimal Methods Used in the Solution of Problems P1 and P2

| Coefficient Numerator | Fourth Order | Fifth Order | Sixth Order |
|----------------------------|--------------|-------------|-------------|
| α_0 | 464 400 | 27 720 000 | 49 824 000 |
| α_1 | -306 600 | -35 100 000 | -72 000 000 |
| α_2 | 87 288 | 22 500 000 | 53 280 000 |
| α_3 | -5 088 | -7 200 000 | -18 720 000 |
| α_4 | | 1 080 000 | 1 440 000 |
| α_5 | | | 576 000 |
| β_{-1} | 109 512 | 4 159 875 | 6 563 520 |
| β_0 | 24 165 | -4 117 125 | -9 114 120 |
| β_1 | -64 993 | -1 212 750 | 1 415 800 |
| β_2 | 19 199 | 4 067 250 | 4 667 600 |
| β_3 | 4 829 | -1 537 125 | -1 129 200 |
| β_4 | | 259 875 | -1 858 120 |
| β_5 | | | 894 520 |
| Coefficient Denominator | 240 000 | 9 000 000 | 14 400 000 |

BIBLIOGRAPHY

1. Cryer, C. W., A New Class of Highly Stable Methods: A_0 -Stable Methods, BIT, 13, 153-159 (1973).
2. Dill, C., and C. W. Gear, A Graphical Search for Stiffly Stable Methods for Ordinary Differential Equations, J. Assoc. Comput. Mach., 18, 75-79 (1971).
3. Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, N.J., 1971.
4. Grigorieff, R. D., and J. Schroll, Über $A(\alpha)$ -Stabile Verfahren hoher Konsistenzordnung, Computing, 20, 343-350 (1978).
5. Gupta, G. K., Some New High Order Multistep Formulae for Solving Stiff Equations, Math. Comp., 30, 417-432 (1976).
6. Gupta, G. K., and C. S. Wallace, Some New Multistep Methods for Solving Ordinary Differential Equations, Math. Comp., 29, 489-500 (1975).
7. Kong, A. K., A Search for Better Linear Multistep Methods for Stiff Problems, Dissertation, University of Illinois at Urbana-Champaign, 1977.
8. Lambert, J. D., The Initial Value Problem for Ordinary Differential Equations, In The State of the Art in Numerical Analysis (ed. by D. Jacobs), Academic Press, 1977.
9. Nolting, D. J., Linear Multistep Methods with Near-Optimal Stability, Dissertation, University of Missouri at Columbia, 1979.
10. Thompson, S., and D. J. Rodabaugh, Corrector Methods with Increased Ranges of Stability, Comput. and Math. with App., 3, 197-201 (1977).

11. Wallace, C. S., and G. K. Gupta, General Linear Multistep Methods to Solve Ordinary Differential Equations, Austral. Comput. J., 5, 62-69 (1973).
12. Widlund, O. B., A Note on Unconditionally Stable Linear Multistep Methods, BIT, 7, 65-70 (1967).

P-stable Hybrid Schemes for Initial Value Problems

By

Simeon Ola Fatunla
School of Mathematics,
Trinity College,
Dublin 2.
Ireland

AMS(MOS) classification (1970)

Primary 65L05, Secondary 65O30

Keywords and Phrases: p -stability, Symmetric hybrid
formulae, periodic initial value problems.

*This research work was carried out while the author is on
sabbatical leave from University of Benin, Benin-City,
Nigeria.

January 1982

ABSTRACT:

Henrici (1962) discussed optimal Stormer Cowell class of linear multistep methods for the second order differential equations $y'' = f(x,y)$. Lambert and Watson (1976) established that such formulae are orbitally unstable and developed symmetric Lmm which annihilates the instability. Dahlquist (1978) proved a barrier theorem that the order of accuracy of a P-stable Lmm cannot exceed two. Hairer (1979) generalized these concepts. Fatunla (1981) proposed symmetric hybrid schemes well suited to periodic initial value problems. A generalization of this idea is proposed and a k -stable scheme of order six is realized. The new algorithm compares favourably with the existing schemes for periodic initial value problems i.e. Stiefel and Bettis (1969), Lambert and Watson (1975), Hairer (1979), Fatunla (1981)

Introduction:

Henrici [4] and Lambert [5] discussed the direct application of the Linear Multistep scheme

$$\rho(E)y_n = h^2 \sigma(E)f_n \quad (1.1)$$

to the second order initial value problem

$$y'' = f(x,y), \quad y(a) = \eta, \quad y'(a) = \delta \quad (1.2)$$

$x \in R$ and y, f, η, δ are m -vectors with the assumption that (1.2) satisfies the existence theorem.

$\rho(E)$ and $\sigma(E)$ are respectively first and second characteristic polynomials given as

$$\rho(E) = \sum_{j=0}^k \alpha_j E^j, \quad (1.3)$$

$$\sigma(E) = \sum_{j=0}^k \beta_j E^j, \quad (1.4)$$

and E is the shift operator such that

$$E^j y_n = y_{n+j} \quad (1.5)$$

where

$$y_{n+j} \equiv y(x_{n+j})$$

defined on the discrete point set

$$\{x_n | x_n = a + nh, n = 0, 1, \dots\}.$$

The initial value problem (1.2) in general possesses periodic or oscillatory solutions and an important subclass of the Lmm (1.1) is the Störmer-Cowell formulae whose first characteristic polynomial is simply

$$\rho(E) = E^2 - 2E + 1 \quad (1.6)$$

with the optimal 2-step scheme given by Numerov Scheme, whereby

$$\sigma(E) = \frac{1}{12}(E^2 + 10E + 1) \quad (1.7)$$

In order to explain the concept of P-stability, we consider the application of (1.1) to the scalar test problem (1.2) with

$$f(x,y) = -\omega^2 y, \quad \omega \in \mathbb{R} \quad (1.8)$$

leading to a linear difference equation

$$(\rho(E) + \Omega^2 \sigma(E)) y_n = 0, \quad (1.9)$$

($\Omega = \omega h$) whose characteristic polynomial is

$$\pi(r, \Omega) = \rho(r) + \Omega^2 \sigma(r) \quad (1.10)$$

which possesses a general solution

$$y_n = \sum_{r=1}^k \gamma_r \xi_r^n \quad (1.11)$$

with ξ_r being the roots of $\pi(r, \Omega)$.

Definition:

The Lmm (1.1) is said to be orbitally stable if there exists Ω_0^2 such that for an $\Omega^2 \in (0, \Omega_0^2)$, the principal roots ξ_1 and ξ_2 lie on the unit circle while the spurious roots ξ_3, \dots, ξ_k lie inside the unit circle. Dahlquist [1] established that for the Lmm (1.1) to be unconditionally stable, the order of accuracy cannot exceed 2 and besides, among the unconditionally stable formulas, the scheme of Richtmyer and Morton [7, pp. 263] given by

$$\rho(E) = E^2 - 2E + 1, \quad \sigma(E) = \frac{1}{4}(E^2 + 2E + 1) \quad (1.12)$$

has the minimum local error.

Definition 2 (Lambert and Watson [6]).

The method (1.1) is said to have an interval of periodicity $(0, \Omega_0^2)$ provided that for all $\Omega^2 \in (0, \Omega_0^2)$, the roots ξ_s of $\pi(r, \Omega^2)$ defined by (1.10), satisfy

$$\xi_1 = e^{i\theta(\Omega)}, \quad \xi_2 = e^{-i\theta(\Omega)},$$

$$|\xi_s| < 1, \quad s = 3(1)k$$

where $\theta(\Omega) \in \mathbb{R}$.

Lambert and Watson developed symmetric multistep methods with non-vanishing interval of periodicity; a desirable property deficient in the Störmer Cowell schemes with stepnumber greater than two.

Recently Fatunla [2] proposed symmetric hybrid methods of order six which possess non-vanishing intervals of periodicity. The resultant algorithms denoted by

$P_{\frac{1}{2}}^H E P_{\frac{3}{2}}^H E P_2 E C^H E$ and specified as follows:

$$P_{\frac{1}{2}}^H : y_{n+\frac{1}{2}} = \frac{1}{2}y_n + \frac{1}{2}y_{n+1} - \frac{h^2}{16}(f_n + f_{n+1})$$

$$E : f_{n+\frac{1}{2}} = f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$$

$$P_{\frac{3}{2}}^H : y_{n+\frac{3}{2}} = -\frac{1}{2}y_n + \frac{3}{2}y_{n+1} + \frac{h^2}{16}(f_n + 5f_{n+1})$$

$$E : f_{n+\frac{3}{2}} = f(x_{n+\frac{3}{2}}, y_{n+\frac{3}{2}})$$

$$P_2^{[0]} : y_{n+2}^{[0]} = 2y_{n+1} - y_n + h^2 f_{n+1} \quad (1.14)$$

$$E : f_{n+2}^{[0]} = f(x_{n+2}, y_{n+2}^{[0]})$$

$$C_2^H : y_{n+2}^{[1]} = 2y_{n+1} - y_n + \frac{h^2}{60}(f_{n+2}^{[0]} + 16f_{n+\frac{3}{2}} + 26f_{n+1} + 16f_{n+\frac{1}{2}} + f_n)$$

$$E : f_{n+2}^{[1]} = f(x_{n+2}, y_{n+2}^{[1]})$$

are not P-stable.

In this paper, we extend the concept of Fatunla [2] to derive P-stable hybrid methods also of order six. With a choice of a parameter r , $0 < r < 1$, the development of the new algorithms are based on the following formulae:

Method I

$$P_1^H : y_{n+r} + \alpha_{10}y_n + \alpha_{11}y_{n+1} = h^2(\beta_{10}f_n + \beta_{11}f_{n+1})$$

$$E : f_{n+r} = f(x_{n+r}, y_{n+r})$$

$$P_2^H : y_{n+2-r} + \alpha_{20}y_n + \alpha_{21}y_{n+1} = h^2(\beta_{20}f_n + \beta_{21}f_{n+1})$$

$$E : f_{n+2-r} = f(x_{n+2-r}, y_{n+2-r}) \quad (1.15)$$

$$P_2 : y_{n+2}^{[0]} + \alpha_{30}y_n + \alpha_{31}y_{n+1} = h^2(\beta_{30}f_n + \beta_{31}f_{n+1})$$

$$E : f_{n+2}^{[0]} = f(x_{n+2}, y_{n+2}^{[0]})$$

$$C^H : y_{n+2} + \alpha_{40}y_n + \alpha_{41}y_{n+1} = h^2(\beta_{40}f_n + \beta_{41}f_{n+1}$$

$$+ \beta_{42}f_{n+2}^{[0]} + \gamma_{41}f_{n+r} + \gamma_{42}f_{n+2-r})$$

$$E : f_{n+2}^{[1]} = f(x_{n+2}, y_{n+2}^{[1]})$$

Other alternatives are methods II, method III and method IV whereby the predictor P_2 are respectively replaced by the following predictors:

$$\bar{P}_2^H : y_{n+2}^{[0]} + \alpha_{30}y_n + \alpha_{31}y_{n+1} = h^2(\beta_{31}f_{n+r} + \gamma_{32}f_{n+2-r})$$

$$\bar{P}_2^H : y_{n+2}^{[0]} + \alpha_{30}y_n + \alpha_{31}y_{n+1} = h^2(\beta_{30}f_n + \beta_{31}f_{n+1} + \gamma_{31}f_{n+r})$$

$$\bar{P}_2^H : y_{n+2}^{[0]} + \alpha_{30}y_n + \alpha_{31}y_{n+1} = h^2(\beta_{30}f_n + \beta_{31}f_{n+1} + \gamma_{32}f_{n+2-r})$$

In section 2, we shall discuss the development of the P-stable formulae while section 3 deals with the effective solution of the iteration scheme. The P-stability of the symmetric scheme is discussed in section 4 while section 5 is concerned with the numerical experiment and comparison with the existing schemes.

2. Development of Integration Formulae:

In this section the coefficients of the integration formulae (1.15) are obtained in terms of parameter r , $0 < r < 1$. This is illustrated with the integration formula for the first hybrid step i.e.

$$y_{n+r} + \alpha_{10}y_n + \alpha_{11}y_{n+1} = h^2(\beta_{10}f_n + \beta_{11}f_{n+1}) \dots \dots (2.1)$$

while the other formulae are simply stated.

We now associate with (2.1) the linear difference operator $L_1[y(x), h]$ defined as follows:

$$L_1[y(x_n), h] = y(x_{n+r}) + \alpha_{10}y(x_n) + \alpha_{11}y(x_{n+1}) - h^2(\beta_{10}y'(x_n) + \beta_{11}y'(x_{n+1})). \quad (2.2)$$

If we now ensure that $y(x)$ is the theoretical solution to (1.1) and obtain the Taylor series of terms on the rhs of (2.2) to obtain

$$\begin{aligned} C_{10} &= 1 + \alpha_{10} + \alpha_{11} \\ C_{11} &= r + \alpha_{11} \\ C_{12} &= \frac{1}{2}(r^2 + \alpha_{11}) - (\beta_{10} + \beta_{11}) \\ C_{13} &= \frac{1}{6}(r^3 + \alpha_{11}) - \beta_{11} \\ C_{14} &= \frac{1}{24}(r^4 + \alpha_{11}) - \beta_{11} \end{aligned} \quad (2.3)$$

In general we have

$$C_{1j} = \frac{1}{j!}(r^j + \alpha_{11}) - \beta_{11}, \quad j = 3, 4, 5, \dots$$

With the desire to obtain the highest possible order for the integration formula (2.1), the first four equations in (2.3) are allowed to vanish and the resultant system of linear equations has the following solution:

$$\begin{aligned} \alpha_{10} &= r - 1 \\ \alpha_{11} &= -r \end{aligned} \quad (2.4)$$

$$\beta_{10} = \frac{-r(r-1)(r-1)}{6} \quad (2.4)$$

$$\beta_{11} = r(r-1)(r+1)$$

thus yielding the integration formula:

$$y_{n+r} - ry_{n+1} + (r-1)y_n = \frac{h^2}{6} [-r(r-1)(r-2)f_n + r(r-1)(r+1)f_{n+1}] \quad (2.5)$$

with the error constant obtained from last equation in (2.3) as

$$C_{14} = \frac{r(r-1)}{24}(r^2-r-1). \quad (2.6)$$

Using identical arguments, the integration formula for the second hybrid step can be obtained as:

$$\begin{aligned} y_{n+2-r} + (r-2)y_{n+1} - (r-1)y_n \\ = \frac{h^2}{6} [-(r-1)(r-2)(r-3)f_{n+1} + r(r-1)(r-2)f_n] \end{aligned} \quad (2.7)$$

with error constant

$$C_{24} = \frac{1}{24} [(r-2)(-(2-r)^3 + 1) + 2(r-1)(r-2)(r-3)] \quad (2.8)$$

Similarly, the predictor for method I is obtained as

$$y_{n+2}^{[0]} - 2y_{n+1} + y_n = h^2 f_{n+1} \quad (2.9)$$

and error constant

$$C_{34} = \frac{1}{12}$$

Finally the hybrid corrector is obtained as

$$\begin{aligned}
y_{n+2} - 2y_{n+1} + y_n &= h^2 \left[\frac{1}{12} + \frac{1}{20r(r-2)} \right] (f_{n+2} + f_n) \\
&+ \left[\frac{5}{6} - \frac{1}{10(r-1)^2} \right] f_{n+1} - \frac{1}{20r(r-1)^2(r-2)} (f_{n+r} + f_{n+2-r})
\end{aligned} \tag{2.10}$$

with error constant C_{48}

$$C_{48} = \frac{1}{8!} \left\{ 254 - 8 \left[\frac{37}{6} + \frac{64(r-1)^2 - 2r(r-2) - (r^6 + (2-r)^6)}{20r(r-1)^2(r-2)} \right] \right\}. \tag{2.11}$$

While the two hybrid formulae (2.5) and (2.7) as well as the predictor formula (2.9) are of order 2, the hybrid corrector formula (2.11) is of order six. The corrector formula (2.11) being implicit has to be solved iteratively and this is discussed in the next section.

3. Solution Process for Corrector Formula (2.11)

With a starting value $y_{n+2}^{[0]}$ given by formula (2.9), the corrector formula (2.11) is solved iteratively as

$$\begin{aligned}
y_{n+2}^{[s+1]} &= 2y_{n+1} + y_n + h^2 \left[Af_n + Bf_{n+1} + C(f_{n+r} + f_{n+2-r}) \right] \\
&+ h^2 Af(x_{n+2}, y_{n+2}^{[s]}),
\end{aligned} \tag{3.1}$$

for $s = 0, 1, 2, \dots$

where

$$A = \beta_{40} = \beta_{42} \tag{3.2}$$

$$= \frac{1}{12} + \frac{1}{20r(r-2)} \tag{3.2}$$

$$B = \beta_{41}$$

$$= \frac{5}{6} - \frac{1}{10(r-1)^2} \tag{3.3}$$

and

$$C = \gamma_{41} = \gamma_{42}$$

$$= \frac{1}{20r(r-1)^2(r-2)} \quad (3.4)$$

The convergence of (3.1) is assured provided

$$h^2 \|A \frac{\partial f}{\partial y}\|_{\infty} < 1, \quad (3.5)$$

$\|\cdot\|_{\infty}$ is the maximum norm.

Unfortunately, for highly oscillatory systems of the form (1.2), we have that

$$\|\frac{\partial f}{\partial y}\|_{\infty} \gg 1 \quad (3.6)$$

and thus impose a severe constraint on mesh-size h . This constraint could be averted with the introduction of the Newton Raphson procedure into (3.1) i.e. wish to solve the nonlinear equation

$$F(y_{n+2}) = y_{n+2} - h^2 A f(x_{n+2}, y_{n+2}) - 2y_{n+1} + y_n$$

$$- h^2 [A f_n + B f_{n+1} + C(f_{n+r} + f_{n+2-r})]$$

$$= 0. \quad (3.7)$$

The application of Newton Raphson Scheme to (3.7) yields

$$y_{n+2}^{[s+1]} = y_{n+2}^{[s]} - [I - h^2 A \frac{\partial f}{\partial y}(x_{n+2}, y_{n+2}^{[s]})]^{-1} F(y_{n+2}^{[s]}),$$

$$(3.8)$$

$$s = 0, 1, 2, \dots$$

and this invariably converges in one or two iterations:

We confine our numerical experiments in section 5 to only

one iteration.

4. P-Stability Consideration and Determination of Parameter r

The application of the integration formulae (2.5), (2.7) and (2.11) to the scalar test problem

$$y'' = \omega^2 y, \quad \omega > 0 \quad (4.1)$$

respectively leads to the following difference equations:

$$y_{n+r} = ry_{n+1} - (r-1)y_n - \frac{\Omega^2}{6}[-r(r-1)(r-2)y_n + r(r-1)(r+1)y_{n+1}], \quad (4.2)$$

$$y_{n+2-r} = -(r-2)y_{n+1} + (r-1)y_n - \frac{\Omega^2}{6}[-(r-1)(r-2)(r-3)y_{n+1} + r(r-1)(r-2)y_n], \quad (4.3)$$

and

$$[1 + \Omega^2 A]y_{n+2} - [2 + B\Omega^2]y_{n+1} + [1 + \Omega^2 A]y_n + \Omega^2 C[y_{n+r} + y_{n+2-r}] \quad (4.4)$$

where A, B and C are specified by equations (3.2), (3.3) and (3.4).

The adoption of (4.2) and (4.3) in (4.4) gives the following second order difference equation:

$$R(\Omega)y_{n+2} + S(\Omega)y_{n+1} + R(\Omega); \quad (4.5)$$

where

$$R(\Omega) = \left[1 + \Omega^2 \left(\frac{1}{12} + \frac{1}{20r(r-2)}\right)\right], \quad (4.6)$$

and

$$s(\Omega) = -2 + \left[\frac{5}{6} + \frac{1}{10(r-1)^2} - \frac{1}{10(r-1)^2(r-2)} + \frac{1}{10r(r-1)^2} \right] \Omega^2 + \frac{1}{60} \left[\frac{r+1}{(r-1)(r-2)} - \frac{r-3}{r(r-1)} \right] \Omega^4 \quad (4.7)$$

The characteristic equation associated with (4.5) is given as

$$R(\Omega)r^2 + S(\Omega)r + R(\Omega) = 0$$

which the bilinear transformation

$$r = \frac{1+z}{1-z} \quad (4.8)$$

reduces to

$$[2R(\Omega) - S(\Omega)]z^2 + [2R(\Omega) + S(\Omega)] = 0 \quad (4.9)$$

The roots of (4.9) are purely imaginary and lie on the unit circle provided

$$\frac{2R(\Omega) + S(\Omega)}{2R(\Omega) - S(\Omega)} = 1 \quad (4.10)$$

i.e.

$$S(\Omega) = 0 \quad (4.11)$$

This implies

$$-2 + \left[\frac{5}{6} + \frac{1}{10(r-1)^2} - \frac{1}{10(r-1)^2(r-2)} + \frac{1}{10r(r-1)^2} \right] \Omega^2 + \frac{1}{60} \left[\frac{r+1}{(r-1)(r-2)} - \frac{r-2}{r(r-1)} \right] \Omega^4 = 0 \quad (4.12)$$

At $\Omega = \infty$, equation (4.12) reduces to

$$\frac{r+1}{(r-1)(r-2)} - \frac{r-2}{r(r-1)} = 0 \quad (4.13)$$

which gives

$$r = \frac{4}{5} \quad (4.14)$$

Hence, the adoption of (4.14) in the integration formula (3.8) leads to a P-stable scheme.

5. Numerical Experiments

We first consider the nearly periodic initial value problem of Stiefel and Bettis [8]

$$z'' + z = 0.001e^{ix}, \quad z \in C \quad (5.1)$$

$$z(0) = 1, \quad z'(0) = 0.9995i$$

whose theoretical solution is given as

$$\begin{aligned} Z(x) &= u(x) + iv(x), \\ U(x) &= \cos x + 0.0005x \sin x, \\ V(x) &= \sin x - 0.0005x \cos x. \end{aligned} \quad (5.2)$$

(5.2) denotes a motion on a perturbed circular orbit which spirals outwards at a distance

$$\gamma(x) = \sqrt{u^2(x) + v^2(x)} \quad (5.3)$$

from the origin.

Problem (5.1) was solved numerically in the interval $0 < x < 40\pi$ adopting uniform mesh-sizes

$$h = \left\{ \frac{\pi}{12}, \frac{\pi}{9}, \frac{\pi}{6}, \frac{\pi}{5}, \frac{\pi}{4} \right\}$$

in the P_r^H E P_{2-r}^H E P_2 E C^H E mode with $r = \frac{4}{5}$ considering the cases:

$x = 40\pi$ $\gamma(x) = 1.001972$

T A B L E 1

| h | Stormer-Cowell [8] | Lambert & Shaw [6] | Fatunla [2] | P-stable scheme (3.8) |
|----------|-----------------------|-----------------------|----------------|--------------------------|
| $\pi/4$ | 0.965645 | 1.003067 | 1.003021 | 1.002132 |
| $\pi/5$ | 0.993734 | 1.002217 | 1.002421 | 1.002033 |
| $\pi/6$ | 0.999596 | 1.002047 | 1.002180 | 1.002000 |
| $\pi/9$ | 1.001829 | 1.001978 | 1.002020 | 1.001977 |
| $\pi/12$ | 1.001953 | 1.001973 | 1.001991 | 1.001973 |

TABLE 2

| h | Stormer-Cowell [8] | $\gamma(x) = 1.001972$ | $10^6 \times (\gamma(x) - \gamma_{True})$ | | | P-stable Scheme (3.8) |
|----------|--------------------|------------------------|---|-------------|-----|-----------------------|
| | | | Lambert & Watson [6] | Fatunla [2] | | |
| $\pi/4$ | -36327 | | 1095 | 1049 | 160 | |
| $\pi/5$ | -8238 | | 245 | 339 | 61 | |
| $\pi/6$ | -2376 | | 75 | 208 | 28 | |
| $\pi/9$ | -143 | | 6 | 48 | 5 | |
| $\pi/12$ | -19 | | 1 | 19 | 2 | |

- (a) without incorporating Newton Raphson i.e.
in formula (2.10)
- (b) incorporating the Newton Raphson scheme
i.e. formula (3.8)

Tables 1 and 2 give a detailed comparison of the proposed scheme

- (i) five step Störmer Cowell scheme of order six [8]
(ii) Symmetric multistep methods of order 6 [6]

While the proposed scheme whose corrector is also of order six compare favourably with the symmetric multistep scheme of Lambert and Watson [6], they are both superior to the Störmer Cowell Methods [8].

We finally consider the example proposed by Lambert and Watson [6] and designed to illustrate P-stability:

$$\begin{aligned} y_1'' + \omega^2 y_1 &= f''(x) + \omega^2 f(x); \\ y_1(0) &= a + f(0), \quad y_1'(0) = f'(0) \end{aligned} \quad (5.4)$$

$$\begin{aligned} y_2'' + \omega^2 y_2 &= f''(x) + \omega^2 f(x); \\ y_2(0) &= f(0), \quad y_2'(0) = \omega a + f'(0) . \end{aligned}$$

The theoretical solution to (5.4) is given as

$$\begin{aligned} y_1(x) &= a \cos \omega x + f(x) \\ y_2(x) &= a \sin \omega x + f(x) \end{aligned} \quad (5.5)$$

with

$$f(x) = e^{-0.05x} \quad (5.6)$$

and three different values of parameter a are adopted i.e.

$$a = \{0, 0.1, 0.2\}$$

The numerical integration scheme was applied to ivp (5.4) in the interval $0 < x < 20\pi$ using a uniform mesh size $h = \frac{\pi}{32}$.

The same problem was solved by Lambert and Watson in [6] using

- (i) Numerov Method (1.6 - 1.7)
- (ii) Symmetric Multistep Scheme $p = 2, C = -\frac{5}{12}$
- (iii) Symmetric Multistep scheme $p = 2, C = -\frac{1}{4}$

While the Numerov Scheme is not P-stable, both the proposed scheme as well as symmetric scheme of Lambert and Watson [6] are P-stable. Table 3 gives the error E in the radius

$$R = \sqrt{y_1^2 + y_2^2}$$

at $x = 20\pi$.

R E F E R E N C E S

1. G. Dahlquist, "On accuracy and unconditional stability of linear multistep methods for second order differential equations", BIT (1978) 18, 133-136
2. S.O. Fatunla, "Symmetric Hybrid methods for periodic initial value problems", Manuscript (1981).
3. E. Hairer, "Unconditionally stable methods for second order differential equations", Numer. Maths. (1979), 32, 373-379.
4. P. Henrici, Discrete Variable methods in ordinary differential equations (1962), John Miley and Sons.
5. J.D. Lambert, Computational methods in ordinary differential equations (1973), John Miley and Sons.
6. J.D. Lambert and I.A. Watson, "Symmetric multistep methods for periodic initial value problems", Jour: Inst. Maths. Applics. (1976) 18, 189-202.
7. R. Richtmyer and K.W. Morton, "Difference Methods for initial Value Problems, second edition (1967) Interscience Publishers.
8. E. Stiefel and D.G. Bettis, "Stabilization of Cowell's method", Numer. Maths. (1969) 13, 154-175.

Some Comments on Stability and Error Analysis
for Stiff Nonlinear Problems

Germund Dahlquist
Royal Institute of Technology, Stockholm
and Stanford University

ABSTRACT

First some basic facts and notions concerning stiff nonlinear ODE's will be presented, partly by examples. Techniques for deriving stability properties and error bounds for certain classes of nonlinear systems will be described. Some of the techniques are applicable to variable step size.

If stability can be proved for a system with a Jacobian $J(t)$, when the step size is $h(t)$, then a modified stability result is valid for the system with Jacobian $J(t) + \Delta J(t)$, if (in the case of one-leg multistep methods),

$$\int \| (M + \Delta M) \cdot \Delta J \cdot M \| dt$$

is reasonably bounded, where $M = (I - hJ\beta_k/\alpha_k)^{-1}$.

Results are known for monotonic systems, i.e., when the Jacobian has a negative logarithmic norm, relative to some inner-product norm, and for systems with a negative diagonal-dominant Jacobian, see e.g., Kreiss. The generalized contractivity analysis for one-leg multistep methods will be summarized, see also Dahlquist 1978, Söderlind and Dahlquist 1981. In the latter paper improved results are given for systems of singular perturbation type, also covering the case when different methods are used for different subsystems. Improved results for monotonic systems, when the time-derivative of the Jacobian is bounded, are given by Söderlind

1981. Contractivity analysis for linear multistep methods and one-leg methods, in maximum norm, is discussed by Nevanlinna and Liniger (1978) and Sand (1981), also for the case of variable step size. Hundsdorfer (1981) has extended the analysis of contractivity (B-stability) for Runge-Kutta methods, given by Butcher (1975) and Burrage and Butcher (1979), to a Rosenbrock method. Dahlquist and Jeltsch studied generalized contractivity properties of Runge-Kutta methods, which need not be A-stable.*

Much remains to be done, in particular concerning the following things:

i) instabilities, which may be caused by the change of step size and order,

ii) the practical verification during the computation that the assumptions of the theory are satisfied at the point and with the step size proposed,

iii) the automatic detection, if one of the eigenvalues of $hJ(t)$ with large modulus approaches the imaginary axis (in particular the origin).

A sketch of an asymptotics for variable step size has recently been worked out by Dahlquist (1981, rep. 8110).

*Nevanlinna and Odeh (1981) have recently developed a very powerful technique for nonlinear stability analysis for fixed step size.

REFERENCES

- Burrage, K. and Butcher, J.C. (1979) Stability Criteria for Implicit Runge-Kutta Methods, SIAM J. Numer. Anal. 16, 46-57.
- Butcher, J.C. (1975) A Stability Property of Implicit Runge-Kutta Methods, BIT 15, 358-361.
- Dahlquist, G. (1975) Error Analysis of a Class of Methods for Stiff Nonlinear Initial Value Problems, Numerical Analysis, Dundee, Springer Lecture Notes in Mathematics 506, 60-74.
- Dahlquist, G. (1978) G-Stability is Equivalent to A-Stability, BIT 18, 384-401.
- Dahlquist, G. (1981) On the Local and Global Errors of One-leg Methods, Report TRITA-NA-8110.
- Dahlquist, G. and Jeltsch, R. (1979) Generalized Disks of Contractivity for Explicit and Implicit Runge-Kutta Methods, Report TRITA-NA-7906.
- Dahlquist, G. and Söderlind, G. (1982) Some Problems Related to Stiff Nonlinear Differential Systems, Proc. of the 5th International Conference on Computing Methods in Applied Sciences and Engineering, INRIA, Versailles 1981, North-Holland Publ. Co.
- Hundsdoerfer, W.H. (1981) Nonlinear Stability Analysis, For a Simple Rosenbrock Method, Report No. 81-31, Institute of Applied Mathematics and Computer Science, University of Leiden, The Netherlands.
- Kreiss, H.O. (1978) Difference Methods for Stiff Ordinary Differential Equations, SIAM J. Numer. Anal. 15, 21-58.
- Nevanlinna, O. and Liniger, W. (1978) Contractive Methods for Stiff Differential Equations, Part I, BIT 18, 457-474, Part II, BIT 19, 53-72.
- Nevanlinna, O. and Odeh, F. (1981) Multiplier Techniques for Linear Multistep Methods, Numer. Funct. Anal. and Optimiz., 3, 377-423.
- Sand, J. (1981) On One-Leg and Linear Multistep Formulas with Variable Step Sizes, Report TRITA-NA-8112.
- Söderlind, G. (1981) Multiple-Step G-Contractivity With Applications to Slowly Varying Linear Systems, Report TRITA-NA-8107.
- Söderlind, G. and Dahlquist, G. (1981) Error Propagation in Stiff Differential Systems of Singular Perturbation Type, Report TRITA-NA-8108.

Contractivity of Multistep and One-leg Methods with Variable Steps

Werner Liniger
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Typed by: J. Genzano

Abstract: Some of the work on contractivity of multistep and one-leg methods for ordinary differential equations is surveyed. This theory provides sufficient conditions for stability of such methods as applied with variable steps to certain classes of nonlinear systems and equations with variable coefficients. It also identifies explicitly families of A_0 - and A -stable formulas which can be extended to variable steps in such a way that these stability properties are preserved for some of the problem classes mentioned above.

Acknowledgement

The author's research was sponsored in part by the Air Force Office of Scientific Research (AFSC), United States Air Force, under Contract No. F44620-77-C-0088. The United States Government is authorized to reproduce and distribute reports for government purposes notwithstanding any copyright notation hereon.

1. Introduction

A linear multistep formula,

$$\sum_{j=0}^k a_j x_{n-j} - h \sum_{j=0}^k b_j \dot{x}_{n-j} = 0, \quad (1.1)$$

normalized by

$$\sum_{j=0}^k b_j = 1, \quad (1.2)$$

is said to be *stable* at $q = h\lambda$, if all solutions $\{x_n\}$ of the difference equation

$$\sum_{j=0}^k (a_j - qb_j)x_{n-j} = 0 \quad (1.3)$$

generated by applying (1.1) to the "test equation"

$$\dot{x} = \lambda x, \lambda \text{ const., complex}, \quad (1.4)$$

on the uniform grid $\{t_n | t_n = nh, n = 0, 1, \dots\}$ are *bounded* for a given $h > 0$ as $n \rightarrow \infty$. The set of all q 's at which (1.1) is stable is called the *stability region* S of the *formula*. On the uniform grid and for constant formula coefficients, we have stability at q iff the "root condition" is satisfied.

In practical applications, multistep formulas are used to solve non-linear systems of differential equations, or linear equations with variable coefficients. Furthermore, they are normally implemented with variable steps, and under any or all of these circumstances it is in general hard to find *necessary and sufficient* conditions for asymptotic stability of the difference equation. The theory of contractivity, part of which is summarized in this paper, gives *sufficient* conditions for stability in some of these cases, notably a) for dissipative (monotone negative) nonlinear systems of differential

equations

$$\dot{x} = f(t, x), \quad (1.5)$$

i.e., systems satisfying [4,5]

$$\langle x - y, f(t, x) - f(t, y) \rangle \leq \mu |x - y| \quad (1.6)$$

for some scalar product \langle, \rangle and some $\mu \leq 0$, where $|x|^2 := \langle x, x \rangle$; and b) for the variable coefficient test equation

$$\dot{x} = \lambda(t)x, \quad \lambda(t) \text{ complex} \quad (1.7)$$

with arbitrary $\lambda(t)$ in the negative left half-plane or in some subset thereof. Furthermore, contractivity results can be gotten for variable (in some cases, arbitrary) step sequences $\{h_n\}$ as well as for constant steps. For variable steps, the formula usually has variable coefficients,*

$$\sum_{j=0}^k a_{j,n} x_{n-j} - h_n \sum_{j=0}^k b_{j,n} \dot{x}_{n-j} = 0, \quad (1.8)$$

normalized again by

$$\sum_{j=0}^k b_{j,n} = 1. \quad (1.9)$$

In general, however, stability results for variable steps are obtained only if the formula (1.8) is implemented as a *one-leg* (OL) method [4,5]

$$R_n x_n - h_n f(S_n t_n, S_n x_n) = 0, \quad (1.10)$$

* In some cases, it is advantageous to write the variable-step formulas in terms of a step other than the leading step h_n , as discussed in Section 3 hereafter.

where

$$R_n x_n := \sum_{j=0}^k a_{j,n} x_{n-j}, \quad S_n x_n := \sum_{j=0}^k b_{j,n} x_{n-j}, \quad (1.11)$$

rather than in the usual multistep (MS) form

$$R_n x_n - h_n S_n f(t_n, x_n) = 0. \quad (1.12)$$

In studying contractivity of the discrete solutions $\{x_n\}$ of the test equation (1.4) or (1.7) (respectively, the non-linear system (1.5)) we ask whether $\{x_n\}$ (respectively the difference $\{x_n\}$ of any two solutions $\{y_n\}$ and $\{y_n + x_n\}$) is *non-increasing* in an appropriate sense (rather than bounded as in discussing stability)? More precisely, we ask whether $\|X_{n-1}\| \geq \|X_n\|$ in some given norm $\|\cdot\|$ (independent of n), where $X_n := (x_{n-k+1}, x_{n-k+2}, \dots, x_n)$? If this is the case for all $n \geq k$, then $\|X_{k-1}\| \geq \|X_n\|$ and we have stability. When applied to the constant coefficient test equation (1.4) with fixed steps, both the MS- and the OL-implementations of (1.1) lead to the difference equation (1.3). In this case, the formula (1.1) is said [15] to be contractive at $q = h\lambda$ if $\{x_n\}$ is non-increasing in the above sense for any solution of (1.3) computed with constant step size h . The set of all q at which the formula is contractive (for the given norm) is called the *contractivity region*, K . Clearly, contractivity at q implies stability at q and thus $K \subseteq S$ for any norm. By analogy to the corresponding stability concepts, we say that a formula is A -contractive, A_0 -contractive, etc. if, respectively, the left half q -plane, the negative real q -axis, etc., are contained in K . Every one of these contractivity properties implies the corresponding stability property.

When applied to (1.4), both implementations of the variable-step formula (1.8) give

$$\sum_{j=0}^k (a_{j,n} - q_n b_{j,n}) x_{n-j} = 0, \quad (1.13)$$

where $q_n = h_n \lambda$. Note that $q_n \in K_n$ for all n is sufficient for stability, i.e., q need not be constant. When implemented as a OL-method and applied to the variable-coefficient test equation (1.7), the formula (1.8) still generates a difference equation of the form (1.13) but with $q_n = h_n \lambda(S_n, t_n)$, and again we have stability if $\{q_n\} \subseteq K_n$. By contrast, the MS-method applied to (1.7) produces a difference equation of a more complicated type,

$$\sum_{j=0}^k (a_{j,n} - q_{j,n} b_{j,n}) x_{n-j} = 0,$$

one who's contractivity depends on $(k + 1)$ complex parameters $q_{j,n} = h_n \lambda(t_{n-j})$, rather than on only one of them, and this is the case even for constant steps.

Most of the existing work on contractivity falls into one of two categories. In the first, contractivity in inner product norms of OL-solutions of dissipative nonlinear systems (G-stability) was introduced in [4,5] and further investigated in [6,7,9,11,13,14,15,18,19,20,22]. Contractivity of Runge-Kutta methods for the same problem class (B-stability) was studied in [3,8,17].

The second approach, the one which is mainly surveyed in this paper, is to investigate contractivity in the max norm or related norms for the test equation (1.4), respectively (1.7). For constant steps, contractivity in this sense was first studied in [1] for classical Adams multistep and Runge-Kutta methods applied to (1.7) with $\lambda(t) \leq 0$. In [15,20] many aspects of the contractivity theory were developed systematically; Section 2 hereafter mainly summarizes that work.

For non-uniform grids, contractivity of Backward Differentiation Methods in certain "polygonal" norms was analyzed in [2,16,21]. Some results on max-norm contractivity of variable-step Adams-type OL-methods were given in [20]. A-Contractivity was studied in [22] for the class of all second-order ($p = 2$) two-step ($k = 2$) formulas, a

two-parameter family. It was shown that, for any given ratio of the two steps, all formulas of this class which are A-contractive, in either the max norm or in constant inner product norms, are members of one and the same one-parameter sub-family of all $p = k = 2$ -formulas. The OL-implementations of these formulas are A-stable with arbitrary step sequences for any dissipative nonlinear system and for (1.7) with any $\lambda(t)$, $\text{Re } \lambda(t) \leq 0$. Finally it was shown in [25] that for any $k \geq 1$ and for arbitrary step sequences the set of all $p = 2$, k -step formulas which are A-contractive in the max norm represent a $(k - 1)$ -parameter sub-family of the $(2k - 2)$ -parameter family of all $p = 2$, k -step formulas. The OL-implementation of these formulas are A-stable with respect to (1.7) for any $\lambda(t)$, $\text{Re } \lambda(t) \leq 0$, for arbitrary variable steps. The A-contractivity work is summarized in Section 3 of this paper.

2. General Discussion of Contractivity

a) Description of the contractivity region.

The contractivity region K has a simple geometry. With respect to any norm, K is closed and connected by arcs of circles ([15], Th. 2.1). For explicit formulas, K is convex. In contrast to this, the stability region S in general consists of several disconnected components, as illustrated in [12] for a formula given in [10].

With respect to the max norm, we have ([15], Th. 3.1)

$$K = \{q \in \mathbb{C} \mid F(q) := \sum_{j=1}^k |a_j - qb_j| - |a_0 - qb_0| \leq 0\}. \quad (2.1)$$

The boundary of K is $\partial K = \{q \mid F(q) = 0\}$. It is smooth except possibly at its intersection points with the real axis. Many examples of contractivity curves are plotted in [12,20] in comparison with the corresponding root-locus curves.

For any Backward Differentiation Formula (defined by $b_0 = 1$, $b_1 = b_2 = \dots = b_k = 0$) the max-norm contractivity region is a circle [15] of center a_0 and radius $\sum_{j=1}^k |a_j|$. In inner product norms, all formulas have contractivity regions which are either half-planes or the inside or outside of circles [18].

b) Algebraic constraints for contractivity*.

It was shown ([15], Th. 3.2) that the formula (1.1) is contractive at $q = 0$ iff

$$a_0 > 0, a_j \leq 0, j = 1, \dots, k. \quad (2.2)$$

It is contractive at $q = \infty$ iff

$$\gamma := b_0 - \sum_{j=1}^k |b_j| \geq 0 \quad (2.3)$$

and A_∞ -contractive (strictly contractive at $q = \infty$) iff $\gamma > 0$. The formula is A_0 -contractive iff both (2.2) and (2.3) hold. (The corresponding statement for A_0 -stability is, of course, not true in general.) Finally, the formula (1.1) is A -contractive iff ([15], Th. 3.3) it is A_0 -contractive and

$$\sum_{j=1}^k [(a_j^2 + b_j^2 \eta) / (a_0^2 + b_0^2 \eta)]^{1/2} \leq 1, 0 \leq \eta \leq +\infty. \quad (2.4)$$

For a given η , the inequality (2.4) is necessary and sufficient for contractivity at $q = iy$, y real, with $\eta = y^2$.

* From here on, unless otherwise stated, contractivity is to be understood in the max-norm.

c) Existence of high-order contractive formulas

For any $\delta > 0$ and any $\psi < \pi/2$ there exist ([15], Th.4.1) $A(\alpha)$ -contractive formulas of Adams type (i.e., with $a_0 = 1, a_1 = -1, a_2 = \dots = a_k = 0$) and of any order of accuracy p such that γ defined by (2.3) satisfies* $\gamma \geq 1 - \delta$ and $\alpha \geq \psi$. It should be noted, however, that these formulas exist only for $k \geq k_{\min}(p)$ where $k_{\min}(p)$ is a rather rapidly increasing function (empirically, $k_{\min} \geq 3p$). Furthermore, formulas with $p > 2$ which are $A(\alpha)$ -contractive (and thus $A(\alpha)$ -stable) with α near $\pi/2$ cannot be accurate, i.e., they must have large error constants [23].

A sequence of $A(\alpha)$ -contractive formulas of Adams type and of orders $p \leq 6$ was given in [20]. Among them is the $p = k = 2$ -formula

$$x_n - x_{n-1} - \frac{h}{4}(3\dot{x}_n + \dot{x}_{n-2}) = 0, \quad (2.5)$$

which is characterized in Section 3 hereafter and which is A -stable and $A(\alpha)$ -contractive with $\alpha = 70.5^\circ$, and a formula with $p = 3$ and $k = 4$,

$$x_n - x_{n-1} - \frac{h}{12}(8\dot{x}_n + 5\dot{x}_{n-2} - \dot{x}_{n-4}) = 0, \quad (2.6)$$

which is $A(\alpha)$ -stable with $\alpha = 78^\circ$ and $A(\alpha)$ -contractive with $\alpha = 41^\circ$. In [20] variable-step extensions of (2.5) and (2.6) were defined uniquely by requiring that they, too, have $a_{2,n} = b_{1,n} = 0$, respectively $a_{2,n} = a_{3,n} = b_{1,n} = b_{3,n} = 0$, and it was shown that the variable-step extension of (2.5),

$$x_n - x_{n-1} - h_n \left[\frac{2+r_n}{2+2r_n} \dot{x}_n + \frac{r_n}{2+2r_n} \dot{x}_{n-2} \right] = 0 \quad (2.7)$$

* It is trivial to verify that, subject to (1.2), we have $\gamma \leq 1$.

remains A_0 -contractive for arbitrary step ratios $r_n := h_n/h_{n-1}$.

The familiar Backward Differentiation Formulas (BDF) with $p = k \geq 2$ are not contractive in the max-norm at $q = 0$ [1,15] (nor are they, therefore, A_0 -, $A(\alpha)$ -, or A -contractive in this norm). The two-step second-order BDF,

$$\frac{3}{2}x_n - \frac{4}{2}x_{n-1} + \frac{1}{2}x_{n-2} - h\dot{x}_n = 0, \quad (2.8)$$

is A -stable and thus [13] A -contractive in an inner product norm (G -stable). However, as shown in [22] and discussed in Section 3 hereafter, this contractivity does not naturally extend to variable grids (see also the remark at the end of Sub-section 3a hereafter).

d) Boundedness theorem for nonlinear systems.

In [20], Th. 5.1, a boundedness result was given for the discrete solution of a nonlinear $m \times m$ -system of the form

$$\dot{x} = A(t, x)x + d(t, x), \quad (2.9)$$

where $A = (a_{ij})$ with $a_{ii}(t, x) < 0$, $i, j = 1, \dots, m$, and $d(t, x)$ is bounded. A somewhat sharper result is the following [24]:

Assume $\{x_n\}$ is a discrete solution of (2.9), generated by a one-leg method which is both A_0 - and A_∞ -contractive in the max-norm, and let $B = (b_{ij})$ with $b_{ii} = -a_{ii}\gamma > 0$ and $b_{ij} = -(\sum_{j=0}^k |b_j|) \sup_{x,f} |a_{ij}(St_n, Sx_n)| \leq 0$. If we let $\xi = (\xi_i)$, $\delta = (\delta_i)$, $\xi_i := \|x_i\|_\infty$ and $\delta_i := \|d_i\|_\infty$ where $\|u\|_\infty$ denotes the max-norm of a scalar sequence $\{u_n\}$, then we have

$$B\xi \leq \delta, \quad \xi \geq 0, \quad \delta \geq 0 \quad (2.10)$$

and if B is a M -matrix (which is the case if A has a sufficient amount of diagonal dominance) then we have input-output stability,

$$\|\xi\|_{\infty} \leq \kappa \|\delta\|_{\infty}, \quad (2.11)$$

for some κ because B has a bounded inverse.

3. A-contractivity of multistep formulas with variable steps

a. A-contractivity of two-step formulas with variable steps

In [22] contractivity in inner product norms (G-stability) for dissipative nonlinear systems and A-contractivity in the max-norm were analyzed for the three-parameter family of all methods with $p = 1$ and $k = 2$ and/or the two-parameter family of all $p = k = 2$ -methods. We summarize the results of this theory.

For variable-step G-stability analysis it turns out to be useful to write the two-step formula

$$a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} - \hat{h}_n (b_0 \dot{x}_n + b_1 \dot{x}_{n-1} + b_2 \dot{x}_{n-2}) = 0 \quad (3.1)$$

in terms of the step

$$\hat{h}_n := a_0 h_n - a_2 h_{n-1}, \quad (3.2)$$

where $h_n = t_n - t_{n-1}$ and $h_{n-1} = t_{n-1} - t_{n-2}$. With this choice of the step, the coefficients of the consistent ($p = 1$) two-step formulas can be written in terms of three

parameters a, b, c as

$$\begin{aligned} a_0 &= \frac{1}{2}(c+1), & b_0 &= \frac{1}{4}[1+b+(a+c)], \\ a_1 &= -c, & b_1 &= \frac{1}{2}(1-b), \\ a_2 &= \frac{1}{2}(c-1), & b_2 &= \frac{1}{4}[1+b-(a+c)]. \end{aligned} \quad (3.3)$$

We characterize the non-uniformity of the steps by the grid-parameter

$$\varepsilon = \varepsilon_n := (h_n - h_{n-1}) / (h_n + h_{n-1}), \quad |\varepsilon| < 1, \quad (3.4)$$

where $\varepsilon \gtrless 0$ iff $h_n \gtrless h_{n-1}$. Then $h_n = \bar{h}_n(1 + \varepsilon)$, $h_{n-1} = \bar{h}_n(1 - \varepsilon)$, and $\hat{h}_n = \bar{h}_n(1 + \varepsilon c)$, where $\bar{h}_n := \frac{1}{2}(h_n + h_{n-1})$ is the average step. For equal steps ($\varepsilon = 0$), $\hat{h}_n = \bar{h}_n = h_n = h_{n-1}$. In general, ε is related to the step ratio $r = r_n := h_n/h_{n-1}$ via

$$r = (1 + \varepsilon)/(1 - \varepsilon) \iff \varepsilon = (r - 1)/(r + 1). \quad (3.5)$$

It is easy to verify that the formula whose coefficients are specified by (3.3) is second-order ($p = 2$) accurate iff

$$a = \varepsilon \left[\frac{1 - c^2}{1 + \varepsilon c} - b \right]. \quad (3.6)$$

Thus for $\varepsilon = 0$ we have $p = 2$ iff $a = 0$.

Global contractivity,

$$\|X_n\| \leq \|X_{k-1}\|, \quad n \geq k, \quad (3.7)$$

which implies A-stability, is assured for all discrete solutions of dissipative nonlinear systems generated by variable-step one-leg methods provided these methods are (locally) G-stable, i.e., $\|X_i\| \leq \|X_{i-1}\|$ for all i , $i = k, k+1, \dots, n$, in a G-norm $\|\cdot\|$ which is *constant* with respect to n , i.e., independent of the step changes. It was

shown [22] that the only second-order two-step formulas for which a constant G-norm exists are defined by

$$\begin{aligned} a(\epsilon) &= \epsilon^2 c(1 - c^2)/(1 + \epsilon c)^2, \\ b(\epsilon) &= (1 - c^2)/(1 + \epsilon c)^2, \\ c(\epsilon) &= c = \text{const.}, 0 \leq c \leq 1. \end{aligned} \quad (3.8)$$

They represent a one-parameter subfamily of the two-parameter family of all $p = k = 2$ -formulas defined by (3.3), subject to the constraint (3.6).

In analyzing A-contractivity in the max-norm for $k = 2$, the inequality (2.4) was squared twice to obtain an equivalent rational condition in the coefficients. Subject to the constraints for second-order accuracy and A_0 -contractivity, the latter turns out to define the *same* one-parameter subfamily (3.8) of the two-parameter family of all $p = k = 2$ -formulas as does the contractivity analysis in inner product norms (except that, from the viewpoint of A-contractivity in the max-norm $c(\epsilon)$ need not be constant).

In [15] the particular formula (2.5) was derived by minimizing, over all A_0 -contractive $p = k = 2$ -formulas, a bound for the global error generated by applying the OL-implementation of any such formula to $\dot{x} = \lambda(t)x$, $\lambda(t) \leq -a$, for some arbitrarily small $a > 0$. The objective function whose minimization led to (2.5) was $|c_3|/\gamma$, where c_3 is the error constant and γ is defined by (2.3). As an example of an A-contractive formula, we give the one minimizing the same objective function over all A-contractive $p = k = 2$ -formulas [15]:

$$\frac{5}{6}x_n - \frac{4}{6}x_{n-1} - \frac{1}{6}x_{n-2} - h\left(\frac{5}{9}\dot{x}_n + \frac{2}{9}\dot{x}_{n-1} + \frac{2}{9}\dot{x}_{n-2}\right) = 0. \quad (3.9)$$

The formula (3.9) is associated with $a = 0$, $b = \frac{2}{9}$ and $c = \frac{2}{3}$. Its variable-step

extension (3.1) is defined by $c = \frac{2}{3}$,

$$\hat{h}_n = \frac{5}{6}h_n + \frac{1}{6}h_{n-1}, \quad (3.10)$$

and by (3.3) with

$$\begin{aligned} a &= a(\epsilon_n) = 10\epsilon_n^2/[3(3+2\epsilon_n)^2], \\ b &= b(\epsilon_n) = 5/(3+2\epsilon_n)^2, \end{aligned} \quad (3.11)$$

and with ϵ_n defined by (3.4). The error constant of this variable-step formula (with respect to \hat{h}_n) can be computed from the general expression

$$c_3 = c_{3,n} = \frac{1}{6}(-u^3\alpha_0 + v^3\alpha_2) - \frac{1}{2}(u^2\beta_0 + v^2\beta_2) \quad (3.12)$$

which is valid for any $p = k = 2$ -formula; here

$$\begin{aligned} u &= u_n := h_{n-1}/\hat{h}_n = (1-\epsilon)/(1+\epsilon c), \\ v &= v_n := h_n/\hat{h}_n = (1+\epsilon)/(1+\epsilon c), \end{aligned} \quad (3.13)$$

and again $c = 2/3$ for (3.9).

For uniform steps, the $p = k = 2$ Backward Differentiation Formula (BDF) (2.8) is A-stable and thus G-contractive as stated above. However, it is not a member of the family (3.8) for $\epsilon = 0$ and its variable-step extension (normally defined by setting $b_{1,n} = b_{2,n} = 0$) is not contractive in any *fixed* G-norm. In fact, the BDF was shown [22, 24] to become unstable for problems of type (1.7) with marginally stable, oscillatory solutions of increasing period and for geometrically increasing step sequences with a fixed number of steps per period. For similar problems with decreasing period and steps, the BDF was stable but overdamped. By contrast, the OL-method associated with the variable-step version of (3.9) was not only stable for both problems but gave

a much more accurate amplitude response than the BDF.

b. A-contractivity of second-order multistep formulas with variable steps

Consider the k -step formula (1.8) written in terms of the leading step, h_n , and let

$$\theta_j = \theta_{j,n} := (t_n - t_{n-j})/h_n, \quad j = 0, 1, \dots, k,$$

so that $\theta_0 = 0$ and $\theta_1 = 1$ by definition. For $k = 2$, $\theta_2 = 2/(1 + \epsilon)$ where ϵ is defined by (3.4). For uniform grids, $\theta_j = j$. In terms of the weighted moments

$$\begin{aligned} A_m = A_{m,n} &:= \sum_{j=0}^k \theta_j^m (-a_j), \\ B_m = B_{m,n} &:= \sum_{j=0}^k \theta_j^m b_j, \end{aligned} \quad (3.15)$$

where m is any integer ≥ 0 , the constraints for p^{th} -order accuracy of (1.8) are [18]

$$A_0 = 0, \quad A_m = mB_{m-1}, \quad m = 1, \dots, p \quad (3.16)$$

and the error constant (with respect to h_n) is given by

$$c_{p+1} = \frac{(-1)^p}{(p+1)!} [A_{p+1} - (p+1)B_p]. \quad (3.17)$$

The squaring technique used in [22] to analyze max-norm A-contractivity for $k = 2$ is impractical for $k > 2$. But another approach given in [25] is applicable to arbitrary k : subject to (2.2) and for $a_j \neq 0$ one can write (2.4) in the form

$$F(\eta) := \sum_{j=0}^k a_j [1 + (b_j^2 \eta / a_j^2)]^{1/2} \geq 0. \quad (3.18)$$

By consistency, $F(\eta) = \frac{1}{2}F\eta + O(\eta^2)$, and thus

$$F := \sum_{j=0}^k (b_j^2/a_j) \geq 0 \quad (3.19)$$

is necessary for (3.18) to hold. But it is easy to prove that second-order accuracy implies $F \leq 0$. Thus at best, $F = 0$. For any given fixed a_j (satisfying the consistency relations), F is a quadratic function of the b_j which takes a unique maximum,

$$F_{\max} = 0, \quad (3.20)$$

identically in the a_j , for

$$b_j = (\theta_j - \frac{1}{2}A_2)(-a_j), \quad j = 0, 1, \dots, k. \quad (3.21)$$

Thus the formulas defined by (3.21) do satisfy the necessary condition (3.19) and for $a_j \neq 0$ no other formulas will.

One finally proves that, subject to A_0 -contractivity and second-order accuracy, (3.19) is also sufficient for (3.18) to hold, and thus for A -contractivity. This is done by squaring (3.18) once and by majoring the remaining irrational terms, which are geometric means of any two of the radicands, by the corresponding arithmetic means. The result is (3.19).

From the consistency relations and normalization, $A_0 = 0$ and $A_1 = B_0 = 1$, it follows that

$$\begin{aligned} a_0 &= 1 - \sum_{j=2}^k (\theta_j - 1)(-a_j), \\ a_1 &= - \left[1 - \sum_{j=2}^k \theta_j (-a_j) \right], \end{aligned} \quad (3.22)$$

and thus the contractivity condition (2.2) at $q = 0$ defines a simplex in the $(k - 1)$ -dimensional space of the parameters a_j , $j = 2, \dots, k$, whose vertices are the origin and the intersection points $\{-a_j = 1/\theta_j, a_i = 0, i \neq j, j = 2, \dots, k\}$ of the a_j -axes with the plane $a_1 = 0$. One immediately verifies that these extreme points represent, respectively, the Trapezoidal Rules with step lengths $t_n - t_{n-j} = \theta_j h_n$, $j = 1, \dots, k$.

From the above it follows that all $p = 2$, k -step formulas which are A-contractive in the max-norm are members of the $(k - 1)$ -parameter sub-family (3.20) of the $(2k - 2)$ -parameter family of all $p = k = 2$ -formulas, with the parameters $\{a_2, \dots, a_k\}$ representing any point in the above mentioned simplex. A particularly interesting case is $p = 2, k = 3$, in view of the fact that, for evaluating the local error of a second-order method, at least three backward data must be available at every integration step. In this case, the above construction provides a two-parameter family of A-contractive formulas for arbitrary step sequences.

References

1. W. Liniger, "Zur Stabilität der numerischen Integrationsmethoden für Differentialgleichungen," Doctoral Thesis, University of Lausanne (1957).
2. R. Brayton and C. Conley, "Some results on the stability and instability of the backward differentiation methods with non-uniform time-steps", *Topics in Numerical Analysis*, Proc. Royal Irish Acad. Conf., Academic Press, NY (1972), pp. 13-33.
3. J. Butcher, "A stability property of implicit Runge-Kutta methods", *BIT* 15 (1975) 358-361.
4. G. Dahlquist, "On stability and error analysis for stiff nonlinear problems", Report TRITA-NA-7508, Royal Inst. Tech. Stockholm (1975).
5. G. Dahlquist, "Error analysis for a class of methods for stiff non-linear initial value problems", *Numerical Analysis Dundee 1975*, Lecture Notes in Mathematics 506, Springer-Verlag, Berlin (1976), pp. 60-74.
6. W. Liniger and F. Odeh, "On Lyapunov stability of nonlinear multistep difference equations," Report RC 5900, IBM Research Center, Yorktown Heights, NY (March 1976).

7. O. Nevanlinna, "On error bounds for G-stable methods", *BIT* 16 (1976) 79-84.
8. G. Wanner, "A short proof of nonlinear A-stability", *BIT* 16 (1976) 226-227.
9. G. Dahlquist, "On the relationship of G-stability to other stability concepts for linear multistep methods," *Topics in Numerical Analysis III* (J. Miller, Ed.) Academic Press, London (1977) pp. 349-362.
10. R. Grigorieff and J. Schroll, "Ueber A(α)-stabile Verfahren hoher Konsistenzordnung," Report No. 34, Fachbereich Mathematik, Technische Universität Berlin (1977).
11. O. Nevanlinna, "On the numerical integration of nonlinear initial value problems by linear multistep methods", *BIT* 17 (1977) 58-71.
12. O. Nevanlinna and W. Liniger, "Contractive methods for stiff ordinary differential equations," Report RC 7122, IBM Research Center, Yorktown Heights, NY (1977).
13. G. Dahlquist, "G-stability is equivalent to A-stability", *BIT* 18 (1978) 384-401.
14. G. Dahlquist, "Positive functions and some applications to stability questions for numerical methods", *Recent Advances in Numerical Analysis* (C. de Boor and G. Golub, Eds.) Academic Press (1978) pp. 1-29.
15. O. Nevanlinna and W. Liniger, "Contractive methods for stiff ordinary differential equations. Part I," *BIT* 18 (1978) 457-474.
16. R. Brayton and C. Tong, "Stability of dynamical systems: A constructive approach", *IEEE Trans. Circuits and Systems CAS-26* (1979) 224-234.
17. K. Burrage and J. Butcher, "Stability criteria for implicit Runge-Kutta methods", *SIAM J. Numer. Anal.* 16 (1979) 46-57.
18. G. Dahlquist, "Some properties of linear multistep and one-leg methods for ordinary differential equations", Reports TRITA-NA-7904, Royal Inst. Tech. Stockholm, and UIUCDCS-R-79-963 (R. D. Skeel, Ed.), University of Illinois, Urbana, Ill. pp. 1-1/1-4 (1979).
19. G. Dahlquist, "Some contractivity questions for one-leg and linear multistep methods", Report TRITA-NA-7905, Royal Inst. Tech. Stockholm (1979).
20. O. Nevanlinna and W. Liniger, "Contractive methods for stiff differential equations. Part II," *BIT* 19 (1979) 53-72.
21. R. Brayton and C. Tong, "Constructive stability and asymptotic stability of dynamical systems," *IEEE Trans. Circuits and Systems CAS-27* (1980).
22. G. Dahlquist, W. Liniger and O. Nevanlinna, "Stability of two-step methods for variable integration steps", Report RC 8494, IBM Research Center, Yorktown Heights, NY (1980).

23. R. Jeltsch and O. Nevanlinna, "Lower bounds for the accuracy of linear multistep methods," Report No. 6, Inst. for Geometry and Practical Mathematics, Technical University Aachen, Aachen, Germany (1980).
24. F. Odeh and W. Liniger, "On A-stability of second-order two-step methods for uniform and variable steps," Proc. IEEE Intl. Conf. Circuits and Computers 1980 (N. B. Rabbat, Ed.) Vol. 1, pp. 123-126.
25. W. Liniger "A-contractivity of second-order multistep formulas with variable steps," Report RC 9281, IBM Research Center, Yorktown Heights, NY (1980).

A Survey of Runge-Kutta Methods

for the Numerical Integration of Stiff Differential Systems*

1. Introduction

A q-stage Runge-Kutta formula for the numerical integration of the initial value problem

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0, \quad y \in \mathbb{R}^S \quad (1.1)$$

can be written in the general form

$$y_{n+1} - y_n = h \sum_{i=1}^q b_i k_i \quad (1.2a)$$

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^q a_{ij} k_j), \quad 1 \leq i \leq q, \quad (1.2b)$$

where $h = x_{n+1} - x_n$.

Such formulae can be represented conveniently by the arrays

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1q} \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ c_q & a_{q1} & a_{q2} & \cdots & a_{qq} \\ \hline & b_1 & b_2 & \cdots & b_q \end{array} \equiv \frac{c|A}{b^T} \quad (1.5)$$

Classically all Runge-Kutta formulae were explicit, i.e. $a_{ij} = 0$ for $j \geq i$, but they became interesting for the numerical integration of stiff differential systems with the introduction of fully implicit Runge-Kutta formulae by Butcher (1964). Butcher showed that for all q there exist fully implicit q-stage Runge-Kutta formulae of order 2q and Ehle (1969) has shown that all of these maximal order formulae are A-stable. Thus, amongst the class of

*J.R. Cash
Department of Mathematics
Imperial College of Science and Technology

implicit Runge-Kutta formulae we have the possibility of deriving A-stable methods of arbitrarily high order whereas it is well known from a celebrated result of Dahlquist (1963) that the highest attainable order of an A-stable linear multistep method is limited to two. In addition to this, the pioneering work of J.C. Butcher (1963, 1965) has led to the development of a general algebraic framework which allows the investigation of quite high order Runge-Kutta formulae in a comparatively straightforward manner.

One of the main arguments against the use of fully implicit Runge-Kutta formulae was on the grounds of the amount of computational effort required to solve (1.2b) for the k_i . If (1.2) is applied to a stiff system of ordinary differential equations it is necessary to solve the resulting system of algebraic equations, defining the k_i , using some form of Newton iteration. The system of algebraic equations to be solved is of size qs and so a modified Newton iteration scheme, if applied in the most obvious and naive way, will for large s require the order of $\frac{1}{3} q^3 s^3$ multiplications to obtain the required solution. This is about q^3 times as much work as is required to solve the algebraic equations arising from the use of a linear multistep method. It follows that if these two classes of methods are compared on this basis, the immediate conclusion is to reject fully implicit Runge-Kutta formulae in favour of linear multistep methods. However, because of the inherent stability of some classes of implicit Runge-Kutta formulae, coupled with the desirability of having high order single step integration methods, there has been a large amount of research into the possibility of overcoming these computational difficulties. Broadly speaking, research into finding efficient high order

implicit Runge-Kutta methods has followed two main directions:

- (1) The investigation of the use of transformation methods to obtain a solution of the algebraic equations defining the k_i in an efficient manner.
- (2) The derivation of different classes of implicit Runge-Kutta formulae which do not call for the solution of a system of q s simultaneous algebraic equations.

In what follows we shall survey some of the recent results concerned with Runge-Kutta formulae. The literature on implicit Runge-Kutta formulae is very large and an attempt to cover it fully in a single chapter has proved impossible. Instead, what we have done is to gather together those results which we think will have a significant effect on the way in which Runge-Kutta codes will be developed in the near future.

2. Stability Requirements for Implicit Runge-Kutta Formulae

If the Runge-Kutta formula (1.2a,b) is applied with a fixed stepsize h to the scalar test equation

$$\frac{dy}{dx} = \lambda y, \quad \lambda \in \mathbb{C}, \operatorname{Re}(\lambda) < 0 \quad (2.1)$$

we obtain the relation

$$y_{n+1} = R(z)y_n, \quad z = h\lambda \quad (2.2a)$$

where

$$R(z) = 1 + zb^T(I - zA)^{-1}e, \quad e = \underbrace{[1, 1, \dots, 1]}_q^T, \quad (2.2b)$$

and where I is the $q \times q$ unit matrix.

Here $R(z)$ is a rational function of z with both the numerator and the denominator having maximum degree q . The complex number z lies inside the region of absolute stability of (1.2) if $|R(z)| \leq 1$. The first, and still most widely used, stability criterion for Runge-Kutta formulae applied to stiff differential systems is that of A-stability due to Dahlquist (1963).

Definition 1

A numerical method is said to be A-stable if its region of absolute stability contains the whole of the left-hand half-plane $\text{Re}(z) < 0$.

A stronger condition, which ensures that the method has the correct damping at $z = -\infty$, is that of L-stability due to Ehle (1969).

Definition 2

A Runge-Kutta formula is said to be L-stable if it is A-stable and, in addition,

$$\lim_{\text{Re}(z) \rightarrow -\infty} |R(z)| = 0. \quad (2.3)$$

The A-stability criterion was originally developed for linear multi-step methods but has been widely applied to Runge-Kutta formulae. Many results are known concerning the conditions under which the rational function $R(z)$ has modulus less than unity for all z in the complex left-hand half-plane. (Ehle (1969) calls such a property A-acceptability and if, in addition, (2.3) holds, then $R(z)$ is said to be L-acceptable.) In particular, it is known that if $R(z)$ is a diagonal of the Padé table for e^z , then $R(z)$ is A-acceptable

(Birkhoff and Varga (1965)). Furthermore, if $R(z)$ lies on one of the two subdiagonals then $R(z)$ is L-acceptable (Ehle (1969)). Runge-Kutta formulae which, when applied to (2.1), give rational approximations of this type, have been investigated by Axelsson (1969), Chipman (1971) and Ehle (1969). However, the investigation of the A-acceptability of rational approximations to e^z remains a difficult task although there have recently been some important advances in this area (see, for example, Nørsett (1975), Iserles (1979), Warner, Hairer and Nørsett (1978)). In particular, we mention the concept of order stars developed by Warner et al. which is fast becoming a very important tool in the theory of approximation.

Although the criterion of A-stability has proved to be a very valuable one, particularly in helping us to discard those methods which are not promising for integrating stiff systems, it has the obvious limitations that the model test equation (2.1) is linear, homogeneous and has constant coefficients. Several authors have given examples of stiff equations for which a particular A-stable method with a certain variable stepsize gives a violently unstable solution (see, for example, Stetter (1973, p. 181)). A quite widely used generalisation of A-stability is that of S-stability due to Prothero and Robinson (1974). This is associated with the inhomogeneous test equation

$$y' = g'(x) + \lambda(y - g(x)) . \quad (2.4)$$

Prothero and Robinson sought to categorise those one-step methods which give a stable numerical solution when applied with a step-

length h to any equation of the form (2.4) where λ is a complex constant with negative real part and where $g'(x)$ is any function that is defined and bounded in some interval $x \in [x_0, \bar{x}]$. They define the concept of S-stability in the following way.

Definition 3.

A one-step numerical method is said to be S-stable if, for a differential equation of the form (2.4) and for any real positive constant λ_0 , there exists a real positive constant h_0 such that

$$\left\| \frac{y_{n+1} - g(x_{n+1})}{y_n - g(x_n)} \right\| < 1$$

providing $y_n \neq g(x_n)$, for all $0 < h < h_0$ and all complex λ with $\text{Re}(-\lambda) \geq \lambda_0$, and $x_n, x_{n+1} \in [x_0, \bar{x}]$.

In addition an S-stable one-step method is said to be strongly S-stable if $\frac{y_{n+1} - g(x_{n+1})}{y_n - g(x_n)} \rightarrow 0$ as $\text{Re}(-\lambda) \rightarrow \infty$ for all $h > 0$ such that $x_n, x_{n+1} \in [x_0, \bar{x}]$. Clearly, the concept of S-stability reduces to that of A-stability in the case $g(x) \equiv 0$. For further comments on S-stability see Stetter (1975).

Although many of the Runge-Kutta formulae that have recently been proposed for the integration of stiff systems still seek to satisfy the conditions of A-stability or S-stability, the test equation (2.4) has the major limitation that it is linear. A significant step in overcoming this deficiency was made with the G-stability criterion of Dahlquist (1975) for multistep methods and the B-stability criterion of Butcher (1975) for Runge-Kutta methods. The concept of B-stability is concerned with the nonlinear test equation

$$\frac{dy}{dx} = f(y(x)) \quad (2.5)$$

and is defined by Butcher (1975) in the following way.

Definition 4

Let $y_{n-1}, y_n, \dots, z_{n-1}, z_n, \dots$ be two sequences of approximate solutions of (2.5) computed by an implicit Runge-Kutta method using the same stepsize h . Let $\langle \cdot, \cdot \rangle$ denote a scalar product on \mathbb{R}^N and $\|\cdot\|$ the corresponding norm. The method is defined to be B-stable if, for any f satisfying

$$\langle f(u) - f(v), u - v \rangle \leq 0 \quad \text{for all } u, v \in \mathbb{R}^N \quad (2.6)$$

it holds that $\|y_n - z_n\| \leq \|y_{n-1} - z_{n-1}\|$.

This condition was further extended by Burrage and Butcher (1979) to the nonlinear system

$$y' = f(x, y(x)), \quad f: \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N \quad (2.7)$$

satisfying the monotonicity condition

$$\langle f(x, y) - f(x, z), y - z \rangle \leq 0 \quad \text{for all } y, z \in \mathbb{R}^N$$

and all $x \in \mathbb{R}$. (2.8)

Associated with the test equation (2.7), Burrage and Butcher (1979) give the following definition of BN-stability.

Definition 5

Let $\{y_n\}, \{z_n\}$ be two sequences of approximations to the solution of (2.7) computed by (1.2a,b) using a fixed step h . If

$$\|y_n - z_n\| \leq \|y_{n-1} - z_{n-1}\| \quad (2.9)$$

then the method is said to be BN-stable.

Condition (2.9), which with slight modifications is equivalent to the concept of contractivity used by Nevalinna and Liniger (1978), is stronger than that of A-stability since A-stability requires only that $\|y_n - z_n\|$ is bounded as $n \rightarrow \infty$. Burrage and Butcher give the following sufficient condition for BN-stability:

Consider the quadratic form

$$Q(\theta_1, \theta_2, \dots, \theta_q) \equiv \sum_{i,j=1}^q m_{ij} \theta_i \theta_j \quad \text{where } m_{ij} = b_i a_{ij} + b_j a_{ji} - b_i b_j.$$

If the method (1.2a,b) is such that $b_i \geq 0$ for $1 \leq i \leq q$ and Q is non-negative, then (1.2) is BN-stable. This leads to the following definition of algebraic stability due to Burrage and Butcher (1979).

Definition 6

A Runge-Kutta method which is such that $b_i \geq 0$ and $\zeta^T M \zeta$, where $M \equiv \{m_{ij}\}$, is a non-negative form, is said to be algebraically stable.

This condition has been given independently by Crouzeix (1979). *The condition of BN-stability is equivalent to the A-contractivity condition of Nevalinna et al (1978)*

Dahlquist and Jeitsch (1979) *also* observe that it is necessary to have $b_i > 0$ for all i since if one of the b_i is zero we have $m_{ii} = 0$ and the non-negative definiteness of M implies that the i^{th} row of M must be zero and the method becomes reducible. Burrage and Butcher (1979) have shown that if the Runge-Kutta formula is non-confluent (i.e. c_1, c_2, \dots, c_q are distinct) then

algebraic stability \Leftrightarrow BN-stability.

A further interesting result has been proved by Hundsdorfer

AD-A122 170

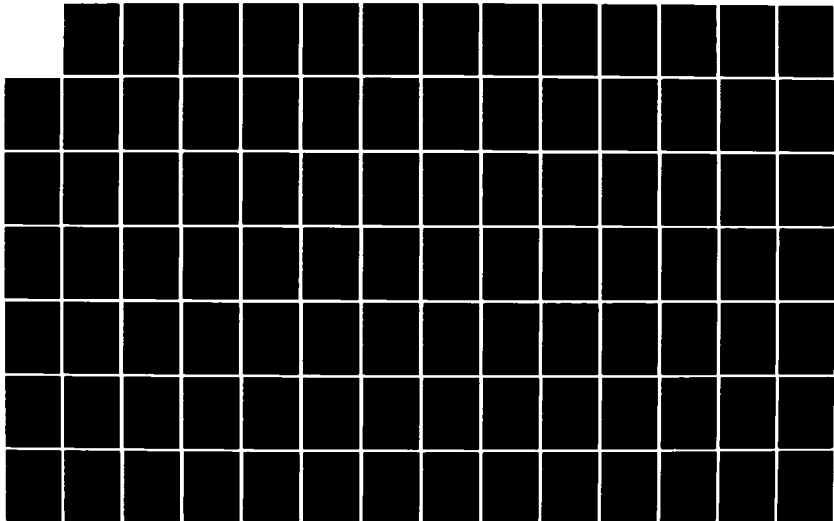
PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON STIFF
COMPUTATION APRIL 12. (U) UTAH UNIV SALT LAKE CITY DEPT
OF CHEMICAL ENGINEERING R C AIKEN 1982
AFOSR-TR-82-1036-VOL-2 AFOSR-82-0038

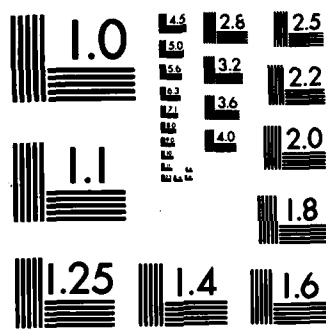
4/5

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Spijker (1981) who show that for any irreducible Runge-Kutta formula (see, for example, Stetter (1973)), which is not necessarily non-confluent, the concepts of B-, BN- and algebraic stability are equivalent.

A neat way of proving the B-stability of certain classes of Runge-Kutta formulae has been given by Warner (1976) who shows that the implicit Runge-Kutta formulae based on Gaussian points and Ehle's II_A methods based on Radau points are B-stable by using the well known fact that these particular Runge-Kutta formulae are equivalent to collocation methods (Wright (1970)). This technique has recently been extended by Nørsett and Warner (1981) to include other classes of Runge-Kutta formulae. The problem of classifying classes of algebraically stable Runge-Kutta formulae has been considered by Burrage (1978b). He shows that the q-stage methods of order $2q$ are all algebraically stable and has given a classification of all algebraically stable q-stage Runge-Kutta formulae of order $2q-2$ which are non-confluent. However, Burrage makes the point that such a classification is much harder for lower order formulae.

In conclusion we can say that in the past few years there has been considerable development in the stability analysis of Runge-Kutta formulae applied to nonlinear stiff differential systems. The condition of algebraic stability has two major advantages to recommend it over A-stability. Firstly, it seems likely that an algebraically stable Runge-Kutta method will be superior for nonlinear stiff systems to one which is only A-stable. Secondly, it is possible to give an algebraic condition for algebraic stability (i.e. one depending only on the coefficients of the Runge-Kutta method).

Because of this, algebraic stability, and the closely related concept of BN-stability, should play an important role in the future development of new Runge-Kutta formulae.

3. Fully Implicit Runge-Kutta Formulae

It was mentioned in the introduction that the main objection to implementing fully implicit Runge-Kutta formulae has traditionally been on the grounds of the amount of computational effort required to solve the nonlinear equations resulting from a modified Newton iteration. Algorithms to overcome this problem for a special class of Runge-Kutta formulae have been given by Butcher (1976) and for more general formulae, including the maximal order ones based on Gaussian points, by Bickart (1977) and Varah (1979). These algorithms are all based on transformation methods and in what follows we shall describe Butcher's approach since this has already been implemented in a successful computer package.

Consider the solution of the differential system

$$y'(x) = f(y(x)) \quad (3.1)$$

using the q-stage formula (1.2). The numerical solution at the point $x_n = x_{n-1} + h$ is computed using

$$y_n = y_{n-1} + h \sum_{j=1}^q b_j f(Y_j) \quad (3.2a)$$

where

$$Y_i = y_{n-1} + h \sum_{j=1}^q a_{ij} f(Y_j) \quad 1 \leq i \leq q. \quad (3.2b)$$

To evaluate Y_1, Y_2, \dots, Y_q satisfying (3.2b) we use a modified Newton iteration scheme so that, at the end of a correction iteration, Y_i is

replaced by $Y_i + \omega_i$. The increments ω_i satisfy

$$\omega_i - h \sum_{j=1}^q a_{ij} J \omega_j - Z_i = 0 \quad 1 \leq i \leq q \quad (3.3a)$$

where J is the $s \times s$ Jacobian matrix evaluated at an appropriate point and

$$Z_i = -Y_i + y_{n-1} + h \sum_{j=1}^q a_{ij} f(Y_j), \quad 1 \leq i \leq q. \quad (3.3b)$$

Let $M = \bar{I} \otimes I - hA \otimes J$ be the matrix of coefficients in (3.2) where \bar{I} is $q \times q$, I is $s \times s$ and \otimes denotes a tensor product. Relation (3.3a) can then be written as

$$M\omega - Z = 0 \quad (3.4)$$

for the appropriate vectors ω , Z and the matrix M .

We assume that system (3.4) is to be solved for ω in the usual way by first LU decomposing the matrix M . The number of multiplications required to carry out this decomposition is $Cs^3/3 + O(s^2)$ for large s and for the back substitution we require $Ds^2 + O(s)$ multiplications where $C = q^3$ and $D = q^2$. The idea behind Butcher's approach is to solve a suitably transformed system of equations so that the factors C and D are considerably lowered.

Let P and Q be non-singular $q \times q$ matrices so that

$$\bar{\omega} = (Q^{-1} \otimes I)\omega, \quad \bar{z} = (P \otimes I)Z.$$

Then $\bar{M} = (P \otimes I)M(Q \otimes I) = PQ \otimes I - h\bar{A} \otimes J$, where $\bar{A} = PAQ$. It

$$\begin{array}{c|c}
 c_1 & \\
 c_2 & \\
 \vdots & \\
 c_q & \\
 \hline
 b_1 & b_2 \dots b_q
 \end{array}$$

where $(b_1, b_2, \dots, b_q) = (1, \frac{1}{2}, \dots, \frac{1}{q})V_q^{-1}$, $V_q = \begin{bmatrix} 1 & \dots & c_1^{q-1} \\ \vdots & & \vdots \\ 1 & \dots & c_q^{q-1} \end{bmatrix}$,

$$A_q = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & \alpha_{1q} \\ 1 & 0 & 0 & \dots & 0 & \alpha_{2q} \\ 0 & \frac{1}{2} & 0 & \dots & 0 & \alpha_{3q} \\ \vdots & & & & & \\ 0 & 0 & \dots & \frac{1}{q-1} & 0 & \alpha_{qq} \end{bmatrix}$$

$\alpha_{kq} = (-1)^{q-k} (k^q - 1)(q-1)! \lambda^{q-k+1} / (k-1)!$, $1 \leq k \leq q$ and where λ is real and non-zero. Burrage (1978a) defines the concept of a transformed Runge-Kutta method and then gives the following definition.

Definition 7

A transformed method of order q or greater whose Runge-Kutta matrix has just one real q -fold eigenvalue is called a singly implicit method.

Burrage also proved that the maximum attainable order of a singly implicit Runge-Kutta method with q stages is $q+1$. The connection between these formulae and the formulae of Nørsett (1976) comes from the observation that if z_1, z_2, \dots, z_q are the zeros of the q^{th} degree Laguerre polynomial

$$L_q(z) = \sum_{j=0}^q (-1)^j \binom{q}{j} z^j / j!$$

then the singly implicit Runge-Kutta method of Burrage is such that

$$c_i = \lambda z_i, \quad 1 \leq i \leq q, \text{ where } \lambda \text{ is the single } q\text{-fold eigenvalue of } A_q,$$

$$\sum_{j=1}^q a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad i, k = 1, 2, \dots, q$$

$$\sum_{j=1}^q b_j c_j^{k-1} = \frac{1}{k}, \quad k = 1, 2, \dots, q.$$

The final remaining step was to derive explicitly the transformation matrices required to implement these singly implicit formulae and this was done by Butcher (1979).

The importance of singly implicit Runge-Kutta formulae from an algorithmic point of view is that it is possible to derive a whole class of formulae of increasing order and that for each order q it is possible to derive an embedded formula of order $q+1$ which is used for the purpose of local error estimation. This has led to the development of the program STRIDE due to Butcher, Burrage and Chipman (1979a,b) which is based on the singly implicit formulae of Burrage. STRIDE is a variable step/variable order program which allows the use of formulae of orders 1(1)15 all of which are $A(\alpha)$ -stable with $\alpha > 83^\circ$. The implementation details of these formulae are too numerous to describe here but are fully described in Butcher, Burrage and Chipman (1979a). Although STRIDE still does not seem to be generally competitive with codes based on backward differentiation formulae (see also Gaffney (1982) for further comments) it should be remembered that linear multistep methods have benefited from a vast amount of testing and re-programming. It is therefore quite possible that the gap between the performance of BDF and R-K methods will continue to decrease as more experience is gained with STRIDE and its strengths and weaknesses become better understood.

4. Diagonally Implicit and Mono-Implicit Runge-Kutta Methods

Closely related to the singly implicit Runge-Kutta formulae implemented in STRIDE is the class of diagonally implicit Runge-Kutta formulae. This class of formulae was originally proposed by Nørsett (1974) and was further developed by Alexander (1977) and Crouzeix (Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge-Kutta, Thesis, University of Paris VI, Paris, 1975). A diagonally implicit Runge-Kutta formula is one whose defining matrix A is lower triangular. As a result, the use of a Newton iteration scheme to solve for the k_i associated with these methods calls for the solution of q sets of s algebraic equations. Normally these methods are constructed so that the diagonal elements are all equal, called DIRK methods by Alexander and SDIRK by others, but formulae where the diagonal elements are "almost" equal also seem worthy of study.

Alexander proved that a q -stage, A -stable DIRK formula is strongly S -stable if $a_{qi} = b_i$, $1 \leq i \leq q$, and $c_q = 1$. In Alexander (1977) he derived strongly S -stable DIRK formulae of order 2 in 2 stages, order 3 in 3 stages and showed that it is impossible to derive a strongly S -stable DIRK formula of order 4 in 4 stages. Cash (1979) gave a strongly S -stable DIRK formula of order 4 in 5 stages and Cooper and Sayfy (1979) gave one of order 5 in 6 stages. In Alexander's paper some numerical results are given which indicate that DIRK formulae can in some cases be competitive with linear multi-step methods especially when low precision is requested. However, Alexander, and others, have pointed out that efficient error estimation and, more particularly, change of order, is a problem with DIRK formulae since it seems difficult to derive families of DIRK formulae suitable for inclusion in a VSVO algorithm.

There are several important theoretical results known concerning the maximum attainable order of q-stage DIRK formulae. Nørsett (1974) has shown that the maximum order attainable by a DIRK formula with q stages is generally $q + 1$ but if $q = 2\mu$, $2 \leq \mu \leq 10$, this maximum order is q. Nørsett and Wolfbrandt (1977) have given an important, and somewhat surprising, result concerning singly implicit Runge-Kutta formulae and DIRK formulae in cases where all the diagonal elements of the defining matrix A are not necessarily equal. They examine rational approximations of the form

$$\sum_{i=0}^m a_i z^i / \prod_{i=1}^n (1 + \gamma_i z) \text{ to } e^{-z} \text{ where } z \in \mathbb{C}, \gamma_i \in \mathbb{R}, \quad (4.1)$$

and show that the maximum attainable order is $m+1$ with the least absolute value of the error constant being when $\gamma_1 = \gamma_2 = \dots = \gamma_n$. This result immediately implies the result of Nørsett that the order of a q-stage semi-implicit Runge-Kutta formula cannot exceed $q+1$. An additional, rather negative result from the point of view of DIRK formulae, has been given by Hairer (1980a) who has shown that the order of an algebraically stable DIRK formula cannot exceed four.

The computational aspects of DIRK formulae have not received much attention so far. The step control procedure used by Alexander (1977) was based on Richardson extrapolation. However, in view of the widely used technique of embedding for explicit Runge-Kutta formulae (Fehlberg (1964)) it was a natural progression for the error estimation in DIRK formulae to be carried out using pairs of embedded formulae with orders differing by one. Nørsett (1974) derived some low order embedded DIRK formulae and Cash (1979) extended this to derive a third order DIRK formula in three stages and a fourth order one in

five stages, both of which are strongly S-stable and contain an embedded formula of one order lower. Present implementations of DIRK formulae often use fixed order methods. There has not been any extensive testing of DIRK formulae to date although a research level code due to Alexander has performed well in the integration of a certain class of oscillatory problems (Gaffney (1982)). Also a DIRK code SIRKUS has been given by Nørsett (1974) and a semi-implicit Runge-Kutta code for large sparse systems has been developed by Houbak and Thomsen (1979).

The possibility of changing both order and stepsize efficiently with DIRK formulae came with the investigation of block DIRK formulae. Block implicit Runge-Kutta formulae have been around for some time (Shampine and Watts (1972), Watanabe (1978), Williams and de Hoog (1974)) but generally these are based on fully implicit Runge-Kutta formulae and often they are only proposed as a starting procedure (Gear (1980)). Block DIRK formulae have been considered in Cash (1982) and such formulae offer important computational advantages over conventional DIRK formulae. For example, the second order block formula given by Cash is

$$\begin{array}{l|cccc}
 1 & 1 & & & \\
 2 & 1 & 1 & & \\
 1 & \frac{1}{2} & -\frac{1}{2} & 1 & \\
 2 & 3 & -1 & -1 & 1 \\
 3 & -\frac{7}{6}-6b & 3b & \frac{11}{3}+6b & -\frac{1}{2}-3b & 1 \\
 \hline
 & -\frac{7}{6}-6b & 3b & \frac{11}{3}+6b & -\frac{1}{2}-3b & 1 & \text{at } n+3 \\
 & 3 & -1 & -1 & 1 & & \text{at } n+2 \\
 & \frac{1}{2} & -\frac{1}{2} & 1 & & & \text{at } n+1
 \end{array} \quad b = 0.256 \quad (4.2)$$

This formula is strongly S-stable and gives second order solutions at $n+1$, $n+2$, $n+3$ as well as first order solutions at $n+1$ and $n+2$. It can be seen that (4.2) gives second order solutions at three step points using a total of five stages so we describe this formula as being second order and requiring " $1\frac{2}{3}$ stages per step". Also given by Cash is a third order formula requiring $2\frac{1}{4}$ stages per step, a fourth order one requiring $2\frac{4}{5}$ stages per step and a fifth order one requiring $3\frac{1}{6}$ stages per step. All these formulae are strongly S-stable, or very nearly so, and the stages required per step are considerably less than for conventional strongly S-stable DIRK formulae. Cash also describes procedures for varying both the order and stepsize of these formulae and some numerical results are given.

In Cash (1982) it is argued strongly that in the case of implicit Runge-Kutta formulae it may not be valid to compare the efficiency of two formulae by counting the number of stages. It is clear that we must also take account of the computational effort required to evaluate the k_i . This point is highlighted if we consider an extrapolation method. The trapezoidal rule using $h - \frac{1}{2}h$ Richardson extrapolation without smoothing can be written as

$$y_{n+1, h} = y_{n, h} + \frac{h}{2} [f(x_{n+1}, y_{n+1, h}) + f(x_n, y_{n, h})],$$

$$y_{n, h} = y_{n, h/2} = y_n. \quad (4.3a)$$

$$y_{n+i/2, h/2} = y_{n+(i-1)/2, h/2} + \frac{h}{4} [f(x_{n+i/2}, y_{n+i/2, h/2}) + f(x_{n+(i-1)/2}, y_{n+(i-1)/2, h/2})], \quad i = 1, 2. \quad (4.3b)$$

The extrapolated solution y_{n+1} is given by

$$y_{n+1} = \frac{1}{3} (4y_{n+1,h/2} - y_{n+1,h}). \quad (4.3c)$$

Formally this process for computing y_{n+1} can be written as the diagonally implicit Runge-Kutta formula

| | | | | | |
|---------------|---------------|----------------|---------------|---------------|--|
| 0 | 0 | | | | |
| 1 | $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| $\frac{1}{2}$ | $\frac{1}{4}$ | 0 | $\frac{1}{4}$ | | |
| 1 | $\frac{1}{4}$ | 0 | $\frac{1}{2}$ | $\frac{1}{4}$ | |
| | $\frac{1}{6}$ | $-\frac{1}{6}$ | $\frac{2}{3}$ | $\frac{1}{3}$ | |

(4.4)

We see that this formula has four stages but this does not tell the whole story. This is because the stage k_4 is likely to be very cheap to evaluate since, when computing the approximation $y_{n+1,h/2}$, we already have the approximations $y_{n+1,h}$ and $y'_{n+1,h}$ available. Many extrapolation methods can be regarded as high order diagonally implicit Runge-Kutta methods which require more than the minimum possible number of stages to achieve a given order but which are such that some of the stages are very cheap to evaluate. Note, however, that extrapolation methods are such that the diagonal elements for the corresponding DIRK method are not all equal.

We now consider the class of Mono-Implicit Runge-Kutta formulae introduced by Cash (1975). The idea of introducing this class of formulae came from the observation that if we take the forward Euler rule

$$y_{n+1} - y_n = hf_n,$$

which has only a small region of absolute stability, and replace h by $-h$ we obtain the backward Euler rule

$$y_n - y_{n-1} = hf_n$$

which is known to be L-stable. This idea can be extended by taking any explicit Runge-Kutta formula and replacing h by $-h$ to obtain what is called the "backward version" of the Runge-Kutta formula. If the explicit Runge-Kutta formula when applied with fixed h to the equation $y' = \lambda y$ gives

$$y_{n+1}/y_n = R(z), \quad z = h\lambda,$$

and the backward version gives

$$y_{n+1}/y_n = \bar{R}(z)$$

it can easily be shown that

$$\bar{R}(z) = \frac{1}{R(-z)}. \quad (4.5)$$

This leads us to the "reflection principle" that the stability region of the backward formula is the complementary set of the image of that of the classical explicit formula reflected in the imaginary axis. For further extensions of this idea see Stetter (1973). This class of backward formulae generally has very good stability properties and it can be shown to be a subset of the more general class of formulae

$$y_{n+1} - y_n = h \sum_{j=0}^k b_j f(x_{n+\alpha_j}, \bar{y}_{n+\alpha_j}) \quad (4.6a)$$

$$\bar{y}_{n+\alpha_j} = \delta_j y_{n+1} + (1-\delta_j) y_n + h \sum_{i=0}^m a_{ji} f(x_{n+\alpha_i}, \bar{y}_{n+\alpha_i}). \quad (4.6b)$$

If in (4.6b) we take $m = j - 1$, $\delta_j \equiv 0$ we obtain a classical explicit Runge-Kutta formula. If $m = j$, $\delta_j = 0$, $a_{jj} = \beta$ for all j we obtain a diagonally implicit Runge-Kutta formula and if $m = k$, $\delta_j = 0$ we obtain a fully implicit Runge-Kutta formula. If, however, we take $m = j - 1$ and $\delta_j \neq 0$ for at least one j we obtain a formula which is implicit in the single unknown y_{n+1} . In Cash (1975) the general class of formulae proposed was of the form (4.6a,b) with

$$\delta_j = 0, \quad m = j - 1, \quad 0 \leq j \leq r,$$

$$\delta_j = 1, \quad m = j - 1, \quad a_{ji} = 0, \quad r + 1 \leq j \leq k, \quad 0 \leq i \leq r,$$

where $r \leq \lfloor \frac{1}{2}k \rfloor$. Some numerical testing of these formulae was done by Birnbaum and Lapidus (1978) and they were found to perform well on a set of test problems from a chemical engineering environment. These formulae were further investigated by van Bokhoven (1980) who derived the general order relations to be satisfied by formulae of order ≤ 6 . As with fully implicit Runge-Kutta formulae, the way in which MIRK formulae are implemented is of great importance. The system of algebraic equations to be solved is of dimension s but the coefficient matrix of the modified Newton iteration scheme is of the form

$$\sum_{i=0}^t \alpha_i h^{i,j} J^i \quad (4.7)$$

where J is an approximation to the Jacobian matrix. In Cash and Singhal (1982) a special class of MIRK formulae were investigated which is such that (4.7) can be approximated by

$$(I - \beta h J)^t$$

since in this case no matrix-matrix products are called for. Some L-stable formulae of this class with order ≤ 4 have been given in

Cash and Singhal (1982) and some numerical results are presented.

5. Defect Correction

The technique of defect correction is a further attempt to overcome the difficulty of solving large systems of nonlinear algebraic equations when applying high-order implicit Runge-Kutta methods. The basic idea behind the Iterated Defect Correction (IDeC) technique is to obtain an approximate solution of the differential equation using a "cheap" low-order implicit R-K method and then to improve this approximate solution iteratively by means of a correction procedure. The technique of IDeC has been given by Frank and Ueberhuber (1977, 1978). They show that every IDeC method based on the backward Euler rule, using piecewise polynomial interpolation, has a fixed point which coincides with the solution obtained using an appropriate polynomial collocation scheme. However, in general, the IDeC technique calls for not much more computational effort than is required by a linear multistep method to solve the algebraic equations arising from a modified Newton iteration. The technique of IDeC as described by Frank and Ueberhuber (1977) is as follows.

Consider the solution of the stiff differential system

$$y' = f(x,y), \quad y(0) = y_0, \quad 0 \leq x \leq H.$$

Define a grid

$$\pi \equiv \{x_\nu = H\zeta_\nu, \nu = 1(1)m \mid 0 < \zeta_1 < \dots < \zeta_m = 1\}.$$

On this grid apply the backward Euler rule

$$\eta_{\nu+1}^{(0)} = \eta_\nu^{(0)} + h_\nu f(x_{\nu+1}, \eta_{\nu+1}^{(0)}), \quad \nu = 0(1)m-1, \quad h_\nu = x_{\nu+1} - x_\nu. \quad (5.1)$$

This gives a solution $\eta^{(0)} = (\eta_0^{(0)}, \eta_1^{(0)}, \dots, \eta_m^{(0)})^T$. Now construct an interpolating polynomial $P^0(x)$ of degree m which interpolates the points $\{x_\nu, \eta_\nu^{(0)}\}_{\nu=0}^m$, i.e.

$$P^0(x_\nu) = \eta_\nu^{(0)}, \nu = 0(1)m. \quad (5.2)$$

Having constructed this polynomial we can define the defect, which is a continuous function of x , as

$$d^0(x) = \frac{d}{dx} [P^0(x)] - f(x, P^0(x)). \quad (5.3)$$

This allows the following initial value problem to be constructed

$$y'(x) = f(x, y) + d^0(x), \quad y(0) = y_0$$

$$\equiv f(x, y) + \frac{d}{dx} [P^0(x)] - f(x, P^0(x)) \quad (5.4)$$

which has the known solution $y(x) = P^0(x)$. We now solve the initial value problem (5.4) in exactly the same way, i.e. using the backward Euler rule on the same grid π , to obtain a numerical solution

$$\tau^{(0)} \equiv (\tau_0^{(0)}, \tau_1^{(0)}, \dots, \tau_m^{(0)}).$$

We now use the known discretization errors $\tau_\nu^{(0)} - P^0(x_\nu)$ as estimates of the unknown errors $\eta_\nu^{(0)} - y(x_\nu)$. This approach follows an original idea due to Zadunaisky (1976). We now replace the unknown errors in the identity

$$y(x_\nu) = \eta_\nu^{(0)} - (\eta_\nu^{(0)} - y(x_\nu)), \nu = 0(1)m$$

by the known error estimates $\tau_\nu^{(0)} - P^0(x_\nu)$. This leads to the following formula to improve the first approximate solution $\eta^{(0)}$:

$$\eta_v^{(1)} = \eta_v^{(0)} - (\tau_v^{(0)} - P^0(x_v)), \quad v = 0(1)m. \quad (5.5)$$

Following an idea due to Stetter (1974) this whole procedure can be used iteratively as

$$\eta_v^{(j+1)} = \eta_v^{(0)} - (\tau_v^{(j)} - P^j(x_v)) \quad (5.6)$$

where $P^j(x)$ is the polynomial of degree m interpolating the points $\{x_v, \eta_v^{(j)}\}_{v=0}^m$. Subject to some general differentiability conditions on the function f , Fränk and Ueberhuber (1977) have proved the following result for the IDeC method applied on an equi-distant grid for a fixed degree of polynomial m and for $h = H/m$:

If an arbitrary Runge-Kutta scheme of order $p(\leq m)$ is used and if f satisfies suitable differentiability conditions then

$$\eta_v^{(j)} - y(x_v) = O(h^{\min(p(j+1), m)}).$$

Thus the technique of IDeC achieves the high order attainable by implicit Runge-Kutta formulae but at a significantly less computation cost. Much of the theoretical analysis concerning IDeC has now been completed (Stetter (1974), (1978), Hairer (1978), Frank and Ueberhuber (1978)) and this approach does seem to be a very promising one.

Actual implementations of defect correction have been largely based on the backward Euler rule although Butcher (1979b) has considered the application of defect correction to a more general class of Runge-Kutta methods. Ueberhuber (1979) has considered many of the computational details associated with a particular implementation of IDeC but it is clear that much more experience is needed in this area before a really successful implementation can be contemplated.

6. Runge-Kutta Methods Using an Approximation to the Jacobian

6.1 Rosenbrock Methods

For the numerical integration of the autonomous differential system

$$\frac{dy}{dx} = f(y), \quad y(x_0) = y_0, \quad (6.1)$$

Rosenbrock (1963) proposed the class of q-stage integration methods defined by

$$y_{n+1} = y_n + h \sum_{r=1}^q b_r k_r \quad (6.2a)$$

$$k_1 = f(y_n) + \alpha_1 h J(y_n) k_1$$

$$k_r = f\left(y_n + h \sum_{s=1}^{r-1} a_{rs} k_s\right) + \alpha_r h J\left(y_n + h \sum_{s=1}^{r-1} \beta_{rs} k_s\right) k_r,$$

$$2 \leq r \leq q, \quad (6.2b)$$

where $J(y_n) \equiv \left. \frac{\partial f}{\partial y} \right|_{y=y_n}$. Such formulae have the major computational

advantage that it is only necessary to solve linear systems of algebraic equations to find the k_r but have the disadvantage that the exact Jacobian is required at each step. In view of this, Rosenbrock methods are only likely to be competitive with other classes of methods on problems for which the Jacobian matrix is not too expensive to evaluate. An important class of problems of this type has been given by Edsberg (1974). Edsberg considered problems where the elementary steps of some composite reaction taking place in a homogeneous solution according to the mass action law are known. In such cases the system of O.D.E.s describing the kinetic behaviour of the system

can be set up from the coefficients and the structure of the reactions. Edsberg showed that such problems can often be written in the form

$$y' = Ap, \quad y(0) \text{ given}, \quad (6.3)$$

where A is an M x N matrix with integer entries and p is an N-vector with

$$p_j = k_j \prod_{i=1}^M y_i^{r_{ji}}. \quad (6.4)$$

Here the $r_{ji} \geq 0$ are integers describing the reactions and the $k_j > 0$ are the rate constants. For such problems we can obtain the Jacobian matrix f_y directly from the relation

$$\frac{\partial p_j}{\partial y_i} = r_{ji} p_j / y_i. \quad (6.5)$$

Most early studies of Rosenbrock methods appeared in the engineering literature. In particular we mention the work of Calahan (1968), Allen and Pottle (1966), Caillaud and Padmanabhan (1971), Lapidus and Seinfeld (1971) and Michelson (1976). In Cash (1976) the suggestion was made to look for Rosenbrock formulae with $\beta_{rs} = 0$ since these require only one Jacobian evaluation per step. Also given by Cash was a novel form of error estimation which is applicable to a quite wide class of Runge-Kutta methods. To describe it we consider the second order Rosenbrock method

$$y_{n+1} - y_n = h(b_1 k_1 + b_2 k_2) \quad (6.6)$$

$$k_1 = [I - ahJ(y_n)]^{-1} f(y_n)$$

$$k_2 = [I - ahJ(y_n)]^{-1} f(y_n + hc_1 k_1).$$

The coefficients of this formula can be chosen so that it is L-stable (Cash (1976)) and we call the resulting formula $R_2(x_n, \omega_1, \omega_2, a, c_1, h)$. The procedure used to obtain an estimate of the local truncation error is a modified form of Richardson extrapolation whereby two approximate solutions are found at each step point, one using two steps of size $h/2$ and the other using one step of length h . Thus, starting from the point x_{n-1} we use the integration formulae $R_2(x_{n-1}, \omega_1, \omega_2, a, c_1, h/2)$, $R_2(x_{n-2}, \omega_1, \omega_2, a, c_1, h/2)$ to compute an approximate solution y_n at x_n . A second approximate solution \bar{y}_n is computed using the formula $R_2(x_{n-1}, \bar{\omega}_1, \bar{\omega}_2, a/2, c_1/2, h)$. The crucial point to note about using the formula R_2 with a steplength h is that it uses exactly the same k_1 and k_2 as were used by $R_2(x_{n-1}, \omega_1, \omega_2, a, c_1, h/2)$. This means that, in general, a negligible extra computational effort is required to compute the error estimate. In Cash (1976) an L-stable formula of order two in two stages and an L-stable formula of order three in three stages using a Merson-type error estimate were derived. This technique has recently been used by Bui (1981) to obtain a fully embedded formula of order 3.

A modification of Rosenbrock formulae was given by Wolfbrandt (1977). He introduced the class of ROW methods given by

$$[I - \gamma h J(y_n)] k_i = h f(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j) + h J(y_n) \sum_{j=1}^{i-1} \gamma_{ij} k_j,$$

$$1 \leq i \leq q, \quad (6.7a)$$

$$y_{n+1} = y_n + \sum_{i=1}^q b_i k_i.$$

Kaps and Warner (1981) have shown that mathematically this formulation is equivalent to the computationally more efficient class of formulae

where

$$[I - \bar{\gamma}hJ(y_n)]\bar{k}_i = hf(y_n + \sum_{j=1}^{i-1} \bar{\alpha}_{ij}\bar{k}_j) + h \sum_{j=1}^{i-1} \bar{\gamma}_{ij}\bar{k}_j. \quad (6.7b)$$

We shall refer to both these classes as ROW methods and we note that if $\gamma_{ij} = \bar{\gamma}_{ij} \equiv 0$ they reduce to Rosenbrock methods. If (6.7a) is applied with a fixed stepsize h to the scalar equation $y' = \lambda y$ we obtain

$$y_{n+1} = R(z)y_n, \quad z = h\lambda,$$

where, if the method is of order q at least,

$$R(z) = \left[\sum_{j=0}^q z^j \sum_{i=0}^j \binom{q}{i} \frac{(-\gamma)^i}{(j-i)!} \right] / (1-\gamma z)^q.$$

An investigation of how to choose γ to give optimal stability and accuracy has been carried out by Warner (1980). Higher order formulae of the Rosenbrock, or ROW, class have been derived by several people and in particular we mention the work of Bui (1979) who derives an L-stable formula of order 4 in 4 stages, Kaps and Warner (1981) who derive methods of order five and six requiring just one Jacobian evaluation per step (but the step control is by Richardson extrapolation), Kaps and Rentrop (1979) who derive a fourth order method with an embedded third order method for error estimation, Nørsett and Wolfbrandt (1979) who derive the order relations for Rosenbrock methods by extending the Butcher series approach, and the thesis of Wolfbrandt (1977) which investigates ROW methods. Finally, we mention a report by Shampine (1980) who raises several interesting points concerning

the implementation of Rosenbrock methods. Generally speaking, Runge-Kutta formulae evaluate the function $f(x,y)$ several times in the interval $[x_n, x_{n+1}]$. Shampine argues that it is desirable that these function evaluations should span $[x_n, x_{n+1}]$ so that the formula is able to spot any "quasi-discontinuities" in the solution. An example of a formula which does not satisfy this condition, the implicit mid-point rule, is discussed and by considering a differential equation of the form

$$y'(t) = d - by + aE(t), \quad (6.8)$$

where the forcing function $E(t)$ is a square wave, Shampine shows that the implicit mid-point rule can sometimes give a very poor solution to (6.8) due to its inability to spot a quasi-discontinuity.

Another important point raised by Shampine is that the linear system of algebraic equations defining the k_i is almost bound to be ill-conditioned. This is also the case for BDF methods but for these formulae this is not serious since they solve for the change in the solution and normally it is only necessary to get the first few digits correct. However, for Rosenbrock methods this ill-conditioning can be serious and special care must be taken to monitor it. Shampine derives a fourth order formula with an embedded formula of order three, both of which have rational coefficients and satisfy the spanning conditions, and discusses their implementation and practical performance in some detail.

A class of formulae similar to Wolfbrandt's methods was proposed by Cash (1980). These take the form

$$y_{n+1} - y_n = \sum_{i=1}^q \omega_i k_i \quad (6.9a)$$

$$[I - h \gamma \frac{\partial f}{\partial y}(y_n)]^2 k_i = h \{I + ch \frac{\partial f}{\partial y}(y_n)\} f(y_n + \sum_{j=1}^{i-1} b_{ij} k_j). \quad (6.9b)$$

Each k_i calls for the solution of a linear system of equations of the form

$$A^2 k_i = b.$$

This can be done efficiently, i.e. without the need for matrix products, by LU decomposing A and solving the systems

$$LUz = b$$

$$LUk_i = z.$$

Cash gives an L-stable formula of order 3, with an embedded L-stable formula of order 2, of the form (6.9) and the results of some numerical computations are reported.

An obvious extension of Rosenbrock methods is to see what can be done with formulae of the general form (6.2) where J is no longer the exact Jacobian. An interesting contribution in this area is due to Steihaug and Wolfbrandt (1979) who consider methods of the form

$$(I - h d_{ii} A) k_i = f(y_n + h \sum_{j=1}^{i-1} a_{ij} k_j) + h A \sum_{j=1}^{i-1} d_{ij} k_j \quad 1 \leq i \leq q. \quad (6.10)$$

Here A is a real square matrix and h is chosen so that $I - h d_{ii} A$ is non-singular. Since A is no longer the exact Jacobian, there are considerably more order relations to be satisfied than for ordinary ROW methods. In particular Steihaug and Wolfbrandt show that to get order 3 it is necessary to satisfy eight order relations. They give a

second order formula with an embedded error estimate and present the results of some numerical computations.

An alternative approach to the problem of deriving Runge-Kutta formulae with coefficients depending on an inexact Jacobian is the one due to Verwer (1977). Verwer has investigated a class of formulae considered by van der Houwen (1972a,b) and defines a generalised m-point Runge-Kutta method to be

$$y_{n+1} = y_n + \sum_{j=0}^{m-1} \Lambda_{m,j} (h_n J_n) k_n^{(j)}; \quad (6.11)$$

$$k_n^{(j)} = h_n f(x_n + u_j h_n, y_n + \sum_{\ell=0}^{j-1} \Lambda_{j,\ell} (h_n J_n) k_n^{(\ell)}), \quad h_n = x_{n+1} - x_n,$$

where $u_j = \sum_{\ell=0}^{j-1} \Lambda_{j,\ell}^{(0)}$, $j = 0, 1, \dots, m-1$ and where $\Lambda_{j,\ell}$, $j = 1, 2, \dots, m$, $\ell = 0, 1, \dots, j-1$ are rational functions with real coefficients. Verwer derives a class of generalised Runge-Kutta formulae which have order independent of the choice of J_n . Associated with these formulae, Verwer defines the concept of internal S-stability as follows.

Define $y_{n+1}^{(j)}$ by the relation

$$k_n^{(j)} = h_n f(x_n + u_j h_n, y_{n+1}^{(j)}), \quad j = 1, 2, \dots, m.$$

The Runge-Kutta formula (6.11) can then be written as

$$y_{n+1}^{(0)} = y_n$$

$$y_{n+1}^{(j)} = y_n + h_n \sum_{\ell=0}^{j-1} \Lambda_{j,\ell} (h_n J_n) f(x_n + u_\ell h_n, y_{n+1}^{(\ell)}), \quad j=1, 2, \dots, m \quad (6.12)$$

$$y_{n+1} = y_{n+1}^{(m)}.$$

The usual approach in the stability analysis of methods for stiff problems is to examine the stability of the solution only at the end point of the current step. However, for those Runge-Kutta schemes which are such that the solution is built up successively in several intermediate stages it is also of interest to examine the stability properties of the intermediate solutions $y_{n+1}^{(j)}$. To allow for this, Verwer gives the following definition of internal S-stability.

Definition 8

The integration formula (6.12) is said to be internally S-stable if at each j^{th} stage, $j = 1, 2, \dots, m$, the corresponding scheme at stage j is S-stable.

Verwer considers the performance of three second order formulae of the form (6.11) on a set of three test problems. The first formula is L-stable but not S-stable, the second is S-stable but not internally S-stable and the third is internally S-stable. The numerical results presented suggest that the third formula is superior to the other two on the problems considered.

7. Nonlinear Methods

Finally, we mention briefly an interesting class of nonlinear Runge-Kutta methods. It seems likely that such methods will experience difficulties in solving some classes of stiff systems but numerical experience obtained so far indicates that they may have real potential in solving the stiff equations arising from the methods of lines solution of parabolic partial differential equations. Their main feature is that, being nonlinear, they are able to achieve A-stability while still being explicit. Lambert (1974) seems to have been the

first to consider nonlinear Runge-Kutta methods. One particular example he gives is

$$y_{n+1} - y_n = hf(x_{n+\frac{1}{2}}, y_n + \frac{1}{2} hf_n / (y_n - \frac{1}{2} hf_n)). \quad (7.1)$$

This method has the advantage that it is "component applicable" (see Lambert (1974), p. 176) but has the disadvantage that, although having order 2 in general it only has order 1 if $y = 0$. Also, of course, care must be taken to ensure that $y_n - \frac{1}{2} hf_n$ does not vanish. Nonlinear Runge-Kutta methods were also examined by Wambecq (1978) and Hairer (1980b). The general class of formulae they consider is given by

$$y_1 = y_0 + \sum_{i=1}^s \sum_{j=1}^i \omega_{ij} \frac{g_i g_j}{\sum_{k=1}^s b_k g_k} \quad (7.2)$$

$$g_i = hf(y_0 + \sum_{j=1}^{i-1} a_{ij} g_j).$$

These formulae no longer have a simple interpretation in component form. Instead, the expression $\frac{ab}{d}$ appearing in (7.2) must be interpreted for real or complex vectors as

$$\frac{ab}{d} = \frac{a \operatorname{Re}(b,d) + b \operatorname{Re}(d,a) - d \operatorname{Re}(a,b)}{(d,d)}$$

where (a,b) denotes the scalar product of a and b . These methods are more expensive to implement for systems than those due to Lambert but do not suffer from the problem that the order changes if y passes through zero. However, there is still the problem that the denominator may become zero as is emphasised by the second order equation

$$g_1 = hf(y_0)$$

$$g_2 = hf(y_0 + \frac{1}{2} g_1)$$

$$y_1 = y_0 + \frac{g_1 g_1}{2g_1 - g_2} ,$$

so they need to be implemented with particular care.

Hairer (1980b) gives a second order A_0 -stable method with an embedded error control. In numerical tests this method was found to perform poorly on some stiff problems arising in chemical kinetics and so they are probably ruled out for the integration of general stiff systems. However, good results were obtained for the integration of some parabolic P.D.E.s using the method of lines. Because of these results and the very low storage requirement of the formulae (7.2) (they do not require the evaluation or storage of a Jacobian), they seem worthy of further attention as candidates for use with the method of lines solution of parabolic partial differential equations.

- Alexander, R. (1977). Diagonally implicit Runge-Kutta methods for stiff O.D.E.s. SIAM J. Numer. Anal., vol. 14, 1006-1021.
- Allen, R.H. and Pottle, C. (1966). Stable integration methods for electronic circuit analysis with widely separated time constants. In Sixth Annual Allerton Conference on Circuit and System Theory (ed. T. Trick and R.T. Chien), pp. 311 - 320.
- Axelsson, O. (1969). A class of A-stable methods. BIT, vol. 9, 185-199.
- Bickart, T.A. (1977). An efficient solution process for implicit Runge-Kutta methods. SIAM J. Numer. Anal., vol. 14, 1022-1027.
- Birnbaum, I. and Lapidus, L. (1978). Studies in approximation methods II. Chemical Engineering Science, vol. 33, 427-441.
- Birkhoff, G. and Varga, R.S. (1965). Discretization errors for well-set Cauchy problems: 1. J. Math. and Phys., vol. 44, 1-25.
- Bond J.E. and Cash, J.R. (1979). A block method for the numerical integration of stiff systems of ordinary differential equations. BIT, vol. 19, 429-447.
- Bui, T.D. (1979). Some A-stable and L-stable methods for the numerical integration of stiff ordinary differential equations. J.ACM, vol. 26, 483-493.
- Bui, T.D. and Poon, S.W.H. (1981). On the computational aspects of Rosenbrock procedures with built in error estimates for stiff systems. BIT, vol. 21, 168-174.
- Burrage, K. (1978a). A special family of Runge-Kutta methods for solving stiff differential equations. BIT, vol. 18, 22-41.
- Burrage, K. (1978b). High order algebraically stable Runge-Kutta methods. BIT, vol. 18, 373-383.

- Burrage, K. and Butcher, J.C. (1979). Stability criteria for implicit Runge-Kutta methods. *SIAM J. Numer. Anal.*, vol. 16, 30-45.
- Burrage, K., Butcher, J.C. and Chipman, F. (1980). An implementation of singly implicit Runge-Kutta methods. *BIT*, vol. 20, 326-340.
- Butcher, J.C. (1963). Coefficients for the study of Runge-Kutta integration processes. *J. Austral. Math. Soc.*, vol. 3, 185-201.
- Butcher, J.C. (1964). Implicit Runge-Kutta processes. *Math. Comp.*, vol. 18, 50-64.
- Butcher, J.C. (1965). On the attainable order of Runge-Kutta methods. *Math. Comp.*, vol. 19, 408-417.
- Butcher, J.C. (1975). A stability property of implicit Runge-Kutta methods. *BIT*, vol. 15, 358-361.
- Butcher, J.C. (1976). On the implementation of implicit Runge-Kutta methods. *BIT*, vol. 16, 237-240.
- Butcher, J.C. (1979a). A transformed implicit Runge-Kutta method. *J.ACM.*, vol. 26, 731-738.
- Butcher, J.C. (1979b). Some implementation schemes for implicit Runge-Kutta methods, In *Proceedings of the 1979 Dundee Biennial Numerical Analysis Conference*, Springer-Verlag, Berlin, pp. 12-24.
- Butcher, J.C., Burrage, K. and Chipman, F. (1979a). An implementation of singly-implicit Runge-Kutta methods. Research Report No. 149, University of Auckland.
- Butcher, J.C., Burrage, K. and Chipman, F. (1979b). STRIDE: Stable Runge-Kutta integrator for differential equations. Research Report No. 150, University of Auckland.
- Caillaud, J.B. and Padmanabhan, L. (1971). An improved semi-implicit Runge-Kutta method for stiff systems. *Chem. Eng. J.*, vol. 2, p. 227.

- Calahan, D.A. (1968). A stable, accurate method of numerical integration for nonlinear circuits, Proc. IEEE, vol. 56, 744.
- Cash, J.R. (1975). A class of implicit Runge-Kutta methods for the numerical integration of stiff ordinary differential equations. J.ACM, vol. 22, 504-511.
- Cash, J.R. (1976). Semi-implicit Runge-Kutta procedures with error estimates for the numerical integration of stiff systems of ordinary differential equations. J.ACM, vol. 23, 455-460.
- Cash, J.R. (1979). Diagonally implicit Runge-Kutta formulae with error estimates. J. IMA, vol. 24, 293-301.
- Cash, J.R. (1980). A semi-explicit Runge-Kutta formula for the numerical integration of stiff systems of O.D.E.s. The Chemical Engineering Journal, vol. 20, 219-224.
- Cash, J.R. (1982). Block Runge-Kutta methods for the numerical integration of initial value problems in O.D.E.s, to appear.
- Cash, J.R. and Singhal, A. (1982). Mono-implicit Runge-Kutta formulae for the numerical integration of stiff differential systems, to appear.
- Chan, Y.N.I., Birnbaum, I. and Lapidus, L. (1978). Solution of stiff differential equations and the use of imbedding techniques. Industr. and Engin. Chemistry Fundamentals, vol. 17, 133-148.
- Chipman, F.H. (1971). A-stable Runge-Kutta methods. BIT, vol. 11, 384-388.
- Cooper, G.J. and Sayfy, A. (1979). Semi-explicit A-stable Runge-Kutta methods. Math. Comp., vol. 33, 541-556.
- Crouzeix, M. (1979). Sur la B-stabilité des méthodes de Runge-Kutta. Numer. Math., vol. 32, 75-82.

- Dahlquist, G. (1963). A special stability problem for linear multi-step methods. BIT, vol. 3, 27-43.
- Dahlquist, G. (1975). Error analysis of a class of methods for stiff nonlinear initial value problems, Proceedings of the Dundee Conference on Numerical Analysis, Springer-Verlag, Berlin 506, 60-74.
- Dahlquist, G. and Jel'tsch, R. (1979). Generalised disks of contractivity for explicit and implicit Runge-Kutta methods. Report TRITA-NA-7906, Dept. of Computer Science, Royal Institute of Technology, Stockholm.
- Edsberg, L. (1974). Integration package for chemical kinetics. In Stiff Differential Systems (ed. R. Willoughby), pp. 81-94.
- Ehle, B.L. (1969). On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems. University of Waterloo, Depart. of Applied Analysis and Computer Science, Research Report No. CSRR 2010.
- Fehlberg, E. (1964). New high order Runge-Kutta formulas with step-size control for systems of first- and second-order differential equations. Z. Angew. Math. Mech., vol. 44, 17-29.
- Frank, R. and Ueberhuber C.W. (1977). Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations. BIT, vol. 17, 146-159.
- Frank, R. and Ueberhuber C.W. (1978). Iterated defect correction for differential equations, Part 1: Theoretical Results. Computing, vol. 20, 207-228.
- Gaffney, P.W. (1982). A survey of FORTRAN subroutines suitable for solving stiff oscillatory ordinary differential equations. Report ORNL/CSD/TM-134, Oak Ridge National Laboratory, Tennessee.

- Gear, C.W. (1980). Runge-Kutta starters for multistep methods. ACM Transactions on Mathematical Software, vol. 6, 263-279.
- Hairer, E. (1978). On the order of iterated defect correction, an algebraic proof. Numer. Math., vol. 29, 409-424.
- Hairer, E. (1980a). Highest possible order of algebraically stable diagonally implicit Runge-Kutta methods. BIT, vol. 20, 254-256.
- Hairer, E. (1980b). Unconditionally stable explicit methods for parabolic equations. Numer. Math., vol. 35, 57-68.
- Hairer, E. and Warner, G. (1981). Algebraically stable and implementable Runge-Kutta methods of high order. SIAM J. Numer. Anal., vol. 18, 1098-1108.
- Houbak, N. and Thomson, P.G. (1979). SPARKS, a FORTRAN subroutine for the solution of large systems of stiff O.D.E.s with sparse Jacobians. NI-79-02. Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark.
- Hundsdorfer, W.H. and Spijker, M.N. (1981). A note on B-stability of Runge-Kutta methods. Numer. Math., vol. 36, 319-331.
- Iserles, A. (1979). On the generalised Padé approximations to the exponential function. SIAM J. Numer. Anal., vol. 16, 631-636.
- Kaps, P. and Rentrop, P. (1979). Generalised Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations. Numer. Math., vol. 33, 55-68.
- Kaps, P. and Wagner, G. (1981). A study of Rosenbrock type methods of high order. Numer. Math., vol. 38, 279-298.
- Lambert, J.D. (1974). Two unconventional classes of methods for stiff systems. In Stiff Differential Systems (ed. R. Willoughby), pp. 171-186.
- Lapidus, L. and Seinfeld, J.H. (1971). Numerical Solution of Ordinary Differential Equations. Academic Press.

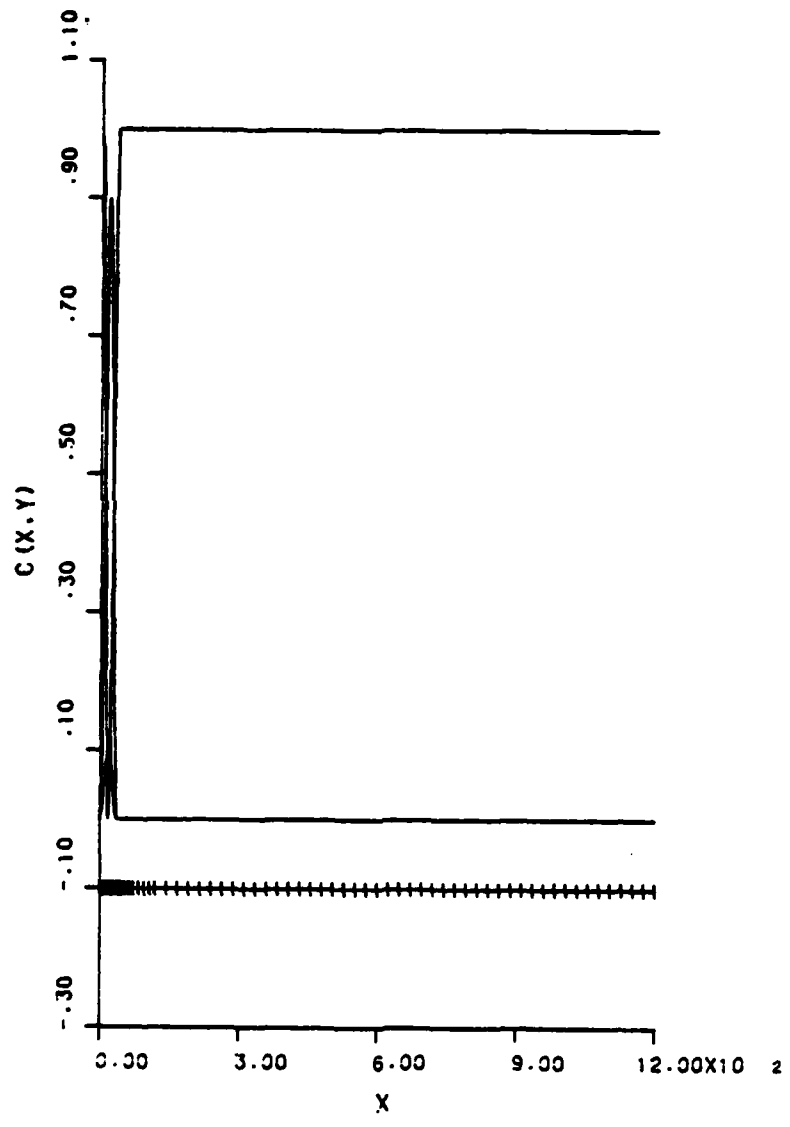
- Michelson, M.L. (1976). An efficient general purpose method for the integration of stiff ordinary differential equations, A.I.Ch.E.J., vol. 22, 594-597.
- Nevalinna, A. and Liniger, W. (1978). Contractive methods for stiff differential equations. BIT, vol. 18, 457-474.
- Nørsett, S.P. (1974). Semi-explicit Runge-Kutta methods. Mathematics and Computation Report No. 6/74, University of Trondheim.
- Nørsett, S.P. (1975). C-polynomials for rational approximation to the exponential function. Numer. Math., vol. 25, 39-56.
- Nørsett, S.P. (1976). Runge-Kutta methods with a multiple real eigenvalue only. BIT, vol. 16, 388-393.
- Nørsett, S.P. and Wolfbrandt, A. (1977). Attainable order of rational approximations to the exponential function with only real poles, BIT, vol. 17, 200-208.
- Nørsett, S.P. and Wolfbrandt, A. (1979). Order conditions for Rosenbrock type methods, Numer. Math., vol. 32, 1-15.
- Nørsett, S.P. and Warner, G. (1981). Perturbed collocation and Runge-Kutta methods. Numer. Math., vol. 38, 193-208.
- Prothero, A. and Robinson, A. (1974). On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. Math. Comp., vol. 28, 145-162.
- Rosenbrock, H.H. (1963). Some general implicit processes for the numerical solution of differential equations. Computer J., vol. 5, 329-330.
- Shampine, L.F. (1980). Implementation of Rosenbrock methods. Report No. SAND80-2367J, Sandia National Laboratories, Albuquerque.
- Shampine, L.F. and Watts, H.A. (1972). A-stable block implicit one-step methods, BIT, vol. 12, 252-266.

- Steihaug, T. and Wolfbrandt, A. (1979). An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Math. Comp.*, vol. 33, 521-534.
- Stetter, H.J. (1973). *Analysis of Discretization Methods for Ordinary Differential Equations*. Springer-Verlag, Berlin-Heidelberg-New York.
- Stetter, H.J. (1974). Economical global error estimation. In *Stiff Differential Systems* (ed. R. Willoughby), 245-258.
- Stetter, H.J. (1975). Towards a theory for discretizations of stiff differential systems. *Lecture Notes in Mathematics*, No. 506, 190-201.
- Stetter, H.J. (1978). The defect correction principle and discretization methods. *Numer. Math.*, vol. 29, 425-443.
- Ueberhuber, C.W. (1979). Implementation of defect correction methods for stiff differential equations. *Computing*, vol. 23, 205-232.
- Van Bokhoven, W.M.G. (1980). Efficient higher order implicit one-step methods for integration of stiff differential equations. *BIT*, vol. 20, 34-43.
- Van der Houwen, P.J. (1972a). Explicit and semi-implicit Runge-Kutta formulas for the integration of stiff equations. Report TW132, Mathematisch Centrum, Amsterdam.
- Van der Houwen, P.J. (1972b). One step methods with adaptive stability functions for the integration of differential equations. In *Lecture Notes in Mathematics*, vol. 333, Springer-Verlag, Berlin-Heidelberg-New York.
- Varah, J.M. (1979). On the efficient implementation of implicit Runge-Kutta methods. *M. Comp.*, vol. 33, 557-561.
- Verwer, J.G. (1977). S-stability properties for generalised Runge-Kutta methods. *Numer. Math.*, vol. 27, 359-370.

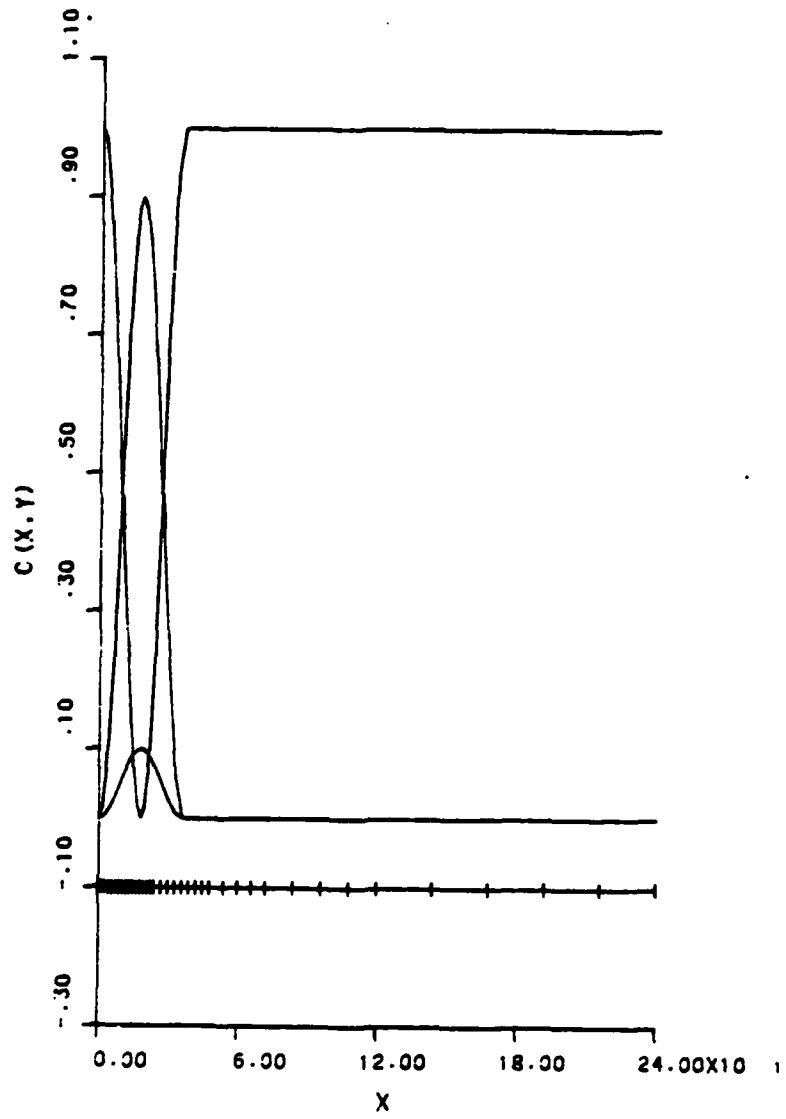
- Wambecq, A. (1978). Rational Runge-Kutta method for solving systems of ordinary differential equations. *Computing*, vol. 20, 333-342.
- Wazner, G. (1976). A short proof of non-linear A-stability. *BIT*, vol. 16, 226-227.
- Wazner, G. (1980). On the choice of γ for singly implicit R-K or Rosenbrock methods. *BIT*, vol. 20, 102-106.
- Wazner, G., Hairer, E. and Nørsett, S.P. (1978). Order stars and stability theorems. *BIT*, vol. 18, 475-489.
- Watanabe, D.S. (1978). Block implicit one step methods. *Math. Comp.*, vol. 32, 405-414.
- Williams, J. and de Hoog, F. (1974). A class of A-stable advanced multistep methods. *Math. Comp.*, vol. 28, 163-177.
- Wolffbrandt, A. (1977). A study of Rosenbrock processes with respect to order conditions and stiff stability. Chalmers Univ. of Technology, Sweden.
- Wright, K. (1970). Some relationships between implicit Runge-Kutta methods, collocation and Lanczos τ -methods and their stability properties. *BIT*, vol. 10, 217-227.
- Zadunaisky, P. (1976). On the estimation of errors propagated in the numerical integration of O.D.E.s. *Numer. Math.*, vol. 27, 21-39.

SOME CHARACTERISTICS OF ODE PROBLEMS
GENERATED BY THE NUMERICAL METHOD OF LINES

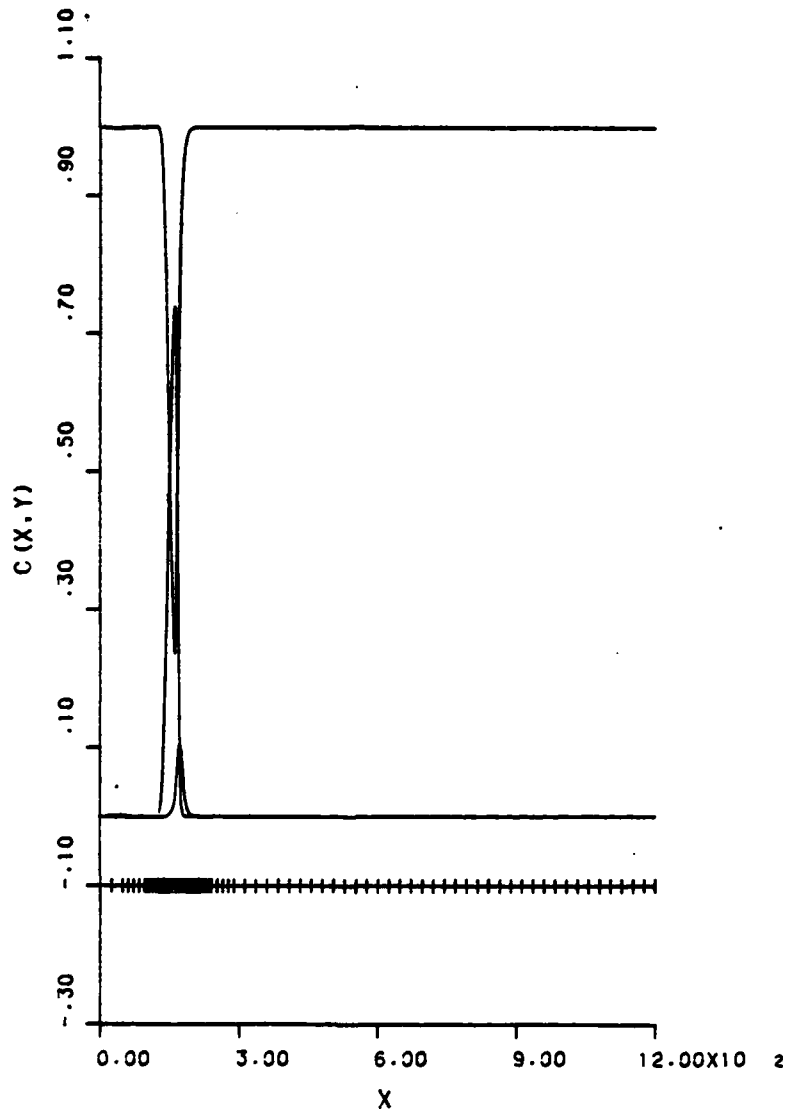
W. E. SCHIESSER
LEHIGH UNIVERSITY
BETHLEHEM, PA. USA



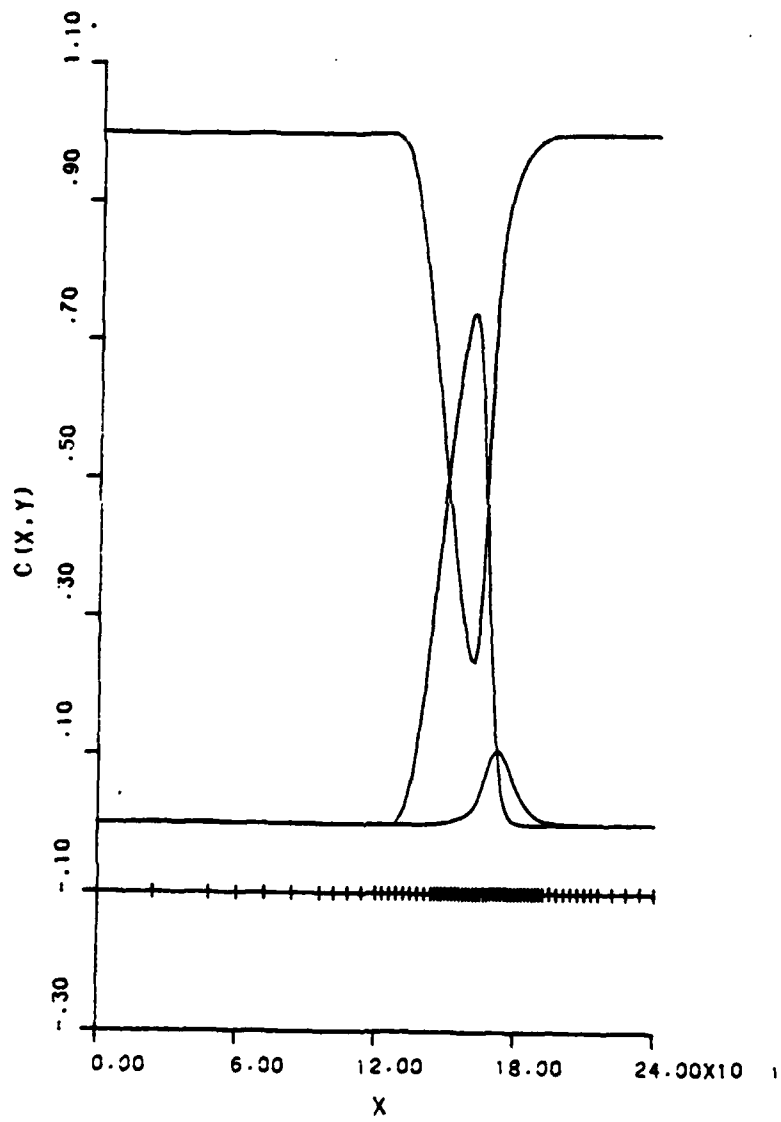
Y=0.00 .



ENLARGED

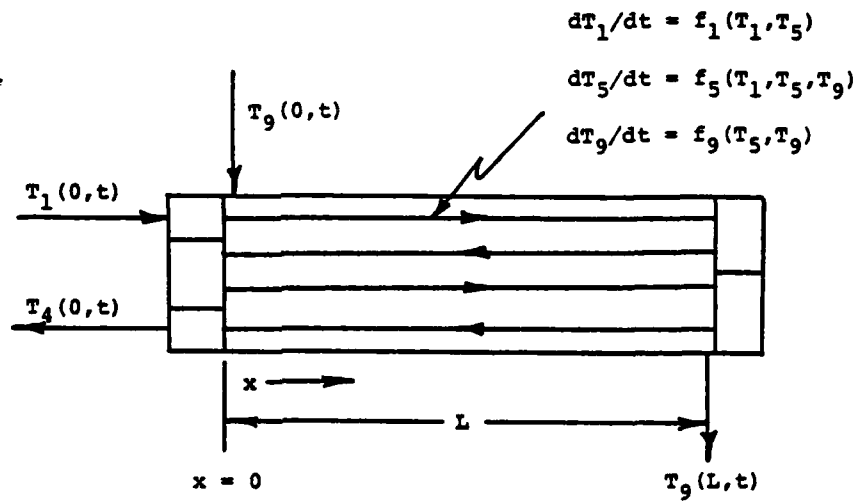


Y=080



ENLARGED

Four-pass Heat Exchanger
(ODE/PDE System)



Nine PDEs

Four tube fluid temperatures

$$T_1(x, t)$$

$$T_2(x, t)$$

$$T_3(x, t)$$

$$T_4(x, t)$$

Four metal temperatures

$$T_5(x, t)$$

$$T_6(x, t)$$

$$T_7(x, t)$$

$$T_8(x, t)$$

Shell fluid temperature

$$T_9(x, t)$$

11-point grid in x

$$\text{Number of ODEs} = 9 \times 11 + 1 = 100$$

SUBROUTINE INITAL

C...
 C... DYNAMIC MODEL OF A FOUR-PASS SHELL AND TUBE HEAT EXCHANGER WITH
 C... MAPPING OF THE MODEL JACOBIAN MATRIX
 C...
 C... A HEAT EXCHANGER IS A DISTRIBUTED SYSTEM IN WHICH CONVECTION AND
 C... TRANSFER OF HEAT ALONG THE EXCHANGER ARE THE PRINCIPAL MODES OF
 C... OPERATION. THUS THE SPATIAL DISTRIBUTION OF TEMPERATURE WITHIN
 C... THE EXCHANGER MUST BE TAKEN INTO ACCOUNT IN A REALISTIC ANALYSIS.
 C... IF THE TRANSIENT OPERATION OF AN EXCHANGER IS ALSO TO BE CON-
 C... SIDERED, THE MATHEMATICAL MODEL MUST NECESSARILY INVOLVE SPACE
 C... AND TIME, I.E., THE MODEL IS EXPRESSED IN TERMS OF PDES. ALSO,
 C... MIXING IN THE HEADERS OF THE EXCHANGER (THE ENTRANCE CHAMBER FOR
 C... EACH TUBE PASS) IS MODELED BY ODES WHICH SERVE AS BOUNDARY CON-
 C... DITIONS FOR THE PDES. THE ANALYSIS OF EXPERIMENTAL DATA FOR HEAT
 C... EXCHANGERS HAS INDICATED THAT THE MIXING IN THE HEADERS CANNOT BE
 C... NEGLECTED AND THUS A MIXED ODE/PDE MODEL IS REQUIRED. TO ILLUS-
 C... TRATE SOME OF THESE CONCEPTS, CONSIDER THE FOLLOWING SYSTEM OF
 C... NINE PDES AND FIVE PDES FOR A FOUR-PASS, SHELL AND TUBE HEAT
 C... EXCHANGER (SUBSCRIPTS T AND X DENOTE PARTIAL DERIVATIVES WITH
 C... RESPECT TO T AND X, RESPECTIVELY)

C...
 C...
 C...
$$T_1 = -C_1 \frac{\partial T_1}{\partial T} + C_3 (T_5 - T_1) \quad (1)$$

C...
 C...
 C...
$$T_2 = C_1 \frac{\partial T_2}{\partial T} + C_3 (T_6 - T_2) \quad (2)$$

C...
 C...
 C...
$$T_3 = -C_1 \frac{\partial T_3}{\partial T} + C_3 (T_7 - T_3) \quad (3)$$

C...
 C...
 C...
$$T_4 = C_1 \frac{\partial T_4}{\partial T} + C_3 (T_8 - T_4) \quad (4)$$

C...
 C...
 C...
$$T_5 = C_4 \frac{\partial T_5}{\partial T} + C_5 (T_9 - T_5) \quad (5)$$

C...
 C...
 C...
$$T_6 = C_4 \frac{\partial T_6}{\partial T} + C_5 (T_9 - T_6) \quad (6)$$

C...
 C...
 C...
$$T_7 = C_4 \frac{\partial T_7}{\partial T} + C_5 (T_9 - T_7) \quad (7)$$

C...
 C...
 C...
$$T_8 = C_4 \frac{\partial T_8}{\partial T} + C_5 (T_9 - T_8) \quad (8)$$

C...
 C...
 C...
$$T_9 = -C_2 \frac{\partial T_9}{\partial T} + C_6 ((T_5 - T_9) + (T_6 - T_9) + (T_7 - T_9) + (T_8 - T_9)) \quad (9)$$

C... THE NINE DEPENDENT VARIABLES, $T_1(x,t)$, $T_2(x,t)$, ..., $T_9(x,t)$ ARE
 C... TO BE COMPUTED. C_1, C_2, \dots, C_6 ARE GIVEN NUMERICAL CONSTANTS
 C... WHICH REFLECT THE THERMAL CAPACITANCES OF THE LIQUIDS FLOWING
 C... THROUGH THE EXCHANGER, THE THERMAL CAPACITANCE OF THE METAL IN
 C... THE EXCHANGER TUBES AND THE HEAT TRANSFER CHARACTERISTICS. EQUA-

C... TIONS (1) TO (4) AND (9) EACH REQUIRE A BOUNDARY CONDITION IN X
 C... AND ALL NINE EQUATIONS REQUIRE AN INITIAL CONDITION IN T. THESE
 C... WILL BE TAKEN AS
 C...
 C... $DT1(0,T)/DT = B1*(TI - T1(0,T))$ (10)
 C...
 C... $DT1(0,T)/DT = B1*(TI - T1(0,T))$ (10)
 C...
 C... $DT2(L,T)/DT = B2*(T1(L,T) - T2(L,T))$ (11)
 C...
 C... $DT3(0,T)/DT = B2*(T2(0,T) - T3(0,T))$ (12)
 C...
 C... $DT4(L,T)/DT = B2*(T3(L,T) - T4(L,T))$ (13)
 C...
 C... $T9(0,T) = TSI$ (14)
 C...
 C... $T1(X,0) = T2(X,0) = T3(X,0) = T4(X,0) =$
 C...
 C... $T5(X,0) = T6(X,0) = T7(X,0) = T8(X,0) =$ (15)
 C...
 C... $T9(X,0) = 0$
 C...
 C... FINALLY, ONE ORDINARY DIFFERENTIAL EQUATION WILL BE ADDED TO MODEL
 C... THE MIXING IN THE HEADER AT THE EXIT OF THE FOURTH PASS
 C...
 C... $DT0/DT = B1*(T4(0,T) - T0)$ (16)
 C...
 C... $T0(0) = 0$ (17)
 C...
 C... EQUATIONS (1) TO (17) CONSTITUTE THE COMPLETE SYSTEM (NINE PDES
 C... AND FIVE ODES). L, B1 AND B2 ARE THE EXCHANGER LENGTH AND MIXING
 C... TIME CONSTANTS FOR THE HEADERS, RESPECTIVELY. TI AND TSI ARE THE
 C... ENTERING TEMPERATURES OF THE TUBE SIDE AND SHELL SIDE FLUIDS,
 C... RESPECTIVELY.
 C...
 C... THE SOLUTION OF EQUATIONS (1) TO (17) IS BASED ON THE APPROXI-
 C... MATION OF THE DERIVATIVES IN X IN EQUATIONS (1) TO (4) AND (9)
 C... OVER A GRID OF 11 POINTS. AN APPROXIMATING ODE IS WRITTEN FOR
 C... EACH GRID POINT FOR EACH PDE. THUS THERE WILL BE A TOTAL OF $11*9$
 C... = 99 ODES APPROXIMATING EQUATIONS (1) TO (9). FINALLY, EQUATION
 C... (16) BRINGS THE TOTAL ODES TO 100.
 C...
 C... IN ORDER TO GAIN A PICTURE OF THE 100 ODE SYSTEM STRUCTURE, THE
 C... FOLLOWING PROGRAM ALSO CALLS SUBROUTINES JMAP AND EIGN TO MAP THE
 C... 100 ODE SYSTEM JACOBIAN MATRIX AND COMPUTE THE TEMPORAL EIGEN-
 C... VALUES, RESPECTIVELY. SUBROUTINE EIGN IN TURN CALLS IMSL ROUTINE
 C... EIGRF. IF EIGRF IS NOT AVAILABLE, THE CALL TO EIGN IN SUBROUTINE
 C... PRINT CAN BE CONVERTED TO A COMMENT (C IN COLUMN 1).
 C...
 C... THE USUAL FUNCTION OF SUBROUTINE INITAL, EVALUATION OF
 C... THE INITIAL CONDITIONS FOR THE DEPENDENT VARIABLES IN
 C... COMMON/Y/, IS ACCOMPLISHED BY A BLOCK DATA ROUTINE.
 C... THEREFORE INITAL IS ESSENTIALLY A DUMMY ROUTINE, AND
 C... DOES NOT REQUIRE THE USUAL COMMON/T/, /Y/ AND /F/.
 C...
 C... RETURN
 C... END

SUBROUTINE DERV

COMMON/T/TIME,NSTOP,NORUN

```

1      /Y/ T1(11), T2(11), T3(11), T4(11), T5(11),
2          T6(11), T7(11), T8(11), T9(11),   T0
3      /F/T1T(11),T2T(11),T3T(11),T4T(11),T5T(11),
4          T6T(11),T7T(11),T8T(11),T9T(11),   T9T
5      /S/T1X(11),T2X(11),T3X(11),T4X(11),T9X(11)
6      /C/   NG,    C1,    C2,    C3,    C4,    C5,
7          C6,    B1,    B2,    T1,    TSI,   XL

```

C...

C...

C... SET BOUNDARY CONDITION (14)
T9(1)=TSI

C...

C... COMPUTE THE FIRST DERIVATIVES WITH RESPECT TO X BY FIVE-POINT
CENTERED DIFFERENCES

```

C... CALL DSS004(0.,XL,NG,T1,T1X)
C... CALL DSS004(0.,XL,NG,T2,T2X)
C... CALL DSS004(0.,XL,NG,T3,T3X)
C... CALL DSS004(0.,XL,NG,T4,T4X)
C... CALL DSS004(0.,XL,NG,T9,T9X)

```

C...

C... ASSEMBLE THE PDES, EQUATIONS (1) TO (9)

```

DO 1 I=1,NG
T1T(I)=-C1*T1X(I) + C3*(T5(I)-T1(I))
T2T(I)= C1*T2X(I) + C3*(T6(I)-T2(I))
T3T(I)=-C1*T3X(I) + C3*(T7(I)-T3(I))
T4T(I)= C1*T4X(I) + C3*(T8(I)-T4(I))
T5T(I)=          C4*(T1(I)-T5(I))
1 T6T(I)=          + C5*(T9(I)-T5(I))
1          + C5*(T9(I)-T6(I))

T7T(I)=          C4*(T3(I)-T7(I))
1          + C5*(T9(I)-T7(I))
T8T(I)=          C4*(T4(I)-T8(I))
1          + C5*(T9(I)-T8(I))
T9T(I)=-C2*T9X(I) - 4.*C6*T9(I)
1          + C6*(T5(I)+T6(I)+T7(I)+T8(I))

```

1

CONTINUE

C...

C... EVALUATE THE ODE DERIVATIVES FROM EQUATIONS (18) AND (13) AND
EQUATION (16). NOTE THAT THESE DERIVATIVES MUST BE COMPUTED

C...

```

C... AFTER THOSE FOR THE PDES
T1T(1)=B1*( T1-T1(1))
T2T(NG)=B2*(T1(NG)-T2(NG))
T3T(1)=B2*(T2(1)-T3(1))
T4T(NG)=B2*(T3(NG)-T4(NG))
T9T(1)=0.
TOT =B1*(T4(1)- T0)
RETURN
END

```

SUBROUTINE PRINT(NI,NO)

COMMON/T/TIME,NSTOP,NORUN

```
1 /Y/ T1(11), T2(11), T3(11), T4(11), T5(11),
2     T6(11), T7(11), T8(11), T9(11),    TO
3 /F/T1T(11),T2T(11),T3Y(11),T4T(11),T5T(11),
4     T6T(11),T7T(11),T8T(11),T9T(11),    T9T
5 /S/T1X(11),T2X(11),T3X(11),T4X(11),T9X(11)
6 /C/   NG,    C1,    C2,    C3,    C4,    C5,
7     C6,    B1,    B2,    TI,    TSI,    XL
```

C... THIS SECTION OF SUBROUTINE PRINT

C... (1) CALLS SUBROUTINE JMAP TO MAP THE JACOBIAN MATRIX OF
C... 100 ODE SYSTEM.

C... (2) CALLS THE IMSL SUBROUTINE EIGRF TO COMPUTE THE TEMPORAL
C... EIGENVALUES OF THE ODE SYSTEM, AND OPTIONALLY, THE ASSOCI-
C... ATED EIGENVECTORS.

C... ABSOLUTE DIMENSIONING OF THE ARRAYS REQUIRED BY SUBROUTINES JMAP
C... (A, Y, YOLD, F, FOLD) AND EIGRF (EIGENV, WF)
C... DIMENSION A(100,100), EIGENV(100), WF(100),
1 Y(100), YOLD(100),
2 F(100), FOLD(100)
EQUIVALENCE (T1(1),Y(1)),(T1T(1),F(1))

C... THE COMPLEX TEMPORAL EIGENVALUES OF THE ODE SYSTEM ARE STORED IN
C... ARRAY EIGENV
C... COMPLEX EIGENV

C... MAP THE JACOBIAN MATRIX OF THE ODE SYSTEM DEFINED IN SUBROUTINE
C... DERV, AND COMPUTE ITS TEMPORAL EIGENVALUES
C... IF(TIME.GT.1.)GO TO 5
N=100
CALL JMAP(N,A,Y,YOLD,F,FOLD)

C... SUBROUTINE EIGEN (PART OF OSS/2) CALLS IMSL SUBROUTINE EIGRF TO
C... COMPUTE THE TEMPORAL EIGENVALUES, AND OPTIONALLY THE EIGENVECTORS,
C... OF THE 100 ODE SYSTEM JACOBIAN MATRIX. IF EIGRF IS NOT AVAILABLE,
C... THE FOLLOWING CALL TO EIGEN CAN BE CONVERTED TO A COMMENT (C IN
C... COLUMN 1)
CALL EIGEN(N,A,EIGENV,WF)


```

C...
C... PRINT A HEADING FOR THE NUMERICAL SOLUTION
      WRITE(NO,3)
3     FORMAT(1H1,9X,50H MIXED ORDINARY/PARTIAL DIFFERENTIAL EQUATION MODE
1L,/)
      WRITE(NO,1)C1,C2,C3,C4
1     FORMAT(
13X,4MC1 = ,E10.3,3X,4MC2 = ,E10.3,3X,4MC3 = ,E10.3,
23X,4MC4 = ,E10.3)
      WRITE(NO,2)C5,C6,81,82
2     FORMAT(
13X,4MC5 = ,E10.3,3X,4MC6 = ,E10.3,3X,4MB1 = ,E10.3,
23X,4MB2 = ,E10.3,/)
      WRITE(NO,4)
4     FORMAT(
1 6X,4MTIME,7X,7MT1(X=0),7X,8MT4(X=12),8X,2MT8,10X,
2 8MT9(X=12))
C...
C... PRINT THE SOLUTION
5     WRITE(NO,6)TIME,T1(1),T4(NG),T8,T9(NG)
6     FORMAT(E12.3,4E14.5)
      RETURN
      END

```


| I | REAL | IMAG | Z | WH |
|-----|--------|--------|-------|-------|
| 1 | -.918 | 0.888 | | |
| 2 | -.988 | 0.888 | | |
| 3 | -.888 | 2.898 | | |
| 4 | -.388 | -2.898 | .838 | 2.898 |
| 5 | -.882 | 2.197 | .839 | 2.898 |
| 6 | -.882 | -2.897 | .839 | 2.898 |
| 7 | -.881 | 2.396 | .839 | 2.198 |
| 8 | -.881 | -2.396 | .839 | 2.396 |
| 9 | -.881 | 2.896 | .839 | 2.897 |
| 10 | -.881 | -2.896 | .839 | 2.897 |
| 11 | -.289 | 1.744 | .117 | 1.746 |
| 12 | -.289 | -1.744 | .117 | 1.746 |
| 13 | -.198 | 1.744 | .112 | 1.748 |
| 14 | -.198 | -1.744 | .112 | 1.748 |
| 15 | -.281 | 1.737 | .119 | 1.749 |
| 16 | -.281 | -1.737 | .119 | 1.749 |
| 17 | -.286 | 1.739 | .114 | 1.791 |
| 18 | -.286 | -1.739 | .114 | 1.791 |
| 19 | -.496 | 1.199 | .389 | 1.278 |
| 20 | -.496 | -1.199 | .389 | 1.278 |
| 21 | -.496 | 1.188 | .389 | 1.287 |
| 22 | -.496 | -1.188 | .389 | 1.287 |
| 23 | -.449 | 1.163 | .388 | 1.247 |
| 24 | -.449 | -1.163 | .388 | 1.247 |
| 25 | -.478 | 1.149 | .388 | 1.241 |
| 26 | -.478 | -1.149 | .388 | 1.241 |
| 27 | -.797 | .737 | .734 | 1.886 |
| 28 | -.797 | -.737 | .734 | 1.886 |
| 29 | -.883 | .787 | .623 | .992 |
| 30 | -.883 | -.787 | .623 | .992 |
| 31 | -.498 | .992 | .637 | .769 |
| 32 | -.498 | -.992 | .637 | .769 |
| 33 | -.882 | .928 | .862 | 1.823 |
| 34 | -.882 | -.928 | .862 | 1.823 |
| 35 | -.979 | .449 | .981 | 1.882 |
| 36 | -.979 | -.449 | .981 | 1.882 |
| 37 | -1.221 | .289 | .973 | 1.299 |
| 38 | -1.221 | -.289 | .973 | 1.299 |
| 39 | -1.298 | .114 | .998 | 1.299 |
| 40 | -1.298 | -.114 | .998 | 1.299 |
| 41 | -.374 | .682 | .682 | .949 |
| 42 | -.374 | -.682 | .682 | .949 |
| 43 | -.316 | .898 | .898 | .316 |
| 44 | -.816 | -.316 | .898 | .316 |
| 45 | -.834 | .263 | .126 | .268 |
| 46 | -.834 | -.263 | .126 | .268 |
| 47 | -.299 | .217 | .788 | .338 |
| 48 | -.299 | -.217 | .788 | .338 |
| 49 | -.873 | .188 | .378 | .194 |
| 50 | -.873 | -.188 | .378 | .194 |
| 51 | -.184 | .118 | .687 | .191 |
| 52 | -.184 | -.118 | .687 | .191 |
| 53 | -.172 | .199 | .846 | .284 |
| 54 | -.172 | -.199 | .846 | .284 |
| 55 | -.863 | 0.888 | | |
| 56 | -.173 | 0.888 | | |
| 57 | -1.218 | 0.888 | | |
| 58 | -.788 | 0.888 | | |
| 59 | -.819 | .848 | .999 | .816 |
| 60 | -.819 | -.848 | .999 | .816 |
| 61 | -.948 | .119 | .992 | .998 |
| 62 | -.948 | -.119 | .992 | .998 |
| 63 | -.889 | .873 | .997 | .888 |
| 64 | -.889 | -.873 | .997 | .888 |
| 65 | -1.887 | 0.888 | | |
| 66 | -.943 | .839 | .999 | .943 |
| 67 | -.943 | -.839 | .999 | .943 |
| 68 | -.949 | .899 | .998 | .971 |
| 69 | -.949 | -.899 | .998 | .971 |
| 70 | -.949 | .849 | .999 | .948 |
| 71 | -.949 | -.849 | .999 | .948 |
| 72 | -.917 | .847 | .999 | .918 |
| 73 | -.917 | -.847 | .999 | .918 |
| 74 | -.937 | .844 | .999 | .938 |
| 75 | -.937 | -.844 | .999 | .938 |
| 76 | -.931 | .842 | .999 | .932 |
| 77 | -.931 | -.842 | .999 | .932 |
| 78 | -.933 | .836 | .999 | .934 |
| 79 | -.933 | -.836 | .999 | .934 |
| 80 | -.932 | .832 | .999 | .932 |
| 81 | -.932 | -.832 | .999 | .932 |
| 82 | -.921 | .831 | .999 | .922 |
| 83 | -.921 | -.831 | .999 | .922 |
| 84 | -.923 | .838 | .999 | .924 |
| 85 | -.923 | -.838 | .999 | .924 |
| 86 | -.928 | .829 | 1.888 | .928 |
| 87 | -.928 | -.829 | 1.888 | .928 |
| 88 | -.928 | .826 | 1.888 | .921 |
| 89 | -.928 | -.826 | 1.888 | .921 |
| 90 | -.928 | .827 | 1.888 | .938 |
| 91 | -.928 | -.827 | 1.888 | .938 |
| 92 | -.929 | .826 | 1.888 | .938 |
| 93 | -.929 | -.826 | 1.888 | .938 |
| 94 | -.927 | .821 | 1.888 | .927 |
| 95 | -.927 | -.821 | 1.888 | .927 |
| 96 | -.936 | 0.888 | | |
| 97 | -.918 | 0.888 | | |
| 98 | -.918 | 0.888 | | |
| 99 | -.988 | 0.888 | | |
| 100 | 0.888 | 0.888 | | |

MIXED ORDINARY/PARTIAL DIFFERENTIAL EQUATION MODEL

C1 = 1.980E+00 C2 = 3.000E-01 C3 = 1.300E-01 C4 = 4.761E-01
 C5 = 4.340E-01 C6 = 9.630E-03 B1 = 5.880E-01 B2 = 2.941E-01

| TIME | T1(X=0) | T4(X=12) | T0 | T9(X=12) |
|-----------|-------------|--------------|-------------|--------------|
| 0. | 0. | 0. | 0. | 0. |
| 2.500E+00 | 7.69564E+00 | 6.84342E-10 | 1.71951E-05 | 2.51592E-07 |
| 5.000E+00 | 9.46899E+00 | -2.70340E-07 | 8.52879E-04 | -6.12585E-05 |
| 7.500E+00 | 9.87764E+00 | 4.23113E-05 | 4.97729E-03 | 5.12851E-03 |
| 1.000E+01 | 9.97173E+00 | 2.65403E-03 | 1.40432E-02 | 6.31517E-02 |
| 1.250E+01 | 9.99344E+00 | 1.40254E-02 | 2.80189E-02 | 1.73538E-01 |
| 1.500E+01 | 9.99848E+00 | 3.56641E-02 | 4.61230E-02 | 3.10717E-01 |
| 1.750E+01 | 9.99965E+00 | 6.54722E-02 | 6.94672E-02 | 4.58387E-01 |
| 2.000E+01 | 9.99992E+00 | 1.05060E-01 | 1.01372E-01 | 6.08318E-01 |
| 2.250E+01 | 9.99998E+00 | 2.56215E-01 | 1.42910E-01 | 7.58408E-01 |
| 2.500E+01 | 1.00000E+01 | 6.38641E-01 | 1.92361E-01 | 9.12749E-01 |
| 2.750E+01 | 1.00000E+01 | 1.20361E+00 | 2.53735E-01 | 1.07609E+00 |
| 3.000E+01 | 1.00000E+01 | 1.82186E+00 | 3.83258E-01 | 1.24989E+00 |
| 3.250E+01 | 1.00000E+01 | 2.38733E+00 | 6.41656E-01 | 1.43238E+00 |
| 3.500E+01 | 1.00000E+01 | 2.85129E+00 | 1.01725E+00 | 1.61968E+00 |
| 3.750E+01 | 1.00000E+01 | 3.20799E+00 | 1.44583E+00 | 1.80670E+00 |
| 4.000E+01 | 1.00000E+01 | 3.47273E+00 | 1.86025E+00 | 1.98772E+00 |
| 4.250E+01 | 1.00000E+01 | 3.66745E+00 | 2.21936E+00 | 2.15717E+00 |
| 4.500E+01 | 1.00000E+01 | 3.81257E+00 | 2.50956E+00 | 2.31066E+00 |
| 4.750E+01 | 1.00000E+01 | 3.92404E+00 | 2.73481E+00 | 2.44606E+00 |
| 5.000E+01 | 1.00000E+01 | 4.01296E+00 | 2.90659E+00 | 2.56401E+00 |

SUBROUTINE JMAP(N,A,Y,YOLD,F,FOLD)

C...
C... SUBROUTINE JMAP MAPS THE JACOBIAN MATRIX OF AN NTH-ORDER SYSTEM OF
C... ALGEBRAIC EQUATIONS OR FIRST-ORDER ORDINARY DIFFERENTIAL EQUATIONS
C...
C... COPYRIGHT - LEHIGH UNIVERSITY, 1980
C...
C... AUTHORS
C...
C... G. R. DISSINGER
C... AND
C... W. E. SCHIESSER
C... WHITAKER NO. 5
C... LEHIGH UNIVERSITY
C... BETHLEHEM, PA 18015
C...
C... 215/861-4264
C...
C... ARGUMENT LIST
C...
C... N NUMBER OF ALGEBRAIC OR FIRST-ORDER ORDINARY DIFFEREN-
C... TIAL EQUATIONS (ODES) FOR WHICH THE JACOBIAN MATRIX IS
C... TO BE MAPPED, I.E., THE ORDER OF THE ALGEBRAIC OR ODE
C... SYSTEM (INPUT)
C...
C... A TWO-DIMENSIONAL ARRAY CONTAINING THE JACOBIAN MATRIX
C... OF THE NTH-ORDER ALGEBRAIC OR ODE SYSTEM (OUTPUT)
C...
C... Y ONE-DIMENSIONAL ARRAY OF N DEPENDENT VARIABLES SET TO
C... INITIAL VALUES BY A CALL TO SUBROUTINE INITIAL IN JMAP
C... (INPUT TO SUBROUTINE JMAP VIA SUBROUTINE INITIAL)
C...
C... YOLD ONE-DIMENSIONAL WORK ARRAY TO STORE THE N DEPENDENT
C... VARIABLES IN Y TEMPORARILY SO THAT WHEN THE INDIVIDUAL
C... Y'S ARE PERTURBED IN COMPUTING THE JACOBIAN MATRIX ELE-
C... MENTS, THE Y'S CAN THEN BE RESTORED TO THEIR ORIGINAL
C... VALUES
C...
C... F ONE-DIMENSIONAL ARRAY OF N DERIVATIVES OF THE DEPENDENT
C... VARIABLES SET BY A CALL TO SUBROUTINE DERV IN JMAP
C... (INPUT TO SUBROUTINE JMAP VIA SUBROUTINE DERV)
C...
C... FOLD ONE-DIMENSIONAL WORK ARRAY TO STORE THE N DERIVATIVES
C... IN F SO THAT A FINITE DIFFERENCE APPROXIMATION OF THE
C... INDIVIDUAL ELEMENTS OF THE JACOBIAN MATRIX CAN BE COM-
C... PUTED
C...
C... FROM THIS POINT ON, THE DISCUSSION IS IN TERMS OF FIRST-ORDER
C... ODES, BUT IT CAN AS WELL BE PRESENTED IN TERMS OF LINEAR OR NON-
C... ALGEBRAIC EQUATIONS, OR TRANSCENDENTAL EQUATIONS, OR A COMBINA-
C... TION OF ALL THREE TYPES OF EQUATIONS. THE ONLY REQUIREMENT IS
C... THAT INITIAL CONDITIONS ABOUT WHICH THE JACOBIAN MATRIX IS TO BE
C... COMPUTED MUST BE DEFINED IN SUBROUTINE INITIAL, AND THE EQUATIONS
C... THEMSELVES MUST BE PROGRAMMED IN SUBROUTINE DERV.

C... SUBROUTINES INITIAL AND DERV DEFINING THE ODE SYSTEM ARE SUPPLIED
 C... BY THE USER, AS WELL AS A CALLING PROGRAM FOR JMAP. THE MATHE-
 C... MATICAL CONCEPTS ARE EXPLAINED FURTHER IN THE FOLLOWING COMMENTS.
 C... THE CALCULATION OF THE INDIVIDUAL ELEMENTS OF THE JACOBIAN MATRIX
 C... BY A FINITE DIFFERENCE APPROXIMATION IS PROGRAMMED AS STATEMENT 14
 C... WITHIN DO LOOP 7.

C... THE SYSTEM OF ODES FOR WHICH THE JACOBIAN MATRIX IS MAPPED IS

$$\begin{aligned}
 \text{C... } & \dot{Y}_1/\dot{DT} = F_1(Y_1, Y_2, \dots, Y_N) \\
 \text{C... } & \dot{Y}_2/\dot{DT} = F_2(Y_1, Y_2, \dots, Y_N) \\
 \text{C... } & \vdots \\
 \text{C... } & \dot{Y}_N/\dot{DT} = F_N(Y_1, Y_2, \dots, Y_N)
 \end{aligned}
 \tag{1}$$

C... WHICH CAN BE SUMMARIZED IN VECTOR FORM AS

$$\dot{\bar{Y}}/\dot{DT} = \bar{F}(\bar{Y})
 \tag{2}$$

C... WHERE

$$\begin{aligned}
 \bar{Y} &= (Y_1, Y_2, \dots, Y_N)^T \\
 \bar{F} &= (F_1, F_2, \dots, F_N)^T
 \end{aligned}$$

C... SINCE THE DERIVATIVE VECTOR \bar{F} IS IN GENERAL A NONLINEAR FUNCTION
 C... OF THE DEPENDENT VARIABLE VECTOR \bar{Y} , A TAYLOR SERIES EXPANSION
 C... TRUNCATED AFTER LINEAR TERMS GIVES A LINEARIZED APPROXIMATION OF
 C... THE ORIGINAL SYSTEM

$$\dot{\bar{Y}}/\dot{DT} = \bar{J}\bar{Y}
 \tag{3}$$

C... WHERE \bar{J} IS THE JACOBIAN MATRIX OF THE ORIGINAL SYSTEM, I.E.,

$$\bar{J} = \begin{matrix}
 \begin{matrix} \dots & \dots \\ \cdot F & F \cdot \\ \cdot 11 & 1N \cdot \\ \cdot & \cdot \\ \cdot & F \cdot \\ \cdot & IJ \cdot \\ \cdot & \cdot \\ \cdot F & F \cdot \\ \cdot N1 & NN \cdot \\ \dots & \dots \end{matrix}
 \end{matrix}
 \tag{4}$$

C... F_{IJ} IS THE PARTIAL DERIVATIVE OF F_I WITH RESPECT TO Y_J . THUS THE
 C... JACOBIAN MATRIX IS SQUARE (IN X N).

C...
C... SUBROUTINE JMAP PRINTS A TWO-DIMENSIONAL MAP OF J WITH THE NUMBERS
C... 0 TO 9 INDICATING THE RELATIVE ORDER-OF-MAGNITUDE OF THE INDIVID-
C... UAL ELEMENTS OF THE MATRIX. THE VALUES OF THE ROW SUBSCRIPT I ARE
C... PRINTED DOWN THE LEFT SIDE OF THE MAP AND THE VALUES OF THE COLUMN
C... SUBSCRIPT J ARE PRINTED ACROSS THE TOP. THE MAP IS PRINTED IN
C... SECTIONS 100 COLUMNS WIDE. THUS IF THE DIFFERENTIAL EQUATION
C... SYSTEM IS GREATER THAN 100TH-ORDER, SUCCESSIVE SECTIONS OF THE MAP
C... WILL BE PRINTED VERTICALLY. THESE CAN THEN BE JOINED TOGETHER TO
C... MAKE UP THE COMPLETE MAP.
C...
C... THE N X N PARTIAL DERIVATIVES IN THE JACOBIAN MATRIX ARE COMPUTED
C... APPROXIMATELY BY A SIMPLE DIFFERENCING PROCEDURE. THE INITIAL
C...
C... VALUES OF Y REQUIRED TO START THE CALCULATION ARE OBTAINED BY A
C...
C... CALL TO SUBROUTINE INITAL. VALUES OF F ARE COMPUTED BY A SERIES
C... OF CALLS TO SUBROUTINE DERV. ALTHOUGH THESE SUBROUTINE NAMES
C... PERTAIN SPECIFICALLY TO DSS/2, JMAP CAN EASILY BE ADAPTED FOR USE
C... WITH ANY INITIAL-VALUE ODE INTEGRATION SYSTEM.
C...
C... COMMON/IO/ CONTAINS THE INPUT/OUTPUT UNIT (DEVICE) NUMBERS


```

SUBROUTINE DERV
COMMON/T/ TIME, NSTOP, NORUN
1  /Y/ Y(100)
3  /F/ F(100)
5  /S/ T1X(11), T2X(11), T3X(11), T4X(11), T9X(11)
6  /C/   NG,   C1,   C2,   C3,   C4,   C5,
7       C6,   B1,   B2,   TI,   TSI,  XL
COMMON/A/ T1(11), T2(11), T3(11), T4(11), T5(11), T6(11),
1         T7(11), T8(11), T9(11), T0
COMMON/B/ T1T(11), T2T(11), T3T(11), T4T(11), T5T(11), T6T(11),
1         T7T(11), T8T(11), T9T(11), T0T

C...
C... *****
C... TRANSLATE THE DEPENDENT VARIABLE VECTOR X IN COMMON/Y/ TO THE
C... PROBLEM DEPENDENT VARIABLES
      J=0
      DO 10 I=1, NG
      J=J+1
      T1(I)=Y(J)
      J=J+1
      T2(I)=Y(J)
      J=J+1
      T3(I)=Y(J)
      J=J+1
      T4(I)=Y(J)
      J=J+1
      T5(I)=Y(J)
      J=J+1
      T6(I)=Y(J)
      J=J+1
      T7(I)=Y(J)
      J=J+1
      T8(I)=Y(J)
      J=J+1
      T9(I)=Y(J)
10  CONTINUE
      J=J+1
      T0=Y(J)
C... *****
C...
C... SET BOUNDARY CONDITION (14)
C... T9(1)=TSI
C...
C... COMPUTE THE FIRST DERIVATIVES WITH RESPECT TO X BY FIVE-POINT
C... CENTERED DIFFERENCES
C... CALL DSS004(0., XL, NG, T1, T1X)
C... CALL DSS004(0., XL, NG, T2, T2X)
C... CALL DSS004(0., XL, NG, T3, T3X)
C... CALL DSS004(0., XL, NG, T4, T4X)
C... CALL DSS004(0., XL, NG, T9, T9X)

```

```

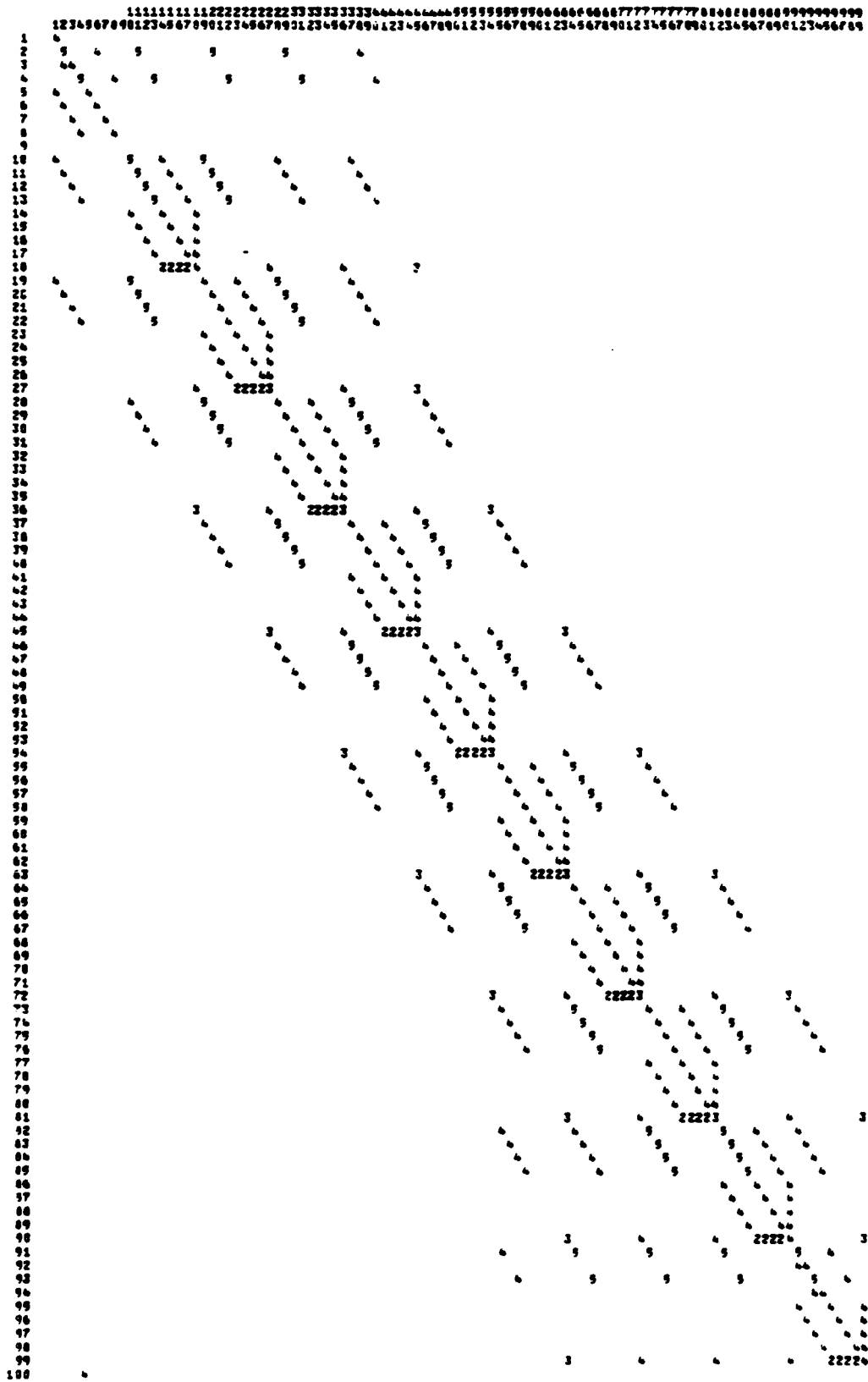
C...
C... ASSEMBLE THE PDES, EQUATIONS (1) TO (9)
      DO 1 I=1,NG
      T1T(I)=-C1*T1X(I) + C3*(T5(I)-T1(I))
      T2T(I)= C1*T2X(I) + C3*(T6(I)-T2(I))
      T3T(I)=-C1*T3X(I) + C3*(T7(I)-T3(I))
      T4T(I)= C1*T4X(I) + C3*(T8(I)-T4(I))
      T5T(I)=
      1          C4*(T1(I)-T5(I))
      T6T(I)=
      1          + C5*(T9(I)-T5(I))
      1          C4*(T2(I)-T6(I))
      T7T(I)=
      1          + C5*(T9(I)-T6(I))
      1          C4*(T3(I)-T7(I))
      T8T(I)=
      1          + C5*(T9(I)-T7(I))
      1          C4*(T4(I)-T8(I))
      T9T(I)=-C2*T9X(I) + 4.*C6*T9(I)
      1          + C6*(T5(I)+T6(I)+T7(I)+T8(I))
1     CONTINUE
C...
C... EVALUATE THE ODE DERIVATIVES FROM EQUATIONS (10) AND (13) AND
C... EQUATION (16). NOTE THAT THESE DERIVATIVES MUST BE COMPUTED
C... AFTER THOSE FOR THE PDES
      T1T( 1)=B1*( T1-T1( 1))
      T2T(NG)=B2*(T1(NG)-T2(NG))
      T3T( 1)=B2*(T2( 1)-T3( 1))
      T4T(NG)=B2*(T3(NG)-T4(NG))
      T9T( 1)=0.
      T0T =B1*(T4( 1)- T0)
C...
C... *****
C... TRANSLATE THE PROBLEM DEPENDENT VARIABLE DERIVATIVES TO THE
C... DEPENDENT VARIABLE DERIVATIVE VECTOR F IN COMMON/F/
      J=0
      DO 11 I=1,NG
      J=J+1
      F(J)=T1T(I)
      J=J+1
      F(J)=T2T(I)
      J=J+1
      F(J)=T3T(I)
      J=J+1
      F(J)=T4T(I)
      J=J+1
      F(J)=T5T(I)
      J=J+1
      F(J)=T6T(I)
      J=J+1
      F(J)=T7T(I)
      J=J+1
      F(J)=T8T(I)
      J=J+1
      F(J)=T9T(I)
11     CONTINUE
      J=J+1
      F(J)=T0T
C... *****
C...
      RETURN
      END

```

DEPENDENT VARIABLE COLUMN INDEX J (FOR YJ) IS PRINTED HORIZONTALLY

DERIVATIVE ROW INDEX I (FOR DYI/DX = PIIY1.Y2.....YJ.....YN) IS PRINTED VERTICALLY

JACOBIAN MATRIX ELEMENT IN THE MAP WITH INDICES I,J IS FOR PPI/PYJ WHERE P DENOTES A PARTIAL DERIVATIVE



| I | REAL | IMAG | Z | MM |
|-----|--------|--------|-------|-------|
| 1 | -.918 | 0.000 | | |
| 2 | -.588 | 0.000 | | |
| 3 | -.062 | 2.097 | .039 | 2.098 |
| 4 | -.382 | -2.097 | .039 | 2.098 |
| 5 | -.000 | 2.096 | .038 | 2.098 |
| 6 | -.000 | -2.096 | .038 | 2.098 |
| 7 | -.001 | 2.096 | .039 | 2.097 |
| 8 | -.001 | -2.096 | .039 | 2.097 |
| 9 | -.001 | 2.094 | .039 | 2.096 |
| 10 | -.001 | -2.094 | .039 | 2.096 |
| 11 | -.196 | 1.744 | .112 | 1.795 |
| 12 | -.196 | -1.744 | .112 | 1.795 |
| 13 | -.205 | 1.744 | .117 | 1.796 |
| 14 | -.205 | -1.744 | .117 | 1.796 |
| 15 | -.200 | 1.739 | .114 | 1.791 |
| 16 | -.200 | -1.739 | .114 | 1.791 |
| 17 | -.231 | 1.737 | .115 | 1.749 |
| 18 | -.201 | -1.737 | .115 | 1.749 |
| 19 | -.494 | 1.195 | .399 | 1.270 |
| 20 | -.494 | -1.195 | .399 | 1.270 |
| 21 | -.496 | 1.188 | .385 | 1.267 |
| 22 | -.496 | -1.188 | .385 | 1.267 |
| 23 | -.449 | 1.163 | .368 | 1.247 |
| 24 | -.449 | -1.163 | .368 | 1.247 |
| 25 | -.470 | 1.146 | .385 | 1.241 |
| 26 | -.470 | -1.146 | .385 | 1.241 |
| 27 | -.797 | .737 | .734 | 1.086 |
| 28 | -.797 | -.737 | .734 | 1.086 |
| 29 | -.683 | .737 | .633 | .952 |
| 30 | -.683 | -.737 | .633 | .952 |
| 31 | -1.221 | .209 | .973 | 1.255 |
| 32 | -1.221 | -.209 | .973 | 1.255 |
| 33 | -1.290 | .114 | .996 | 1.298 |
| 34 | -1.290 | -.114 | .996 | 1.298 |
| 35 | -.975 | .469 | .901 | 1.082 |
| 36 | -.975 | -.469 | .901 | 1.082 |
| 37 | -.842 | .928 | .862 | 1.023 |
| 38 | -.842 | -.928 | .862 | 1.023 |
| 39 | -.498 | .992 | .637 | .769 |
| 40 | -.498 | -.992 | .637 | .769 |
| 41 | -.374 | .402 | .602 | .549 |
| 42 | -.374 | -.402 | .602 | .549 |
| 43 | -.016 | .316 | .050 | .316 |
| 44 | -.016 | -.316 | .050 | .316 |
| 45 | -.034 | .263 | .126 | .265 |
| 46 | -.034 | -.263 | .126 | .265 |
| 47 | -.259 | .217 | .766 | .338 |
| 48 | -.259 | -.217 | .766 | .338 |
| 49 | -.073 | .100 | .376 | .194 |
| 50 | -.073 | -.100 | .376 | .194 |
| 51 | -.003 | 0.000 | | |
| 52 | -.104 | .110 | .607 | .151 |
| 53 | -.104 | -.110 | .607 | .151 |
| 54 | -.172 | .109 | .844 | .204 |
| 55 | -.172 | -.109 | .844 | .204 |
| 56 | -.173 | 0.000 | | |
| 57 | -1.218 | 0.000 | | |
| 58 | -.786 | 0.000 | | |
| 59 | -.948 | .119 | .992 | .996 |
| 60 | -.948 | -.119 | .992 | .996 |
| 61 | -.815 | .040 | .999 | .816 |
| 62 | -.815 | -.040 | .999 | .816 |
| 63 | -.805 | .073 | .997 | .800 |
| 64 | -.805 | -.073 | .997 | .800 |
| 65 | -1.007 | 0.000 | | |
| 66 | -.993 | .035 | .999 | .993 |
| 67 | -.993 | -.035 | .999 | .993 |
| 68 | -.969 | .055 | .998 | .971 |
| 69 | -.969 | -.055 | .998 | .971 |
| 70 | -.949 | .045 | .999 | .950 |
| 71 | -.949 | -.045 | .999 | .950 |
| 72 | -.917 | .047 | .999 | .918 |
| 73 | -.917 | -.047 | .999 | .918 |
| 74 | -.937 | .044 | .999 | .938 |
| 75 | -.937 | -.044 | .999 | .938 |
| 76 | -.931 | .042 | .999 | .932 |
| 77 | -.931 | -.042 | .999 | .932 |
| 78 | -.933 | .036 | .999 | .934 |
| 79 | -.933 | -.036 | .999 | .934 |
| 80 | -.936 | 0.000 | | |
| 81 | -.932 | .032 | .999 | .932 |
| 82 | -.932 | -.032 | .999 | .932 |
| 83 | -.921 | .031 | .999 | .922 |
| 84 | -.921 | -.031 | .999 | .922 |
| 85 | -.923 | .030 | .999 | .924 |
| 86 | -.923 | -.030 | .999 | .924 |
| 87 | -.920 | .026 | 1.000 | .920 |
| 88 | -.920 | -.026 | 1.000 | .920 |
| 89 | -.920 | .026 | 1.000 | .921 |
| 90 | -.920 | -.026 | 1.000 | .921 |
| 91 | -.927 | .021 | 1.000 | .927 |
| 92 | -.927 | -.021 | 1.000 | .927 |
| 93 | -.930 | .027 | 1.000 | .930 |
| 94 | -.930 | -.027 | 1.000 | .930 |
| 95 | -.929 | .026 | 1.000 | .930 |
| 96 | -.929 | -.026 | 1.000 | .930 |
| 97 | -.910 | 0.000 | | |
| 98 | -.910 | 0.000 | | |
| 99 | -.500 | 0.000 | | |
| 100 | 0.000 | 0.000 | | |

MIXED ORDINARY/PARTIAL DIFFERENTIAL EQUATION MODEL

C1 = 1.980E+00 C2 = 3.000E-01 C3 = 1.300E-01 C4 = 4.760E-01
 C5 = 4.340E-01 C6 = 9.630E-03 B1 = 5.880E-01 B2 = 2.940E-01

| TIME | T1(X=0) | T4(X=12) | T0 | T9(X=12) |
|-----------|-------------|--------------|-------------|--------------|
| 0. | 0. | 0. | 0. | 0. |
| 2.500E+00 | 7.70630E+00 | 6.60325E-10 | 1.63208E-05 | 2.89873E-07 |
| 5.000E+00 | 9.47145E+00 | -2.52796E-07 | 8.46567E-04 | -5.90177E-05 |
| 7.500E+00 | 9.87820E+00 | 3.95604E-05 | 4.96740E-03 | 5.01064E-03 |
| 1.000E+01 | 9.97186E+00 | 2.63610E-03 | 1.40335E-02 | 6.30533E-02 |
| 1.250E+01 | 9.99357E+00 | 1.39411E-02 | 2.79863E-02 | 1.73425E-01 |
| 1.500E+01 | 9.99851E+00 | 3.55964E-02 | 4.60878E-02 | 3.10520E-01 |
| 1.750E+01 | 9.99965E+00 | 6.54438E-02 | 6.94014E-02 | 4.58362E-01 |
| 2.000E+01 | 9.99992E+00 | 1.04453E-01 | 1.01294E-01 | 6.08313E-01 |
| 2.250E+01 | 9.99998E+00 | 2.54327E-01 | 1.42843E-01 | 7.58368E-01 |
| 2.500E+01 | 1.00000E+01 | 6.36993E-01 | 1.92319E-01 | 9.12573E-01 |
| 2.750E+01 | 1.00000E+01 | 1.20306E+00 | 2.53294E-01 | 1.07601E+00 |
| 3.000E+01 | 1.00000E+01 | 1.82226E+00 | 3.82244E-01 | 1.24983E+00 |
| 3.250E+01 | 1.00000E+01 | 2.38814E+00 | 6.40658E-01 | 1.43235E+00 |
| 3.500E+01 | 1.00000E+01 | 2.85217E+00 | 1.01675E+00 | 1.61969E+00 |
| 3.750E+01 | 1.00000E+01 | 3.20876E+00 | 1.44590E+00 | 1.80674E+00 |
| 4.000E+01 | 1.00000E+01 | 3.47330E+00 | 1.86068E+00 | 1.98780E+00 |
| 4.250E+01 | 1.00000E+01 | 3.66781E+00 | 2.21990E+00 | 2.15728E+00 |
| 4.500E+01 | 1.00000E+01 | 3.81278E+00 | 2.51009E+00 | 2.31079E+00 |
| 4.750E+01 | 1.00000E+01 | 3.92419E+00 | 2.73525E+00 | 2.44619E+00 |
| 5.000E+01 | 1.00000E+01 | 4.01307E+00 | 2.90692E+00 | 2.56412E+00 |

```

SUBROUTINE DERV
COMMON/T/TIME,NSTOP,NORUN
1   /Y/ Y(100)
3   /F/ F(100)
5   /S/T1X(11),T2X(11),T3X(11),T4X(11),T9X(11)
6   /C/   NG,   C1,   C2,   C3,   C4,   C5,
7         C6,   B1,   B2,   T1,   TSI,  XL
COMMON/A/ T1(11), T2(11), T3(11), T4(11), T5(11), T6(11),
1         T7(11), T8(11), T9(11),   T0
COMMON/B/T1T(11),T2T(11),T3T(11),T4T(11),T5T(11),T6T(11),
1         T7T(11),T8T(11),T9T(11),   TOT
C... *****
C... TRANSLATE THE DEPENDENT VARIABLE VECTOR X IN COMMON/Y/ TO THE
C... PROBLEM DEPENDENT VARIABLES
      J=0
      DO 10 I=1,NG
      J=J+1
      T1(I)=Y(J)
      J=J+1
      T5(I)=Y(J)
      J=J+1
      T2(I)=Y(J)
      J=J+1
      T6(I)=Y(J)
      J=J+1
      T9(I)=Y(J)
      J=J+1
      T3(I)=Y(J)
      J=J+1
      T7(I)=Y(J)
      J=J+1
      T4(I)=Y(J)
      J=J+1
      T8(I)=Y(J)
10   CONTINUE
      J=J+1
      T0=Y(J)
C... *****
C...
C... SET BOUNDARY CONDITION (14)
      T9(1)=TSI
C...
C... COMPUTE THE FIRST DERIVATIVES WITH RESPECT TO X BY FIVE-POINT
C... CENTERED DIFFERENCES
      CALL DSS004(0.,XL,NG,T1,T1X)
      CALL DSS004(0.,XL,NG,T2,T2X)
      CALL DSS004(0.,XL,NG,T3,T3X)
      CALL DSS004(0.,XL,NG,T4,T4X)
      CALL DSS004(0.,XL,NG,T9,T9X)

```

```

C...
C... ASSEMBLE THE PDES, EQUATIONS (1) TO (9)
DO 1 I=1,NG
T1T(I)=-C1*T1X(I) + C3*(T5(I)-T1(I))
T2T(I)= C1*T2X(I) + C3*(T6(I)-T2(I))
T3T(I)=-C1*T3X(I) + C3*(T7(I)-T3(I))
T4T(I)= C1*T4X(I) + C3*(T8(I)-T4(I))
T5T(I)= C4*(T1(I)-T5(I))
1 + C5*(T9(I)-T5(I))
T6T(I)= C4*(T2(I)-T6(I))
1 + C5*(T9(I)-T6(I))
T7T(I)= C4*(T3(I)-T7(I))
1 + C5*(T9(I)-T7(I))
T8T(I)= C4*(T4(I)-T8(I))
1 + C5*(T9(I)-T8(I))
T9T(I)=-C2*T9X(I) - 4.*C6*T9(I)
1 + C6*(T5(I)+T6(I)+T7(I)+T8(I))
1 CONTINUE
C...
C... EVALUATE THE ODE DERIVATIVES FROM EQUATIONS (10) AND (13) AND
C... EQUATION (16). NOTE THAT THESE DERIVATIVES MUST BE COMPUTED
C... AFTER THOSE FOR THE PDES
T1T( 1)=81*( TI-T1( 1))
T2T(NG)=82*(T1(NG)-T2(NG))
T3T( 1)=82*(T2( 1)-T3( 1))
T4T(NG)=82*(T3(NG)-T4(NG))
T9T( 1)=0.
T8T =81*(T4( 1)- T0)
C...
C... *****
C... TRANSLATE THE PROBLEM DEPENDENT VARIABLE DERIVATIVES TO THE
C... DEPENDENT VARIABLE DERIVATIVE VECTOR F IN COMMON/F/
J=0
DO 11 I=1,NG
J=J+1
F(J)=T1T(I)
J=J+1
F(J)=T5T(I)
J=J+1
F(J)=T2T(I)
J=J+1
F(J)=T6T(I)
J=J+1
F(J)=T9T(I)
J=J+1
F(J)=T3T(I)
J=J+1
F(J)=T7T(I)
J=J+1
F(J)=T4T(I)
J=J+1
F(J)=T8T(I)
11 CONTINUE
J=J+1
F(J)=T0T
C... *****
C...
C... RETURN
END

```


| I | REAL | IMAG | Z | MM |
|-----|--------|--------|-------|-------|
| 1 | -.999 | 0.000 | | |
| 2 | -.918 | 0.000 | | |
| 3 | -.802 | 2.097 | .039 | 2.098 |
| 4 | -.682 | -2.097 | .039 | 2.098 |
| 5 | -.558 | 2.096 | .039 | 2.099 |
| 6 | -.438 | -2.096 | .038 | 2.098 |
| 7 | -.311 | 2.096 | .039 | 2.098 |
| 8 | -.181 | -2.096 | .039 | 2.098 |
| 9 | -.051 | 2.096 | .039 | 2.097 |
| 10 | -.081 | -2.096 | .039 | 2.097 |
| 11 | -.208 | 1.746 | .117 | 1.796 |
| 12 | -.208 | -1.746 | .117 | 1.796 |
| 13 | -.196 | 1.746 | .112 | 1.799 |
| 14 | -.196 | -1.746 | .112 | 1.799 |
| 15 | -.201 | 1.737 | .119 | 1.749 |
| 16 | -.201 | -1.737 | .119 | 1.749 |
| 17 | -.200 | 1.739 | .114 | 1.751 |
| 18 | -.200 | -1.739 | .114 | 1.751 |
| 19 | -.406 | 1.108 | .309 | 1.207 |
| 20 | -.406 | -1.108 | .309 | 1.207 |
| 21 | -.496 | 1.199 | .399 | 1.279 |
| 22 | -.496 | -1.199 | .399 | 1.279 |
| 23 | -.449 | 1.163 | .368 | 1.247 |
| 24 | -.449 | -1.163 | .368 | 1.247 |
| 25 | -.478 | 1.165 | .369 | 1.241 |
| 26 | -.478 | -1.165 | .369 | 1.241 |
| 27 | -.797 | .737 | .734 | 1.086 |
| 28 | -.797 | -.737 | .734 | 1.086 |
| 29 | -.663 | .737 | .633 | .992 |
| 30 | -.663 | -.737 | .633 | .992 |
| 31 | -1.221 | .209 | .973 | 1.299 |
| 32 | -1.221 | -.209 | .973 | 1.299 |
| 33 | -1.290 | .114 | .996 | 1.299 |
| 34 | -1.290 | -.114 | .996 | 1.299 |
| 35 | -.979 | .469 | .981 | 1.002 |
| 36 | -.979 | -.469 | .981 | 1.002 |
| 37 | -.802 | .520 | .862 | 1.023 |
| 38 | -.802 | -.520 | .862 | 1.023 |
| 39 | -.498 | .992 | .637 | .769 |
| 40 | -.498 | -.992 | .637 | .769 |
| 41 | -.376 | .402 | .602 | .969 |
| 42 | -.376 | -.402 | .602 | .969 |
| 43 | -.816 | .316 | .890 | .316 |
| 44 | -.816 | -.316 | .890 | .316 |
| 45 | -.836 | .263 | .126 | .268 |
| 46 | -.836 | -.263 | .126 | .268 |
| 47 | -.299 | .217 | .769 | .339 |
| 48 | -.299 | -.217 | .769 | .339 |
| 49 | -.073 | .180 | .376 | .194 |
| 50 | -.073 | -.180 | .376 | .194 |
| 51 | -.104 | .110 | .607 | .191 |
| 52 | -.104 | -.110 | .607 | .191 |
| 53 | -.172 | .109 | .646 | .204 |
| 54 | -.172 | -.109 | .646 | .204 |
| 55 | -.863 | 0.000 | | |
| 56 | -.173 | 0.000 | | |
| 57 | -1.210 | 0.000 | | |
| 58 | -.706 | 0.000 | | |
| 59 | -.815 | .040 | .999 | .816 |
| 60 | -.815 | -.040 | .999 | .816 |
| 61 | -.940 | .119 | .992 | .996 |
| 62 | -.940 | -.119 | .992 | .996 |
| 63 | -.889 | .073 | .997 | .888 |
| 64 | -.889 | -.073 | .997 | .888 |
| 65 | -1.007 | 0.000 | | |
| 66 | -.993 | .439 | .999 | .993 |
| 67 | -.993 | -.439 | .999 | .993 |
| 68 | -.969 | .099 | .990 | .971 |
| 69 | -.969 | -.099 | .990 | .971 |
| 70 | -.949 | .046 | .999 | .990 |
| 71 | -.949 | -.046 | .999 | .990 |
| 72 | -.817 | .067 | .999 | .910 |
| 73 | -.817 | -.067 | .999 | .910 |
| 74 | -.937 | .044 | .999 | .913 |
| 75 | -.937 | -.044 | .999 | .913 |
| 76 | -.931 | .042 | .999 | .932 |
| 77 | -.931 | -.042 | .999 | .932 |
| 78 | -.933 | .036 | .999 | .936 |
| 79 | -.933 | -.036 | .999 | .936 |
| 80 | -.936 | 0.000 | | |
| 81 | -.932 | .032 | .999 | .932 |
| 82 | -.932 | -.032 | .999 | .932 |
| 83 | -.927 | .021 | 1.000 | .927 |
| 84 | -.927 | -.021 | 1.000 | .927 |
| 85 | -.930 | .027 | 1.000 | .930 |
| 86 | -.930 | -.027 | 1.000 | .930 |
| 87 | -.929 | .026 | 1.000 | .930 |
| 88 | -.929 | -.026 | 1.000 | .930 |
| 89 | -.921 | .031 | .999 | .922 |
| 90 | -.921 | -.031 | .999 | .922 |
| 91 | -.923 | .030 | .999 | .924 |
| 92 | -.923 | -.030 | .999 | .924 |
| 93 | -.920 | .026 | 1.000 | .921 |
| 94 | -.920 | -.026 | 1.000 | .921 |
| 95 | -.920 | .025 | 1.000 | .920 |
| 96 | -.920 | -.025 | 1.000 | .920 |
| 97 | -.910 | 0.000 | | |
| 98 | -.910 | 0.000 | | |
| 99 | -.900 | 0.000 | | |
| 100 | 0.000 | 0.000 | | |

MIXED ORDINARY/PARTIAL DIFFERENTIAL EQUATION MODEL

C1 = 1.980E+00 C2 = 3.000E-01 C3 = 1.300E-01 C4 = 4.760E-01
 C5 = 4.340E-01 C6 = 9.630E-03 B1 = 5.880E-01 B2 = 2.940E-01

| TIME | T1(X=0) | T4(X=12) | T0 | T9(X=12) |
|-----------|-------------|--------------|-------------|--------------|
| 0. | 0. | 0. | 0. | 0. |
| 2.500E+00 | 7.70630E+00 | 6.60325E-10 | 1.63208E-05 | 2.89873E-07 |
| 5.000E+00 | 9.47145E+00 | -2.52796E-07 | 9.46567E-04 | -5.90177E-05 |
| 7.500E+00 | 9.87820E+00 | 3.95604E-05 | 4.96740E-03 | 5.01064E-03 |
| 1.000E+01 | 9.97186E+00 | 2.63610E-03 | 1.40335E-02 | 6.30533E-02 |
| 1.250E+01 | 9.99357E+00 | 1.39411E-02 | 2.79863E-02 | 1.73425E-01 |
| 1.500E+01 | 9.99951E+00 | 3.55964E-02 | 4.60979E-02 | 3.10620E-01 |
| 1.750E+01 | 9.99965E+00 | 6.54438E-02 | 6.94014E-02 | 4.58362E-01 |
| 2.000E+01 | 9.99992E+00 | 1.04453E-01 | 1.01294E-01 | 6.08313E-01 |
| 2.250E+01 | 9.99998E+00 | 2.54327E-01 | 1.42843E-01 | 7.58368E-01 |
| 2.500E+01 | 1.00000E+01 | 6.36993E-01 | 1.92319E-01 | 9.12673E-01 |
| 2.750E+01 | 1.00000E+01 | 1.20306E+00 | 2.53294E-01 | 1.07601E+00 |
| 3.000E+01 | 1.00000E+01 | 1.82226E+00 | 3.82244E-01 | 1.24983E+00 |
| 3.250E+01 | 1.00000E+01 | 2.38814E+00 | 6.40658E-01 | 1.43235E+00 |
| 3.500E+01 | 1.00000E+01 | 2.85217E+00 | 1.01675E+00 | 1.61969E+00 |
| 3.750E+01 | 1.00000E+01 | 3.20876E+00 | 1.44590E+00 | 1.80674E+00 |
| 4.000E+01 | 1.00000E+01 | 3.47330E+00 | 1.86068E+00 | 1.98780E+00 |
| 4.250E+01 | 1.00000E+01 | 3.66781E+00 | 2.21990E+00 | 2.15728E+00 |
| 4.500E+01 | 1.00000E+01 | 3.81278E+00 | 2.51009E+00 | 2.31079E+00 |
| 4.750E+01 | 1.00000E+01 | 3.92419E+00 | 2.73525E+00 | 2.44619E+00 |
| 5.000E+01 | 1.00000E+01 | 4.01307E+00 | 2.90692E+00 | 2.56412E+00 |

A MAP OF THE JACOBIAN MATRIX INDICATES:

- (1) THE ODE SYSTEM OVERALL STRUCTURE
(BANDEDNESS, SPARSENESS, ETC.)
- (2) ORDER OF MAGNITUDE OF THE MATRIX
ELEMENTS ..
- (3) THE DETAILED RELATIONSHIPS BETWEEN
DEPENDENT VARIABLES AND DERIVATIVES
- (4) THE DEGREE OF NONLINEARITY AS REFLECTED
BY CHANGES IN THE MATRIX

THE TABULATION OF THE ODE SYSTEM EIGENVALUES
INDICATES:

- (1) STABILITY
- (2) TIME SCALE
- (3) STIFFNESS
- (4) CONTRIBUTIONS OF INDIVIDUAL ODES
- (5) PARAMETER SENSITIVITIES

THE NUMERICAL METHOD OF LINES
FOR ODE/PDE SYSTEMS

$$\bar{u}_T = \bar{F}(\bar{x}, T, \bar{u}, \bar{u}_{\bar{x}}, \bar{u}_{\bar{x}\bar{x}}, \bar{u}_{\bar{x}\bar{x}\bar{x}}, \dots)$$

$$\bar{g}(\bar{x}_B, T, \bar{u}, \bar{u}_{\bar{x}}, \bar{u}_{\bar{x}\bar{x}}, \dots) = \bar{0}$$

$$\bar{u}(\bar{x}, T_0) = \bar{u}_0(\bar{x})$$

NUMBER OF PDES = M

($\bar{u}(\bar{x}, T)$) IS AN M VECTOR

$$\bar{u}(i\Delta\bar{x}, T), \quad i = 0, 1, 2, \dots, N$$

NUMBER OF ODES = MN^D

$$D = 1, 2, 3$$

$$\bar{u}_T(i\Delta\bar{x}, T), \quad i = 0, 1, 2, \dots, N$$

Burger's Equation

$$u_t = \mu u_{xx} - uu_x$$

$$u(x,0) = \phi(x,0)$$

$$u(0,t) = \phi(0,t)$$

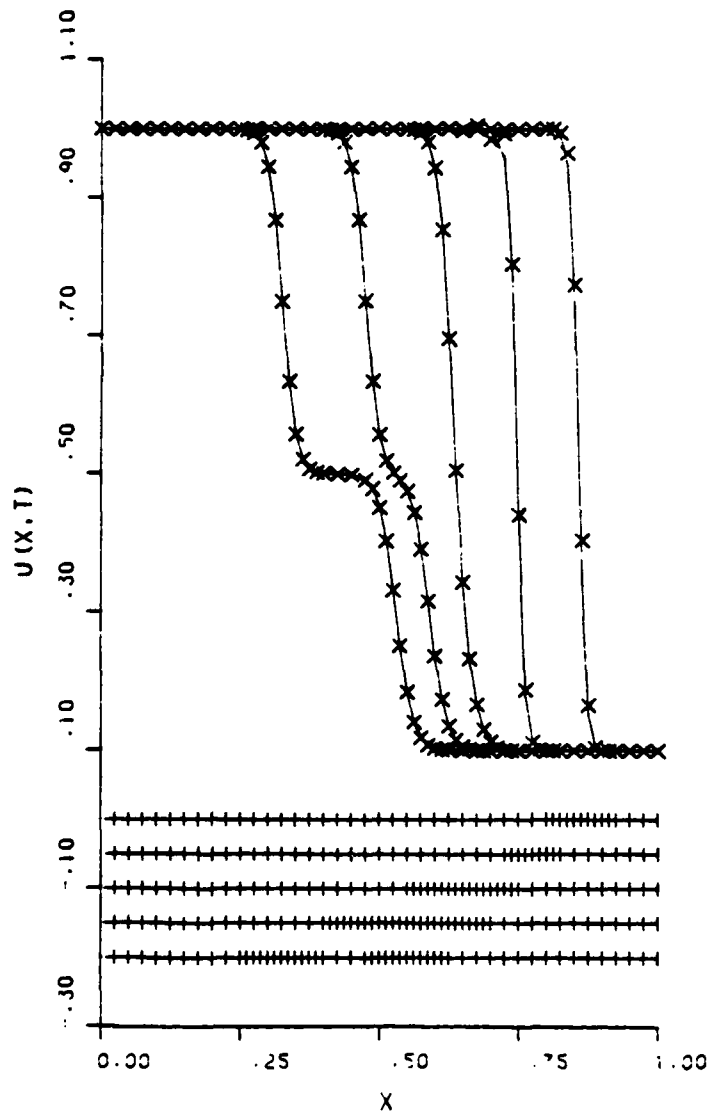
$$u(1,t) = \phi(1,t)$$

$$\phi(x,t) = \frac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}}$$

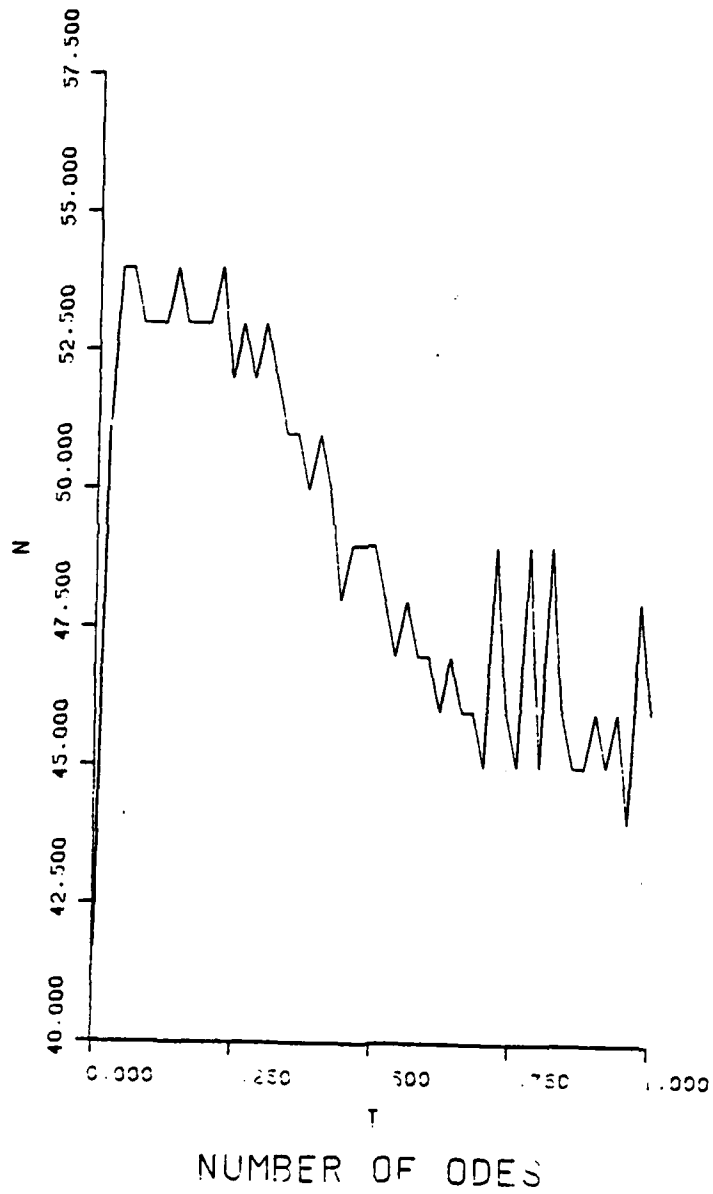
$$A = (0.05/\mu)(x - 0.5 + 4.95t)$$

$$B = (0.25/\mu)(x - 0.5 + 0.75t)$$

$$C = (0.5/\mu)(x - 0.375)$$



BURGER S EQUATION

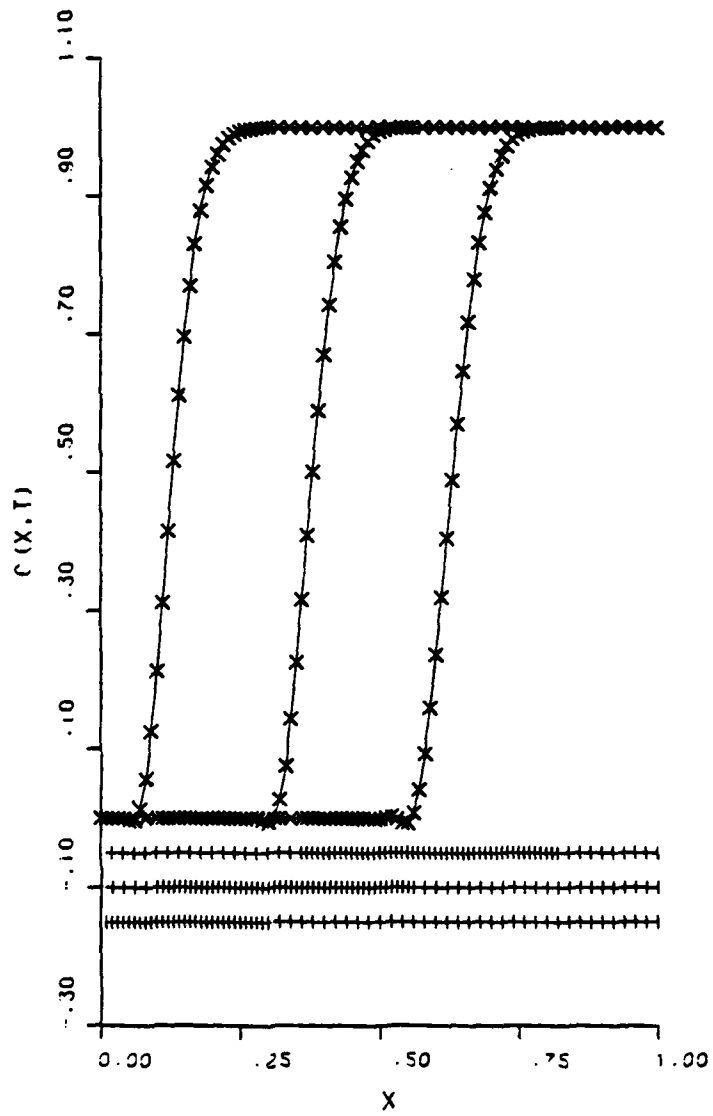


Single-solute Chromatography Equation

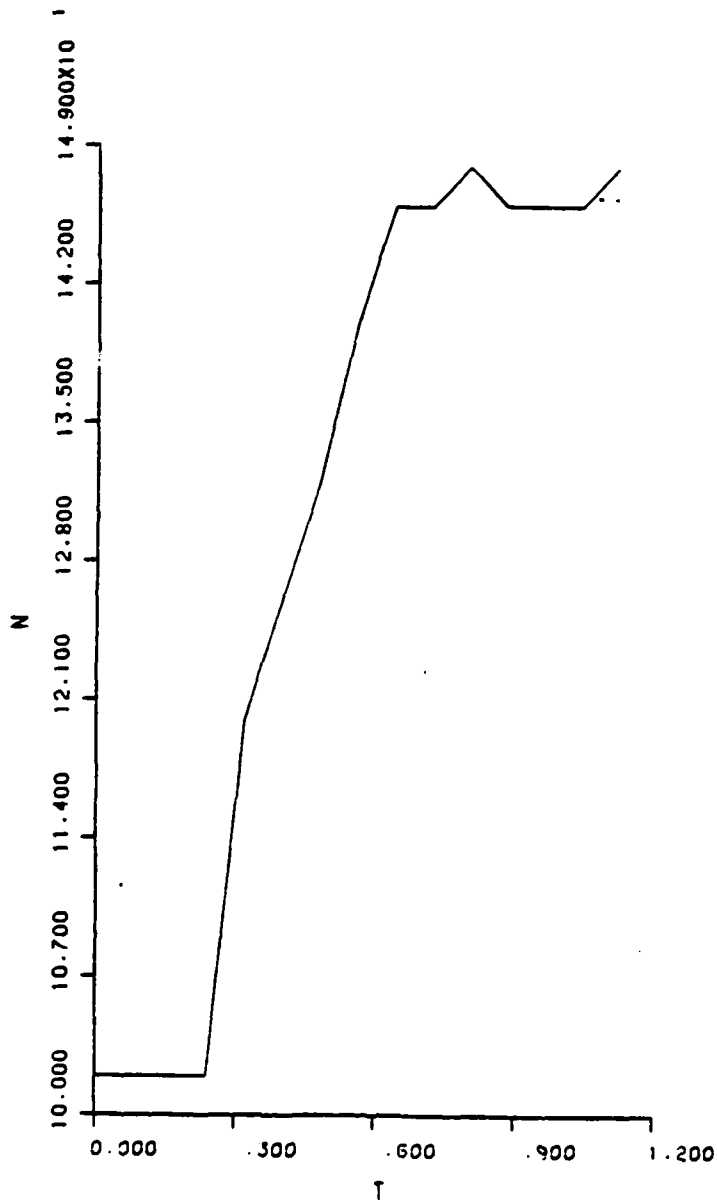
$$c_t = -vc_x - ((1 - \epsilon)/\epsilon)s_t$$

$$s = s_0 \frac{k_a c}{(1 + k_a c)}$$

$$c_t = \frac{-v}{1 + ((1 - \epsilon)/\epsilon)s_0 k_a / (1 + k_a c)} c_x$$



CHROMATOGRAPHY SYSTEM



NUMBER OF ODES

Solution of Stiff Equations Resulting from Partial Differential Equations

Bruce A. Finlayson
Department of Chemical Engineering
University of Washington
Seattle, Wash. 98195

Engineers often need to solve mathematical models that are expressed as partial differential equations. This paper describes several case studies from the author's experience, emphasizing the methods chosen to integrate the equations in the time domain, with special regard for the constraints prescribed by the fact that the time-dependent ordinary differential equations are derived from partial differential equations in space and time.

There are four themes illustrated in the case studies:

1. When a partial differential equation is discretized in the spatial domain by using a finite difference, finite element, or global collocation method, the resulting equations in time are stiff. The degree of stiffness can often be estimated a priori.
2. Integration schemes that work well for large error tolerances are valuable. First, the spatial truncation error may be large and it is wasteful to make the temporal truncation error very small. Secondly, the solution may have the same properties throughout the time interval considered; that is, the type of solution which causes the problem to be very stiff in the first place may move as a wave through space, so that as time proceeds the only thing that changes is the location of the stiffness-producing feature, not the stiffness itself. In that circumstance, methods which rely on taking large time steps during part of the time interval to achieve their success may be less useful.
3. Schemes and packages must be applicable to combined ordinary differential equations and algebraic equations, since these often occur in practice. The time derivative must be allowed to enter the problem as a matrix multiplied by the vector of time derivatives, since that form arises in most Galerkin finite element methods.
4. For large time-dependent problems in two and three spatial dimensions, extensive codes have already been developed using schemes that may not be the best for stiff problems. The techniques developed for stiff problems, though, must be done in a way that the schemes can be easily fit into those codes.

The themes are illustrated in case studies. The first study is for chemical reactors in the form of packed beds. While kinetic behavior has traditionally been stiff, the packed bed reactor can become more stiff due to sharp changes in temperature. Such reactors provide a useful and demanding testing ground for stiff integrators. The second case study is for flow through porous media. Here the problem is extremely stiff and good numerical solutions are difficult to obtain under certain conditions. The spatial variation of the solution plays a role both in defining the stiffness and in limiting certain advantages of stiff integration codes. The final example is the convective diffusion equation. This equation has similar difficulties to the equation for flow through porous media, but is linear. Even so, in two-dimensional time-dependent simulations the most sophisticated techniques are required.

Chemical Reactors

A prototype model of a packed bed reactor is given by the following equations.

$$\frac{\partial c}{\partial t} = \frac{\alpha}{r} \frac{\partial}{\partial r} \left(r \frac{\partial c}{\partial r} \right) + \beta R(c, T) \quad (1)$$

$$\frac{\partial T}{\partial t} = \frac{\alpha'}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) + \beta' R(c, T) \quad (2)$$

$$\frac{\partial c}{\partial r} = \frac{\partial T}{\partial r} = 0 \quad \text{at } r = 0 \quad (3)$$

$$c = c_0, \quad T = T_0 \quad \text{at } t = 0$$

$$\frac{\partial c}{\partial r} = 0, \quad -\frac{\partial T}{\partial r} = Bi_w (T - T_w) \quad \text{at } r = 1 \quad (4)$$

The physical meaning of the dimensionless coefficients is given elsewhere [1]. One method for solving these equations is to use the finite difference method to represent the spatial variation of the solution and then use some other method of solution in time, such as Runge-Kutta, trapezoid rule (giving the

Crank-Nicolson method), or a GEAR package. If we divide the distance $0 \leq r \leq 1$ by a uniformly spaced set of grid points, with spacing h , we can write

$$T_j(t) = T(r_j, t); \quad r_j = h(j-1) \quad (5)$$

The differential equation (2) then becomes (for an internal point not on the boundary):

$$\frac{dT_j}{dt} = \frac{\alpha'}{h^2} (T_{j+1} - 2T_j + T_{j-1}) + \frac{\alpha'}{r_j 2h} (T_{j+1} - T_{j-1}) + \beta' R(c_j, T_j) \quad (6)$$

This equation can then be written as

$$\frac{dT_j}{dt} = \alpha' \sum B_{ji} T_i + \beta' R(c_j, T_j) \quad (7)$$

where the matrix B_{ji} is derived from Eq. (6). The task is then to integrate the i,j set of ordinary differential equations, as an initial value problem.

Another method of solution is to represent the spatial variation of the solution with a polynomial defined over the entire domain, $0 \leq r \leq 1$. Collocation is applied at the Gaussian quadrature points, giving the orthogonal collocation method [1,2]. The result of applying this method to Eq. (2) has the same form as Eq. (7), except that the matrix B_{ji} is different. In particular it is dense (every entry is non-zero) instead of banded.

If a Galerkin finite element method is applied to Eq. (2) the result is slightly different.

$$\sum C_{ji} \frac{dT_j}{dt} = \alpha' \sum B_{ji} T_i + 2\beta' \int_0^1 N_j R(c_j, T_j) r dr \quad (8)$$

Notice here that the left-hand side involves a matrix multiplication. This means that explicit schemes for integrating Eq. (8) are ruled out, or must at least involve one LU decomposition of the matrix C_{ji} . Furthermore some packages for integrating ordinary differential equations do not allow equations in such a form, or require extra manipulation. For example, the code LSODI, developed by Hindmarsh at the Lawrence Livermore Laboratory, can be applied to equations in this form, whereas the GEAR or GEARB packages developed by Hindmarsh must be applied to the revised equation,

$$\frac{dT_i}{dt} = \sum (C^{-1})_{ij} \left\{ \alpha' \sum B_{jk} T_k + 2\beta' \dots \right\}$$

where the notation C^{-1} is meant to denote an LU decomposition rather than an inversion, to preserve the sparse nature of the matrices B_{ij} and C_{ij} .

All methods lead to sets of ordinary differential equations that must be integrated in time. The various packages RKF45, GEAR, GEARB, LSODE, LSODI can all be applied to some subset of the problems. The finite difference method and the finite element method lead to problems with banded matrices, whereas the orthogonal collocation method leads to problems with dense matrices. Despite this difference it turns out that all the methods lead to stiff equations if very many grid points are used. This point is illustrated with the diffusion equation.

For a diffusion problem

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} \quad (9)$$

we can apply finite difference, finite element or orthogonal collocation methods to obtain a discretized set of equations.

$$\sum C_{ji} \frac{dT_i}{dt} = \sum B_{ji} T_i \quad (10)$$

The difficulty of integration of these equations depends on the eigenvalues of the matrix, in particular:

$$\det |C_{ji} \lambda - B_{ji}| = 0 \quad (11)$$

For collocation and finite difference methods the matrix C_{ji} is the identity matrix. If we apply separation of variables to Eq. (9) we are led to the eigenvalue problem.

$$\frac{d^2 X}{dx^2} + \lambda X = 0 \quad (12)$$

If we apply collocation, finite difference, or finite element methods to this eigenvalue problem we get Eq. (11). Thus the first eigenvalue from Eq. (11) represents an approximation (and usually a quite good approximation) to the first eigenvalue of the physical problem, Eq. (12). The largest eigenvalue,

however, is found to be represented by the following equation,

$$\lambda_{max} = LB/h^2 \quad (13)$$

where the value of LB is listed in Table I for a variety of methods [1]. What this means is that the stiffness ratio for a linear diffusion problem is given by

$$SR = LB/\lambda, h^2 \quad (14)$$

and that as h becomes smaller, when we add more grid points, the problem becomes stiffer. These features are magnified for the nonlinear problem, such as Eq. (1,2).

Table I. Values of LB in Eq. (13)

| Method | LB |
|-----------------------|-----|
| Finite difference | 4 |
| Collocation, FEM | |
| cubic functions | 36 |
| quartic functions | 98 |
| quintic functions | 222 |
| Collocation, Hermite | |
| cubic functions | 36 |
| Galerkin, FEM | |
| linear | 12 |
| quadratic | 60 |
| Galerkin, FEM, lumped | |
| linear | 4 |
| quadratic | 24 |

The implication of this information is that if the problem demands a large number of finite difference grid points or finite elements to represent the spatial changes in the solution, then the temporal problems are guaranteed to be stiff, and integration methods must be used that are applicable to stiff equations.

The solution methods described above are embodied in two computer codes, REACOL for orthogonal collocation, and REACFD for finite difference. The temporal integration is done by one of three methods: a fixed-time-step second-order Runge-Kutta method, a variable time-step fourth-order Runge-Kutta method, and the GEARB package (Hindmarsh version).* The program is arranged so

*GEARB is used for orthogonal collocation only because it was available, even though the appropriate package is GEAR.

that the user can easily change the reaction rate expression and the number of components to fit the situation. The advantage to using such a code in an academic setting is that every student can obtain results for his or her reactor model even though the problem may be very difficult. It is very discouraging (and not too conducive to learning) to try to solve a difficult problem, get unstable numerical results, and not know where the difficulty lies - in the coding or the problem. Use of GEARB obviates that uncertainty; if GEARB has difficulty the student may still want to look at the coding for possible errors, but half the battle is won.

As a simple example take Eq. (1,2) with

$$R = c \exp(\gamma - \gamma(T))$$

and the parameters $\alpha = \alpha' = 1$, $\beta = 0.3$, $\beta' = 0.2$, $\gamma = 20$, $Bi_w = 20$, $T_w = 1$, $c_0 = T_0 = 1$. Solutions obtained with the orthogo-

nal collocation method are shown in Table II. Note particularly that if only two collocation points are used in the radial direction (a number which suffices for some problems) all the temporal methods take about the same computation time and have comparable errors. The error is almost entirely due to the spatial approximation, as can be seen by comparing the results for $N = 2$ and $N = 5$. For $N = 5$, however, the problem is stiffer, and the methods designed for stiff equations are faster, although not any more accurate. When the finite difference method is employed with a fixed time step and a second-order Runge-Kutta method, the solution varies with grid spacing and time step.

$$\langle c \rangle = a + b \Delta r^2 - c \Delta t^2 \quad (15)$$

There is no need to take t very small to have an accurate answer in time when there is still some truncation error in space. In fact because of the cancellation of errors in Eq. (15) this method may be more accurate than a solution derived using a variable time-step Runge-Kutta method or GEAR method.

Table II. Solutions to Reaction Problem
Giving values of average concentration at one $t = 0.6$.

| N in orthogonal collocation | c | Method in time | CPU time sec |
|-----------------------------|--------|--------------------------------|--------------|
| 2 | 0.8068 | RK-2, $t = 0.005$ | 0.97 |
| | 0.8031 | RKINIT, $\text{eps} = 10^{-5}$ | 0.97 |
| | 0.8088 | GEARB, $\text{eps} = 10^{-5}$ | 1.00 |
| 5 | 0.9197 | RK-2, $t = 0.001$ | 14.1 |

| | | | |
|-------|---------|--------------------|-----|
| | 0.9198 | RKINIT, eps = 10-5 | 7.9 |
| | 0.9206 | GEARB, eps = 10-5 | 3.1 |
| exact | 0.91926 | | |

A phthalic anhydride reactor provides a real case with interesting mathematical results (3,4). This time the packed bed reactor is operated under conditions in which the temperature of the packing need not be the same as the temperature of the fluid flowing past the packing. This temperature difference occurs because reaction occurs on the catalytic packing and energy is given off, raising its temperature. The energy must be transferred to the fluid, but encounters a heat transfer resistance between the packing and the fluid. The resulting equations are similar to Eq. (1-4), except that there are several species to consider.

$$\frac{\partial c_i}{\partial t} = \frac{\alpha}{r} \frac{\partial}{\partial r} \left(r \frac{\partial c_i}{\partial r} \right) + Da_I R_i(\{c_i\}, T_s) \quad (16)$$

$$\frac{\partial T}{\partial t} = \frac{\alpha'}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) + \gamma(T_s - T) \quad (17)$$

$$\gamma(T_s - T) = Da_{III} R(\{c_i\}, T_s) \quad (18)$$

The function R is a nonlinear function of concentration and packing temperature, T_s . Thus Eq. (18) represents an algebraic equation to be solved at each position r and t . When orthogonal collocation is applied to Eq. (16-17) one obtains a set of ordinary differential equations, as in Eq. (6), coupled with a set of algebraic equations. This type of problem can be solved using REACOL and GEARB provided the reaction rate expression solves the algebraic equation for T_s given the fluid temperature and all the concentrations. This was done using Newton-Raphson to solve the nonlinear equations. At the conference, results also will be presented using the program LSODI.

This reactor is parametrically sensitive. Figure 1 shows the solution when the inlet and wall cooling temperature are 620 K and 630 K. In the first case, the conversion of ortho-xylene to phthalic anhydride is well behaved and the temperature of the reactor remains within bounds. If the temperature is raised just 10 K, though, the temperature rises very sharply, and all the phthalic anhydride is further reacted to form carbon dioxide.

The GEARB package is capable of performing the integration up until the point where the reaction temperature rises precipitiously; there the integration stops because it cannot proceed with a very small required step size. This is a good example of the power of having a robust stiff integrator available. When a robust integrator is used and the reactor "blows up" we know that something dramatic is happening in the physics and chemistry, because otherwise the integrator would handle the problem easily. This same feature of reliability was especially valuable in a pioneering study of the kraft pulping process. [5]. In this case reaction and diffusion occurred in a wood chip, and the equations are similar to Eq. (1-4). The reaction rate expressions are unusual, however, and lead to some difficult integration problems. The REACOL program with orthogonal collocation and GEARB performed admirably and gave valuable results that otherwise might have been clouded by numerical problems and uncertainties.

Flow Through Porous Media

The next case study concerns the movement of water through very dry soil. The equation is

$$-\frac{dS}{dp_c} \frac{\partial p}{\partial t} = \frac{\partial}{\partial x} \left(k_r(p_c) \frac{\partial p}{\partial x} \right), \quad p_c = -p \quad (19)$$

with boundary conditions appropriate to injecting liquid water at one end by keeping the pressure or saturation at a fixed value.

$$p(0, t) = B P_1$$

$$\partial p / \partial x = 0 \quad \text{at } x = 1$$

The initial conditions are

$$p(x, 0) = B P_0$$

When the soil is very dry, the relative permeability, k_r , can vary over many orders of magnitude; factors of 10^6 are not uncommon. The derivative of a saturation with capillary pressure can also make large variations. For the cases considered here we take these functions as

$$k_r = 1 / (1 + p_c L / 146)^{6.65}, \quad S = 0.32 + 0.68 / (1 + p_c L / 231)^{3.65}$$

Typical solutions are shown in Figure 2 as a function of the initial dryness; as can be seen, if the soil is initially very dry the profile is very steep and the water front moves through the porous media as a wave.

It is clear that to interpolate such a function it is necessary to have a very fine mesh near the front. Unfortunately, the front moves in time, so that the fine mesh must either be everywhere or else it must somehow be moved to occur in the right location at the right time. A straightforward application of the finite difference method to Eq. (19) gives

$$-\frac{dS}{dP_c} \left| \frac{dp_i}{dt} \right| = \frac{1}{\Delta x^2} \left[k_{i+1/2} (P_{i+1} - P_i) - k_{i-1/2} (P_i - P_{i-1}) \right] \quad (20)$$

One method of evaluating the permeability is to take an average value.

$$k_{i-1/2} = \frac{1}{2} (k_i + k_{i-1})$$

These equations can be integrated using a package for stiff equations, such as GEARB. At any time one can stop the integration and determine the eigenvalues of the Jacobian of Eq. (20). Typical values are listed in Table III. Notice that some of the stiffness ratios are quite large, 10^9 .

Table III. Eigenvalues of Jacobian, L = 100

| BPO | BP1 | max EV | min EV | SR | CR |
|------|-----|--------|--------|-------|-------|
| 0 | -2 | 1.2E6 | 0.25 | 4.9E6 | 1.4E5 |
| -0.5 | -2 | 1.3E3 | 0.22 | 5.6E3 | 1.4E2 |
| 0.05 | -3 | 1.5E8 | 0.14 | 1.1E9 | 1.3E8 |
| -0.5 | -3 | 1.2E3 | 0.032 | 3.9E4 | 1.2E3 |
| 0 | -5 | 7.5E3 | 0.0053 | 1.4E7 | 3.0E5 |

We can also define a coefficient ratio as follows.

$$CR = \frac{\max_i \left| \frac{1}{k_r} \frac{dS}{dP_c} \right|_i}{\min_i \left| \frac{1}{k_r} \frac{dS}{dP_c} \right|_i}$$

This ratio can be calculated a priori since it depends only on the pressure, and the range of pressure values is known at the

outset. Fortunately, it correlates very well with the calculated stiffness ratio, as indicated in Figure 3. This coefficient ratio then provides a guide for other problems to help identify which problems are going to be especially stiff. It is also useful for choosing a useable time-step for a fixed time-step method. Jensen and Finlayson 6 show how optimal discretization can reduce the stiffness ratio from 10^{11} to 10^6 for one of these problems.

One of the difficulties with this problem is that the pressure can be positive if the soil is saturated. Under those conditions the coefficient, $dS/dp = 0$. If the soil is initially dry, and then one end is put in contact with water (say a layer of water is placed on top of it) then that end will be saturated. The saturated region will move in time. The region of space which is saturated then is governed by an equation like (19) but with no time derivative, i.e. an algebraic constraint. Thus to solve this problem it is necessary to have a stiff integration package which can handle these algebraic constraints, and the number of algebraic constraints depends on the solution and may change in time. A modification of the original Gear program [7] was made in order to do this.

Supposing the time dependent nature of the solution is properly handled, what type of behavior usually results? One solution is shown in Figure 4; it was generated using the Galerkin method with some upstream dispersion and a trapezoid rule in time with a fixed time-step. The solution is clearly in error. Unfortunately, the errors are not caused by the time step used; solving the problem more accurately in time leads to no better solution. The difficulty lies with the spatial approximation. In this case only 20 elements were used and it is clear that the type of solution expected cannot be interpolated with 20 elements. For another problem discussed below it is possible to show how many elements are needed for a given solution. Here we cannot do that, but other calculations indicate something like 200 elements are needed if a cubic trial function is used, giving 601 total points. Sometimes it is not possible to compute with this many points for economic reasons. In that case one might try to use only 20 or 40 elements. If one does so with a stiff integration package, the oscillations caused by the poor spatial approximation will be carefully followed by the integration package, and large time steps will never be achieved. In such cases, then, it does not make sense to use a small error tolerance, since the spatial contribution to the error is large anyway, except that the integration packages may not work except with small values. The author has found that values of ϵ below 10^{-3} are generally necessary for the GEARB package developed by Hindmarsh to work efficiently for these problems.

If one is dissatisfied with a solution containing oscillations, since the oscillations are known to be due to numerical errors, and the calculations must be performed on a fixed budget,

it is necessary to resort to introducing numerical dispersion. In this problem the simplest way to do so is to evaluate the permeability by using the value of permeability at the nearest node upstream (with upstream defined as the direction with the largest pressure).

$$k_{i-1/2} = k_{i-1}$$

There are other more sophisticated and more accurate methods as well. If one does this then the oscillations are damped; the front is smooth but may be much smoother than the true solution. When this approach is taken it clearly does not make sense to solve the equations in time very accurately; the spatial contribution to the error is large. Again stiff integration packages that work well with large error tolerances are important.

Figure 5 shows the error in the pressure at a particular position and time (chosen so that the sharp front is nearby and any errors in the position of the front cause a large error). The finite difference method is used in space and two methods are used in time. One uses a backward Euler method with a fixed time step. As the time step is decreased the error decreases, but approaches an asymptote since eventually all the error is contained in the spatial approximation. A modified version of Gear is also used (modified to allow algebraic equations) and the problem is solved with error tolerances of $\epsilon = 1.0$ and 0.1 . With these values the solution is solved accurately in the sense that there is very little time truncation error, but the economic cost is high since the computation time is more than that obtained with a simple fixed time-step method. Part of the reason the stiff integration package did no better than this is that the problem is stiff because the pressure changes from the boundary value to the initial value, and this leads to large variations in the coefficients. Yet the solution always makes this variation, and is always this stiff, and large time steps are not utilized in any portion of the time integration. Furthermore, small oscillations that appear initially (and are inevitable when there is a step change in the boundary condition at time zero) are carefully tracked by the stiff integration package. This example is not meant to be presented as a failure of the stiff integration package - indeed the package made possible many solutions that were otherwise inaccessible - but to indicate that the dramatic results achieved in other contexts do not always obtain for solving the ordinary differential equations derived from partial differential equations.

If one wants to solve this problem really well it is necessary to either use a very fine mesh or a moving mesh. This was done by one of the author's students [6]. The problem was transformed to a moving coordinate system that moved with the velocity of the front. Then the front stayed in the same position for all

time, and the location of the boundaries moved in time. The solution required defining a set of nodes and solving the problem on a subset of nodes, and the subset kept changing in time. Rather than trying to force such a problem into the structure of the existing stiff integration packages (such as GEARB) it was easier to use the trapezoid rule, with a fixed time-step. Since the solution hardly changed in time, large time steps could be used; indeed the major time-saving resulting from the technique was the large time steps possible; in the moving coordinate system the solution appeared as if it was in steady state. While a stiff integration package was not used in the solution, the experience of using them on similar problems and understanding why they did not always work well was an essential preliminary to devising such a strategy.

Chemical Flooding

One of the methods for producing oil from existing oil fields is to inject a solution of surfactant chemicals in water in order to reduce the interfacial tension between the water and oil phases. A prototype problem related to this problem is the solution of the convective diffusion equation.

$$\frac{\partial c}{\partial t} + Pe \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}$$

$$c(x, 0) = 0$$

$$c(0, t) = 1$$

$$c(1, t) = 0$$

A numerical solution is shown in figure 6. This solution was obtained using the finite difference equation in the form

$$\frac{dc_i}{dt} + Pe \frac{c_{i+1} - c_{i-1}}{2\Delta x} = \frac{1}{\Delta x^2} (c_{i+1} - 2c_i + c_{i-1}) \quad (21)$$

The ordinary differential equations were then solved using GEARB. The error criterion was set low enough that the solutions are essentially the exact solutions of the difference equations (21). If only 50 grid points are used, oscillations develop, as shown. These oscillations are inherent in the spatial approximation - no better solution of the ordinary differential equations is possible. The GEARB routine faithfully follows the development and propagation of each oscillation. If 500 grid points are

used, then the numerical solution is essentially exact, as shown. Here, too, the time dependent equations are solved essentially exactly.

Now consider the problem when the parameter Pe , the Peclet number, increases by 1000 times. Then we need 500,000 points for a good solution, and such a solution is very expensive to obtain. Yet if fewer points are used, oscillations develop due to the spatial approximation, and very accurate time-dependent methods are not helpful. In this case we can introduce upstream dispersion by changing the convection term to the following form.

$$\left. \frac{\partial c}{\partial x} \right|_i = \frac{c_i - c_{i-1}}{\Delta x}$$

Figure 6 shows the solution using this form when only 50 grid points are used. The solution is smooth, the solution to the time-dependent equations is found very accurately by the GEARB routine, but it is clear that the error obtained when comparing the numerical solution to the exact solution is large. Again it makes little sense to obtain very accurate answers to a problem when upstream dispersion has been introduced; a stiff integrator is needed that works well for large values of the error control parameter.

This problem, too, can be solved by a moving coordinate system, and this has been done by Jensen and Finlayson [6]. Very small meshes are used near the front, and the front is stationary in the moving coordinate system, but the location of the boundaries changes in time. In order to achieve economical results it is necessary to use a method of integration which is at least A-stable, and the trapezoid rule was used. Large time steps were then possible. Because some of the nodes are in the solution domain and others are not at any particular time the stiff integration packages were not used.

For the chemical flooding application it is necessary to solve equations like the convective diffusion equation in a two dimensional region.

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} = D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) \quad (22)$$

Typically fluid is injected in one corner of a square and removed in the corner diagonally opposite. The Peclet number is very large, requiring a small grid spacing to prevent oscillations. The small grid spacing makes the calculations very expensive, and numerical dispersion cannot be used because it changes the peak concentration of the surfactant in an unrealistic way, which changes the phenomenon. One solution to this problem is to solve the equations on a moving coordinate system, this time one that

moves from the injection well to the production well. Small finite elements are placed near the front, and the front always stays in the region where the elements are small. To efficiently handle the time-dependent problem, though, it is necessary to use an integration method that is A-stable, so that large time steps can be taken, but also one that is efficient in solving the linear algebra problem which must be solved each time step. Solving these equations with a banded solver is much less efficient than solving with a profile or frontal routine. Thus it was necessary to develop a method which was compatible with a frontal routine, was A-stable, and had an automatic step control feature. In essence, what was needed was a low-order, high-eps version of GEARB.

Since a graded mesh was involved it was most convenient to use the Galerkin finite element method since elements of different sizes are easily handled. This means that the differential equation (22) is in an implicit form.

$$M_{ij} \frac{dc_j}{dt} + A_{ij} c_j = D B_{ij} c_j$$

Gresho, Lee and Sani [8] of the Lawrence Livermore Laboratory have provided an integration routine that has the features of stiff integration packages. It uses a second-order Adams-Bashforth method for a predictor, a trapezoid rule for the corrector, and the difference between the predicted and corrected value to control the step size to meet the specified error tolerance. Gresho, et al. provided both a first and second order method, as we do here. The method used by Gresho, et al., though, is based on time derivatives of the variables. Since the equation is implicit, it is necessary to solve this implicit equation first even to get started. Rather than doing that we devised a method based upon extrapolation. The derivation is given here for the first order method, and the results for the second-order method are in the Appendix.

First we write an equation for an extrapolation of the polynomial going through the solution at t_{n-1} and t_n (see Figure 7). The method is illustrated for $dy/dt = f(y)$.

$$y_{n+1}^p = y_n + \frac{h_n}{h_{n-1}} (y_n - y_{n-1}) \quad (23)$$

We next write a Taylor series for the exact solution, assuming the exact solution is known at the beginning of a time step, $y(t_n) = y_n$ and $y(t_{n-1}) = y_{n-1}$.

$$y(t_{n+1}) = y(t_n) + h_n y'(t_n) + h_n^2 y''(t_n)/2 + \dots$$

Subtracting this from Eq. (23) gives

$$y_{n+1}^p - y(t_{n+1}) = \frac{h_n}{h_{n-1}} (y_n - y_{n-1}) - h_n y'(t_n) - h_n^2 y''(t_n) / 2$$

Using a Taylor series for y_{n-1}

$$y_{n-1} = y_n - h_{n-1} y'_n + h_{n-1}^2 y''_n / 2$$

gives

$$y_{n+1}^p - y(t_{n+1}) = -h_n (h_n + h_{n-1}) y''_n / 2 \quad (24)$$

The corrector is given by the backward Euler method.

$$y_{n+1}^c = y_n + h_n f(y_{n+1}^c) \quad (25)$$

The truncation error of the backward Euler method is

$$y_{n+1}^c - y(t_{n+1}) = h_n^2 y''_n / 2 \quad (26)$$

The values of y_{n+1}^p and y_{n+1}^c are available from Eq.

(23,25). We solve Eq. (24,26) for the unknown $y(t_{n+1})$ and y''_n . The result is

$$y_{n+1}^p - y_{n+1}^c = -[2h_n^2 + h_n h_{n-1}] / 2$$

Thus the truncation error is then estimated using

$$d_{n+1} = y_{n+1}^c - y(t_{n+1}) = \frac{y_{n+1}^c - y_{n+1}^p}{2 + h_{n-1}/h_n} \quad (27)$$

To summarize we use Eq. (23) to predict y_{n+1}^p , Eq. (25) to correct, and Eq. (27) to estimate the truncation error.

The estimate of the truncation error is used to pick the next time step.

$$h_{n+1} = h_n (\epsilon / |d_{n+1}|)$$

where ϵ is the user-specified accuracy. In Jensen's thesis [9] this time step was used for the next calculation. Jensen also used the second order version for the convective diffusion equation.

In practice, of course, the estimate of truncation error may not be a good estimate, and after picking a time step and doing the calculations the estimate of the truncation error in the new step may be larger than ϵ . Consequently, Jensen [10] devised the following strategy to allow recalculation when necessary. Define the error for sets of equations as

$$d = \left(\frac{1}{N} \sum_{i=1}^N d_{n+1,i}^2 \right)^{1/2}$$

or

$$d = \max_i |d_{n+1,i}|$$

If

$$\beta \epsilon < d$$

the timestep is not accepted since the truncation error estimate actually achieved is greater than expected from the previous steps. The step is repeated with a step size half as big. The value of beta is empirically chosen in the range 1 to 1.5. If

$$d \leq \beta \epsilon$$

then the step is accepted. However, if

$$\epsilon < d \leq \beta \epsilon$$

the next timestep is decreased according to

$$h_{n+1} = h_n (\epsilon / d)^{1/2}$$

This allows the estimated error to be slightly larger than desired, as large as beta x epsilon. If

the next timestep is kept the same, and alpha is a parameter empirically chosen in the range 1 to 10. Finally if

$$d \leq \frac{\epsilon}{\alpha}$$

the time step is increased according to

$$h_{n+1} = h_n \left(\frac{\epsilon}{\alpha d} \right)^{1/2}$$

with a maximum ratio of 3 to prevent introducing excessive error. The corresponding formulas based on interpolation using second order expressions are given in the Appendix.

The time-step scheme described above is based on the same ideas contained in the GEARB package developed by Hindmarsh, but has the advantage that it is easily implemented in a code that already uses the backward Euler method. The second-order scheme in the Appendix can likewise be used in a code that now uses the trapezoid rule, or Crank-Nicolson. The second order scheme was applied by Jensen in his thesis to solve for the movement of chemical in the two-dimensional flow pattern described above. He always accepted the predicted step size, and always accepted the solution, even if the truncation error turned out to be too large. When the solution reached a point in which smaller time steps had to be taken it sometimes broke down. Later he devised the strategy given above for accepting a stepsize, and applied this to a large code for steam injection of an oil field, where it worked with good results [10]. Approximately 30 to 50% of the computation time was saved compared to the previous step-size control strategy, and cases converged that had not converged with the previous strategy. The previous strategy was one in which the change in saturation or pressure from one time step to another was kept below a prescribed value. Obviously this limits the truncation error only in some vague, general sense. The schemes discussed here, however, have a theoretical justification and work even better.

Conclusion

The concepts of stiff integration codes can sometimes be applied directly to solve the ordinary differential equations arising from partial differential equations. In some cases, however, the problems are so difficult or large that a straightforward application of the stiff integrators would be prohibitive in cost. In such cases the principles of the stiff integrators can sometimes be incorporated with other solution techniques and tricks, leading to a viable solution method. These principles have been illustrated in three case studies.

References

1. Finlayson, B. A., *Nonlinear Analysis in Chemical Engineering*, McGraw-Hill (1980).
2. Finlayson, B. A., *The Method of Weighted Residuals and Variational Principles*, Academic Press (1972).
3. Finlayson, B. A., *Methods for Solving Partial Differential Equations in Chemical Engineering*, paper presented at joint AIChE-CIESC meeting, Beijing, China, Sept. 19-22, 1982.
4. Froment, G. F., *Ind. Eng. Chem.* vol. 59(2), 18 (1967).
5. Gustafson, R. R., Sleicher, C. A., McKean, W. T., and Finlayson, B. A., "A Theoretical Model of the Kraft Pulping Process," submitted to *Ind. Eng. Chem. Proc. Des. Dev.*
6. Jensen, O. K. and Finlayson, B. A., *Adv. Water Resources*, vol. 3, 9 (1980).
7. Finlayson, B. A. and Nelson, R. W., "A Preliminary Investigation into the Theory and Techniques of Modeling the Natural Moisture Movement in Unsaturated Sediments," Report BCSR-40, Boeing Computer Services, Richland, Sept. 1977.
8. Gresho, P. M., Lee, R. L. and Sani, R. L., Chapter 2 in *Recent Advances in Numerical Methods in Fluids*, Vol. 1, C. Taylor and K. Morgan, ed., Pineridge Press, Ltd., Swansea, U. K., (1980).
9. Jensen, O. K., "Numerical Modeling with a Moving Coordinate System: Application to Flow Through Porous Media," Ph. D. Thesis, University of Washington (1980).
10. Jensen, O. K., "An Automatic Timestep Selection Scheme for Reservoir Simulation," *Soc. Petroleum Eng. paper 9373*, Dallas, Tex, Sept. 21-24, 1980.

Appendix

The formulas analogous to Eq. (23-27), but for a second-order trapezoid rule are:

$$y_{n+1}^P = \alpha_1 y_n + \alpha_2 y_{n-1} + \alpha_3 y_{n-2}$$

$$\alpha_1 = 1 + \frac{h_n}{h_{n-1}} \left[1 + \frac{h_{n-1} + h_n}{h_{n-1} + h_{n-2}} \right]$$

$$\alpha_2 = - \left[\frac{h_n}{h_{n-1}} + \frac{h_n}{h_{n-1} h_{n-2}} (h_n + h_{n-1}) \right]$$

$$\alpha_3 = \frac{h_n (h_n + h_{n-1})}{h_{n-2} (h_{n-2} + h_{n-1})}$$

$$y_{n+1}^c = y_n + \frac{h_n}{2} [f(y_n) + f(y_{n+1}^c)]$$

$$C_1 = 1/12$$

$$C_2 = \left[1 + (2h_{n-1} + h_{n-2})/h_n + (h_{n-1}^2 + h_{n-1} h_{n-2})/h_n^2 \right]$$

$$d_{n+1} = \frac{C_1}{C_1 + C_2} (y_{n+1}^c - y_{n+1}^P)$$

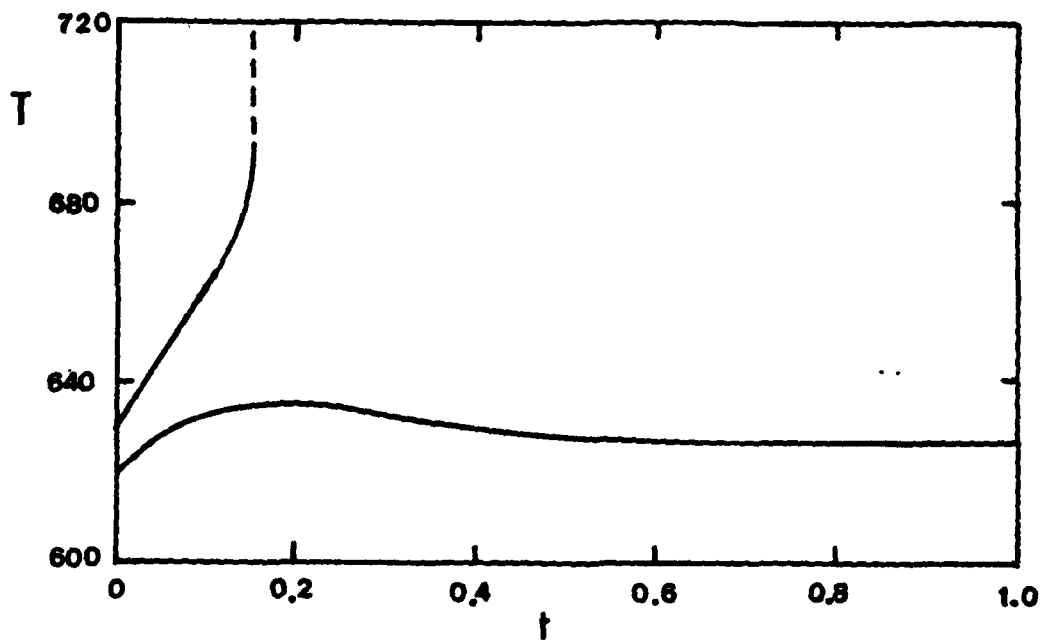


Figure 1. Average temperature in phthalic anhydride reactor [3]

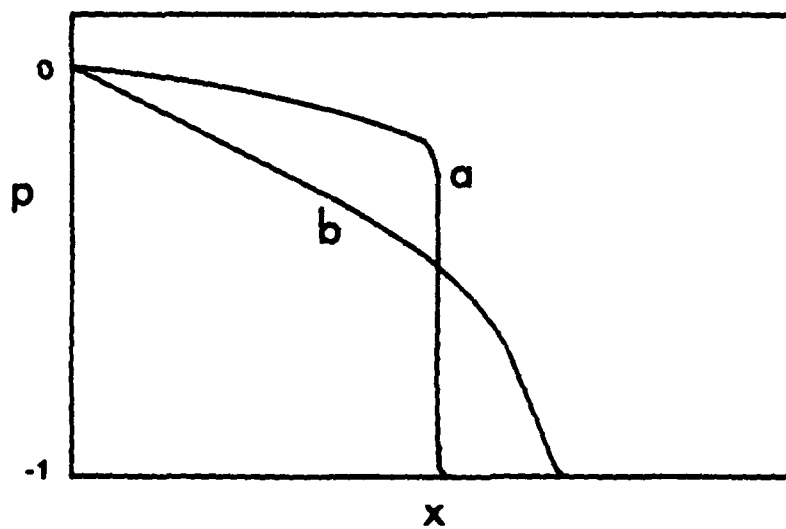


Figure 2. Pressure profiles for flow through porous media.

a, $L=1000$ cm; b, $L=300$ cm.

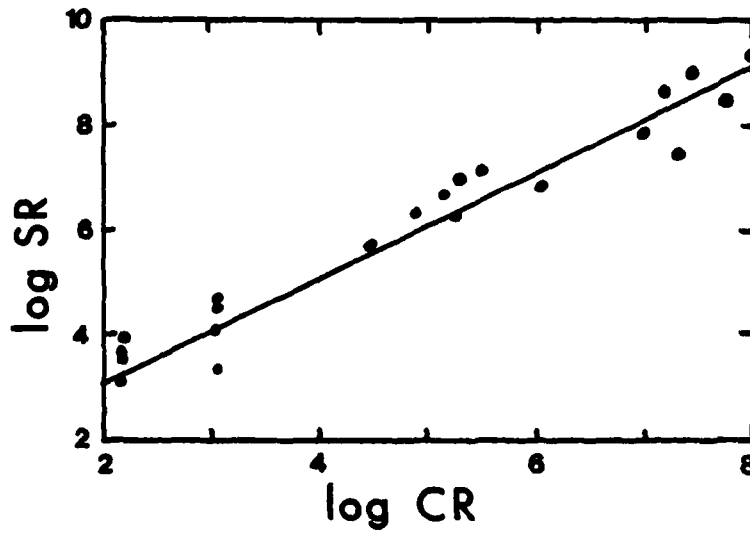


Figure 3. Coefficient ratio for several solutions to Eq. (20).

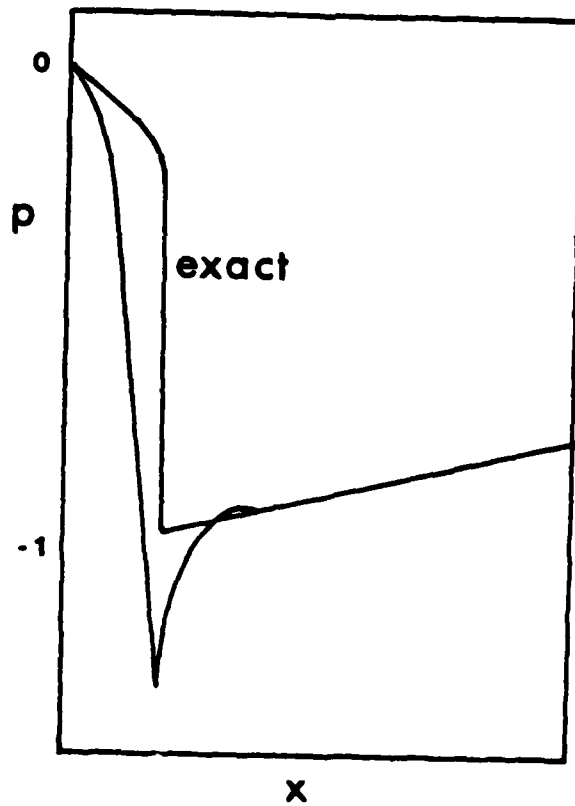


Figure 4. Pressure profile by Galerkin method.

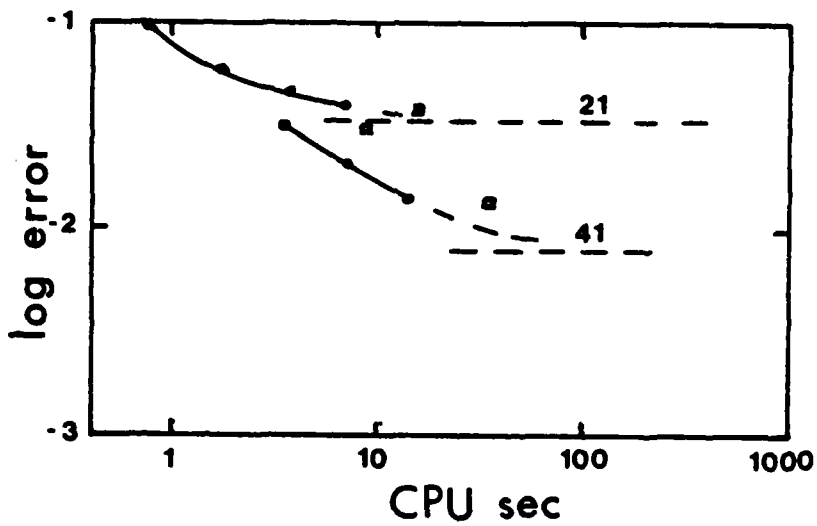


Figure 5. Error in pressure solution. Finite Difference.
Gear \square

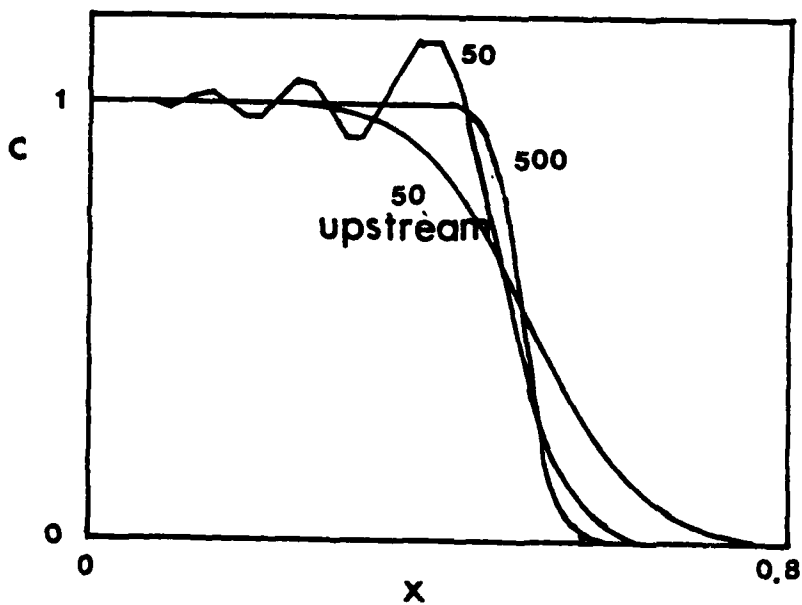


Figure 6. Solution of Convective Diffusion Equation.
Finite Difference, centered, $n = 50,500$.
Finite Difference, upstream, $n = 50$.

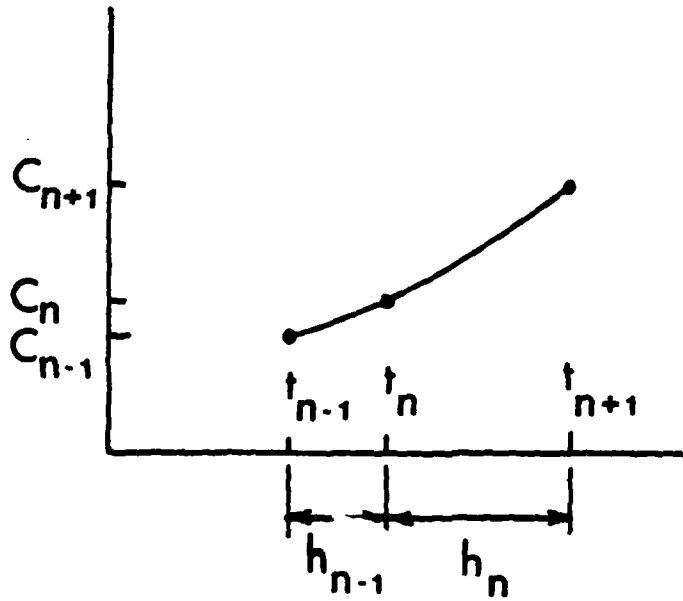


Figure 7. Diagram for timestep control.

STIFFNESS IN HEAT TRANSFER

by

Stuart W. Churchill
The Carl V. S. Patterson Professor of
Chemical Engineering
The University of Pennsylvania
Philadelphia, Pennsylvania 19104, USA

Introduction

Professor Aiken asked me to present an overview of stiffness as encountered in the field of heat transfer. My primary objective will be to draw your attention to current problems of physical reality and importance in that field. I have chosen to interpret his invitation to include energy changes from one form to another, thus encompassing shock, deflagration, and detonation waves. Before considering stiffness I will digress to review the relationships between heat transfer, mass transfer, momentum transfer and chemical kinetics.

Analogies and Couplings of Heat Transfer with Mass Transfer,

Momentum Transfer and Chemical Reactions

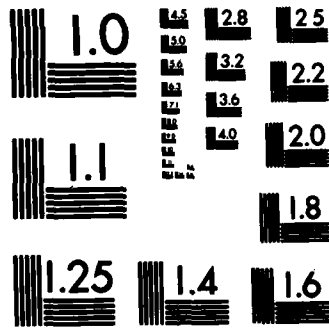
Mass Transfer.

The transfer of chemical species bears a close analogy to heat transfer, although usually further complicated by multicomponents and sometimes by the self-generation of a net flow normal to the surface, the so-called Ackermann effect [see, for example, Gröber, Erk and Brüll (1961) p. 482E, and Bird, Stewart and Lightfoot (1960) p. 656f].

Also, the geometries of greatest interest in mass transfer are not in general the same as in heat transfer. For example, convective heat transfer usually occurs from one confined fluid stream to another through a thin solid wall, whereas mass transfer characteristically occurs from one fluid to another in direct contact or to a solid surface. Mass transfer often occurs simultaneously with and coupled to heat transfer whereas heat transfer generally occurs in the absence of mass transfer. Heat transfer with phase changes (boiling, condensation, melting and solidification) is more important than the analogous processes of mass transfer. On the other hand the transfer of species due to an electric potential does not have an important analog in heat transfer. Heat transfer by radiation has a superficial analogy with the transport of thermal neutrons [see, for example, Chu and Churchill (1956)] as well as with other electromagnetic radiative processes [see, for example, Kerker (1963)].

Momentum Transfer

The current of momentum (ordinarily equal to the shear stress) is a tensor of second order as compared with the currents of energy and species which are vectors (tensors of first order). Hence an analogy between momentum transfer and heat (or mass) transfer is somewhat artificial, and can only occur in degenerate cases. Even so, this concept, as first conceived by Reynolds and improved by Prandtl and others [see, for example, Gröber et al. (1961) p. 242f, 395f], has proven to be invaluable for turbulent transport. The predictions based on this analogy, despite its shaky foundation, are probably more reliable than the available experimental



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

data [see, for example, Churchill (1977)]. Since it is uncertain whether or not a priori solutions of turbulent transport will ever be achieved, this useful analogy should not be scorned for its obvious empiricism.

Momentum and convective heat transfer usually have at least a one-way coupling, i.e., the rate of heat transfer depends critically upon the velocity field. Conversely, momentum transfer depends on heat transfer only insofar as the viscosity and density vary with temperature. In free (unconfined) and natural (confined) convection owing to a density variation and a gravity field, the equations for heat and momentum transfer are strongly intercoupled and must be solved interactively. Some free or natural convection always accompanies forced convection, its relative importance depending on the ratio of the maximum density difference to the square of the imposed velocity [see, for example, Churchill (1982a, 1982b)].

Momentum and Mass Transfer

To return to mass transfer, gravitationally induced changes may also occur, leading to analogous couplings between momentum and mass transfer, and to triple couplings. The latter is an important but relatively unstudied problem [see, for example, Seville and Churchill (1970)]. Coupled mass and momentum transfer may also occur due to the interfacial tension between two immiscible liquids. This is called the Marangoni effect [see, for example, Sternling and Scriven (1959)].

Chemical Reactions

Chemical conversions in tubular reactors are invariably coupled with momentum transfer since the assumption of plug flow is rarely justifiable.

For example, in laminar flow through an empty round tube the velocity distribution is parabolic, and in a tube filled with packing is M-shaped. Turbulent flow does not occur in practical reactors, with singular exceptions such as for the radiantly stabilized burner discussed subsequently. Non-equimolar reactions in turn influence the velocity distribution even for isothermal conditions.

For homogeneous reactions a velocity distribution implies a radial concentration gradient and hence radial mass transfer. If the reaction or reactions are significantly energetic a radial temperature gradient is also necessarily generated. The mass, energy and momentum equations are thus intercoupled. As evidence of the significance of this coupling, Brian (1963) has shown that the coefficient for heat transfer to the wall of a non-adiabatic reactor may be changed radically (as much as a factor of 10) by chemically generated or absorbed energy.

In packed beds of catalyst particles, heat and mass interchange with the surface of the pellets, adsorption, desorption, surface migration, and surface reaction may all count, and in the case of porous particles, diffusion of heat and mass through the pores as well. The thermal conductivity of the particle may be sufficient to justify the use of an isothermal effectiveness factor, but temperature and composition differences between the surface and the bulk of the adjacent fluid will generally be significant [see, for example, Fair (1955)]. It may be possible to account for some of these effects with adequate approximation by a lumped model, but not those of the radial velocity, composition and temperature profiles.

By contrast, for wall-catalyzed reactions the radial profiles of velocity, composition and temperature can be accounted for by lumping, just as for individual catalyst particles.

It is concluded that tubes with a surface coating of catalyst can be modelled by ordinary differential equations, but that homogeneous tubular reactors and packed catalytic reactors cannot be.

Finally, tubular reactors for endothermic processes are, as shown by Calderbank (1954) and others, are essentially heat exchangers. The reaction mechanism determines the temperature level, but the local rate of reaction is in close proportion to the local rate of heat transfer.

Having established the importance of heat transfer in chemical reactors I will defer to others for detailed discussion of this topic.

Behavior Generating Stiffness

Now let me return from this digression on the interrelationships between heat transfer and other processes, and consider stiffness. My posture here will be as a customer rather than as a supplier of techniques. That is, the following will be limited to a description of thermal problems which pose computational difficulties and therefore are worthy of your expert attention. I will also try to identify the physical and mathematical characteristics which lead to these difficulties. Since I am problem-oriented rather than technique-oriented, I ask your indulgence if some of these problems fall outside the narrow definition of stiffness as models composed of ordinary differential equations giving rise to widely separated eigenvalues. I trust that you of this highly selective audience, and particularly those of you who succeed

me at this podium, will provide guidance to the solution of this broader class of problems.

The more important and more difficult problems in heat transfer are necessarily modelled by partial differential equations, representing transient and/or multidimensional behavior. Presumably most of these problems involving partial differential equations can be transformed to ordinary differential equations by standard techniques such as operational methods, local similarity, local non-similarity, the method of characteristics, the method of lines and the method of weighted residuals. Hence I will include such problems in my discussion.

The following are then a representative set of heat transfer problems of varying complexity which may invoke stiffness or related computational difficulties. Attention will first be directed to those problems which can be modelled by ordinary differential equations, then to some that have both one- and two-dimensional aspects, and finally to a few that can only be considered as multidimensional.

Heat Exchangers

The interchange of energy between two or more confined streams can be represented by the corresponding number of ordinary differential equations for most geometrical configurations insofar as the effect of variable transport properties can be lumped. An example of an exception is rectangular cross-flow, which requires partial-differential modelling.

Simple analytical solutions have been obtained for concurrent and countercurrent flow with constant heat capacities and constant heat transfer coefficients. Other geometries, developing flow, developing heat

transfer and variations in the physical properties with temperature may give rise to combinations of differential equations which are difficult to integrate numerically. Most practical problems of geometrical complexity only have already been solved [see, for example, Jakob (1957)], but only a few of the simpler cases of developing flow and developing heat transfer have been considered. The effects of variable physical properties on heat exchange are still not well resolved because of the lack of general models for the temperature-dependence, and because of the strong influence of variable viscosity and density on momentum transfer.

Laminar Thermal Boundary Layers

Heat transfer in laminar boundary layers, although a very old and worked-over problem, still offers some challenges, as illustrated by the following two examples.

Low Prandtl Numbers. The boundary layer equations for free convection are numerically intractable for low Prandtl numbers, apparently because of instability. Although asymptotic solutions are readily derived for a Prandtl number of zero, numerical results have not been obtained for any boundary condition or geometry for finite values less than 0.001. Such low values of the Prandtl number only occur in galactic applications, but solutions would be very useful in developing correlations for terrestrial applications in the generalized form proposed by Churchill and Usagi (1972).

Recovery Factor. The coupled differential equations for the conservation of momentum and energy in dissipative flow over a flat plate have apparently never been solved directly despite the many practical applications, including thermometry. Numerical solutions have been obtained from an integral formulation for some particular Prandtl numbers [see, for example, Gröber, et al. (1961) p. 287f].

Shock Waves

Supersonic compression waves can be generated by the motion of a strong pressure pulse or the rapid motion of a piston. Their normal reflection off a solid surface results in an even stronger wave. After full development, shock waves and reflected shock waves can be modelled as stationary, insofar as their decay due to dissipative processes can be neglected, by choosing the wave front as a frame of reference. Insofar as diffusion and dissipation are negligible, a shock wave can be modelled by a set of non-linear algebraic equations as a step function in pressure, velocity and temperature. The effects of diffusion and viscous dissipation on a stationary wave can be modelled by a set of coupled ordinary differential equations representing the conservation of mass, momentum and energy. Presumably this model is very stiff. However, Bird, et al. (1960) pp. 333-336, present an analytical solution which is exact for a Prandtl number of $3/4$. They also discuss solutions for other conditions.

Shock waves may be generated in a tube, adding the complication of the drag of the wall. Spherical and cylindrical waves are usually postulated to be pseudo stationary.

Detonation Waves

Supersonic compression waves can also be generated by a rapid chemical reaction. If diffusion and dissipation are neglected and chemical equilibrium is assumed behind the wave the fully developed behavior can be modelled algebraically as a step function. This idealized model provides excellent predictions for the detonation velocity [see, for example, Moyle, Churchill and Morrison (1960)]. Even so, computations of the effect of finite reaction rates and diffusion are of interest. For example, Nicholls (1960) established

a stationary detonation wave for a mixture of hydrogen and oxygen experimentally, and thereby determined the reaction rate.

Experimental measurements of the temperature and pressure reveal transients which are presumed to be due to the finite rates of chemical and thermal relaxation. Computation of these effects is a challenging problem involving stiffness which must necessarily be modelled by partial differential equations.

The development of a detonation wave following ignition is another unresolved and challenging problem involving stiffness and partial differential equations.

Deflagration Waves

Unconfined laminar flames produce truly stationary expansion waves, but the postulates of chemical equilibrium and negligible diffusion are not applicable even as first-order approximations. On the other hand the general model for this process is only moderately stiff. Fristrom (1960) has shown that a Bunsen flame can be modelled satisfactorily in terms of diffusion and reaction. Hahn and Wendt (1982) have developed a numerical solution for an opposed flat flame, taking into account the unique flow field, diffusion and free-radical reactions.

The transient problems of ignition and flame development provide a further challenge. In addition radial symmetry is of more direct interest than planar.

Radiantly Stabilized Combustion

The stationary combustion of premixed air and fuel (either gas or an atomized volatile liquid) in a refractory tube has been successfully modelled

by an ordinary integro-differential equation and a large set of coupled ordinary differential equations. This complex model is characterized by split boundary conditions as well as stiffness; the process of solution by instability; and the solutions by multiplicity.

The physical behavior can be described as follows. In passage down the inlet region of the tube the fuel-air mixture is heated to the point of ignition by convection from the tube wall. Combustion then occurs with a rapid rise in temperature. Downstream from this combustion front the burned gas heats the colder tube wall by convection. The resulting hot downstream wall radiates and conducts energy to the colder upstream wall thereby providing the thermal feedback required to preheat the fuel-air mixture. For a sufficiently large tube diameter, the stable range of operation falls in the turbulent regime. Hence the process can be modelled with reasonable accuracy as plug flow. The model consists of separate energy balances for the gas, liquid (if any) and solid (the wall), and separate material balances for each of the chemical species in the gas phase, including free radicals. The energy balance for the wall is an ordinary integro-differential equation. The other balances are ordinary differential equations. The energy balances for the gas and wall are coupled through a convective boundary condition, and the energy and material balances for the gas by the temperature- and composition-dependence of the reaction mechanisms. In the case of liquid droplets, the mass balances as well as the energy balances for the gas and liquid streams are also coupled through convective boundary conditions. Secondary couplings occur due to physical property dependence on temperature and composition, and the dependence of the convective coefficients on the reaction-generated energy. The inlet conditions are known for the fuel and air, but the wall temperature at the inlet is a dependent variable, thereby requiring a shooting method for solution.

We have not yet been able to obtain a convergent solution without making approximations in the model. The general problem involves a double iteration, one for the postulated inlet temperature which satisfies the boundary condition of overall conservation of energy and the other for the temperature distribution in the integral. Our initial numerical solution [J.L.-P. Chen and Churchill (1972)], although not convergent, did yield the correct behavior semi-quantitatively, and unexpectedly predicted the existence of seven stable solutions. These multiple stationary states arise from differences in the eighth significant figure of the trial value of the inlet wall temperature. The inlet wall temperatures which lead to stable solutions are thus equivalent to eigenvalues. (The existence of the multiple stationary states has subsequently been confirmed experimentally for both gaseous and liquid fuels).

We have subsequently [Choi and Churchill (1979)] obtained a convergent solution by approximating and analytically evaluating the integral term. However in so doing we lost five of the stable solutions. We have also carried out calculations using an experimental wall temperature profile and thereby avoiding the computational difficulties associated with the integro-differential component of the model. Such calculations, although stiff in the sense of a 25 to 1 ratio in the eigenvalues, have converged satisfactorily [see, for example, Tang and Churchill (1981)]. The classical postulates of quasi steady states or local equilibrium for the free radicals were not found to be valid for these conditions.

It appears from this experience that an integral term, such as that arising from radiant interchange between surfaces may be a more serious source of computational difficulty than widely separated eigenvalues or split boundary conditions.

The principal idealization in the above modelling is that of plug flow. Two-dimensional modelling with eddy diffusivities for the radial transport of momentum, energy and species would undoubtedly be more realistic. The principal advantage would be the elimination of the present uncertain prediction of the reaction-dependent heat transfer coefficient for the wall. Calculation of the transient behavior would also be of practical value in connection with ignition, quenching and physical perturbations. Both radial and transient modelling require the use of partial differential equations without any relaxation of the previously mentioned difficulties.

Simultaneous Radiation and Conduction in Dispersed Media

Radiative transfer through a dispersed media depends on scattering, absorption and reradiation. This behavior can be modelled by an integro-differential equation involving one to three dimensions and two angles. For the overly idealized case of isotropic scattering, this integro-differential equation can be approximated by the Poisson equation [see, for example, Chu and Churchill (1956) and Chu, Pang and Churchill (1963)]. However for anisotropic scattering this approximation breaks down near the boundaries, precisely where the results are of most interest. Lumping the radiation in two directions as proposed by Schuster (1905) or in three as proposed and Churchill (1955) permits approximate modelling by a set of non-linear, ordinary differential equations. The parameters of the problem all vary

with the wavelength of the radiation, but mean values can be postulated as a further approximation. The transmission of solar radiation through clouds to the surface of the earth is one application. In a thermal insulation or packed bed, radiative transfer may also be coupled with thermal conduction. To be rigorous, conduction in the solid and vapor phases must be considered individually, but as an approximation, a combined, effective conductivity may be used to represent both phases.

These simplified models for radiative transfer and combined radiative and conductive transfer may still pose stiff problems, particularly if the reflectivity of one or both of the boundaries is near unity [see, for example, Larkin and Churchill (1959) and J.C. Chen and Churchill (1963)].

Thermal Regenerators

The fully developed temperature profile in a thermal regenerator can be modelled as stationary. If the solid and fluid are postulated to be at thermal equilibrium locally, and if diffusion is neglected in both phases, the process can be modelled algebraically as a step function. K. Chen and Schiesser (1980) solved the transient model [which they call the advection equation] for this limiting behavior numerically, but stiffness resulted in considerable inaccuracy. The superimposed effect of diffusion can be modelled by one ordinary differential equation and finite convective resistance by two.

This stationary model is not valid for the fundamentally transient behavior near the inlet and outlet of the regenerator, which requires modelling by one partial differential equation in the case of diffusion and two in the case of convection. Cyclic operation introduces the further complication of a different initial temperature distribution for each cycle,

although an asymptotic initial distribution will be approached after many cycles.

Abbrecht, Churchill and Chu (1957) have reviewed some of the analytical solutions in series form and in terms of high-order functions. Interestingly, the stiffness of the various models appears to have a counterpart in the numerical evaluation of these analytical solutions.

Wave-like Mass Transfer

Ion exchange and gas-phase chromatography bear some analogy to thermal regeneration but are generally more complicated.

The dispersion of CO_2 and O_2 in the process of respiration, as described by Boy (1981), is also somewhat analogous, although the conical shape of the bronchial tubes, etc., introduces a different set of complications.

Thermoacoustic Waves

A sonic compression wave is generated in an unconfined gas by a step increase in the temperature of a bounding surface. If the gas is confined, the wave reflects repeatedly. Ozoe and Churchill (1980) failed to obtain a convergent numerical solution for this problem owing to stiffness, although their computed pseudo steady-state pressures, velocities and temperatures are self-consistent.

Forced Convection

Developing heat transfer in forced convection poses a fundamentally two-dimensional problem. Graetz derived a series solution for the temperature field and heat transfer coefficient in fully developed laminar flow in a round tube due

to a step in wall temperature. However, this series does not converge for short distances from the onset of heating. Lévéque used a similarity transformation to derive an approximate solution for the inlet region, which is correct in the limit but not for finite distances. Worsøe-Schmidt developed a series solution in the form of a perturbation on the Lévéque model which is valid for small distances but not for large. A number of numerical solutions have been attempted but none are accurate as the inlet is approached, owing to the ever-decreasing thickness of the boundary layer.

Many closely related problems have been considered for other geometries, other boundary conditions and developing flow. They all have the same limitations. More detailed description of these problems as well as references to the original work are given by Gröber, et al. (1961), Churchill and Usagi (1972) and Churchill and Ozoe (1973a, 1973b).

Transient Conduction

Transient thermal conduction in multiple, solid media, such as an insulated semi-infinite region subjected to a step in surface temperature [see, for example, Churchill (1965)] may pose computational difficulties.

The non-linear boundary condition provided by a phase transition, such as in the freezing of wet soil, particularly in two or three dimensions, may introduce computational problems related to the location of the phase boundary [see, Churchill and Gupta (1977)].

Transient Heating of a Dispersion

The heating of a dispersion of suspended particles by radiation with conduction to the surrounding air poses computational problems because of the

nearly discrete shifts from single-particle heating, to a pseudo steady-state to interparticle effects [see Sleicher and Churchill (1956)].

Instability

Physical instability may cause computational difficulties which are similar to those associated with stiffness. For example, the mode of circulation for natural convection in an enclosure heated from below changes from multiple roll cells with axes parallel to the shorter horizontal dimension to a single circulation with its axis parallel to the axis of inclination. The computed values oscillate chaotically instead of converging for inclinations near the critical value [see, for example, Ozoë, Yamamoto and Churchill (1979)].

Summary and Conclusions

I have now completed my physical description of a representative set of problems in heat transfer whose numerical solution is known or suspected to encounter computational problems. I have also noted some problems in mass transfer which are computationally analogous and some in which heat transfer is coupled with momentum transfer, mass transfer and chemical reactions. I have, however, avoided the discussion of heat transfer in laminar, tubular and packed-bed reactors, per se, out of deference to other speakers.

Most important unsolved problems in heat transfer (as well as in mass transfer, momentum transfer and chemical kinetics) are found to be either transient or multidimensional.

Shock, detonation, deflagration and thermoacoustic waves, thermal regeneration, ion exchange, gas-phase chromatography, respiration, and radiantly stabilized combustion, with dispersion due to diffusion, chemical reactions and/or interphase transfer, whether stationary or transient,

pose classical problems of stiffness. That is, the stiffness is associated with different terms with widely varying time constants or the equivalent.

The simplified model for radiative and conductive transfer through dispersed material, and the illustrative model for transient radiant heating of a dispersion of particles are similarly stiff owing to discordant mechanisms of heat transfer. This same effect occurs in the transient heating of an insulated semi-infinite media and in the freezing of a semi-infinite region of wet soil. (Locating the freezing front further complicates the latter problem, particularly in two or three dimensions).

On the other hand the original integro-differential model for radiative transfer through a dispersion and the integro-differential equation in the model for radiantly stabilized combustion generate even more severe computational difficulties.

Split boundary conditions are a source of difficulty in some boundary layer and heat exchanger problems, and also in solving the model for radiantly stabilized combustion.

The generation of a thermoacoustic wave by a step-function in temperature at the boundary appears to pose a uniquely stiff problem.

Physical instability, such as that associated with low values of the Prandtl number in free and natural convection and with changes in the mode of circulation in natural convection, is another severe source of computational difficulty.

In conclusion, I hope that this physical description of important and computationally difficult problems in heat transfer has provided some new challenges for you.

References

- Abbrecht, P.H., Churchill, S.W. and Chu, C.-M. (1957). Regenerative Heat Transfer in Two- and Three-Dimensional Flow through Porous Media, Ind. Eng. Chem., 49, 1007-12.
- Bird, R.B., Stewart, W.E. and Lightfoot, E.N. (1960). Transport Phenomena. Wiley, New York.
- Boy, G. (1981). A Numerical Method for the Convective and Diffusive Process Simulation, Proc. 2nd World Congress of Chemical Engineering, Vol. V, pp. 327-30.
- Brian, P.L.T. (1963). Turbulent Pipe Flow Heat Transfer with a Simultaneous Chemical Reaction of Finite Rate, AIChE J1., 9, 831-841.
- Calderbank, P.H. (1954). Some Problems in the Design of Light-Hydrocarbon Pyrolysis Coils, Chem. Eng. Prog. Symp. Ser. No. 9, 50, 52-7.
- Chen, J.C. and Churchill, S.W. (1963). Radiant Heat Transfer in Packed Beds, AIChE J1., 9, 35-41.
- Chen, J.L.-P. and Churchill, S.W. (1972). A Theoretical Model for Stable Combustion inside a Refractory Tube. Combustion and Flame, 18, 27-36.
- Chen, K.L. and Schiesser, W.E. (1980). Upwind Approximations in the Numerical Methods of Lines. Integration of Hyperbolic Partial Differential Equations, Lehigh University Report DSS2, Bethlehem, Pennsylvania.
- Choi, B. and Churchill, S.W. (1979). A Model for Combustion of Gaseous and Liquid Fuels in a Refractory Tube. Seventeenth Symposium (International) on Combustion, 915-25. The Combustion Institute, Pittsburgh.
- Chu, C.M. and Churchill, S.W. (1955). Numerical Solution of Problems in Multiple Scattering of Electromagnetic Radiation. J. Phys. Chem., 59, 855-63.
- Chu, C.M. and Churchill, S.W. (1956). Multiple Scattering by Randomly Distributed Obstacles—Methods of Solution. Trans. IRE (PGAP), AP-4, 1942-8.
- Chu, C.M., Pang, S.C. and Churchill, S.W. (1963). A Variable-Order Diffusion-Type Approximation for Multiple Scattering. In Electromagnetic Scattering (ed. M. Kerker), 503-22, MacMillan, New York.
- Churchill, S.W. (1965). Bounded and Patched Solutions for Boundary Value Problems. AIChE J1., 11, 431-5.
- Churchill, S.W. (1977). Comprehensive Correlating Equations for Heat, Mass and Momentum Transfer in Fully Developed Flow in Tubes. Ind. Eng. Chem. Fundam., 16, 109-16.

- Churchill, S.W. (1982a). Combined Free and Forced Convection—Immersed Bodies. Chapter 2.5.9 in Heat Transfer Design Handbook (ed. E.V. Schlunder). Hemisphere, Washington, D.C.
- _____ (1982b). Combined Free and Forced Convection—Channels. Chapter 2.5.10 in Heat Transfer Design Handbook (ed. E.V. Schlunder). Hemisphere, Washington, D.C.
- Churchill, S.W. and Gupta, J.P. (1977). Approximations for Conduction with Freezing and Melting. Int. J. Heat Mass Transfer, 20, 1251-3.
- Churchill, S.W. and Ozoe, H. (1973a). Correlations for Laminar Forced Convection with Uniform Heating in Flow over a Plate and in Developing and Fully Developed Flow in a Tube. J. Heat Transfer, 95B, 78-82.
- _____ (1973b). Correlations for Laminar Forced Convection in Flow over an Isothermal Flat Plate and in Developing and Fully Developed Flow in an Isothermal Tube. J. Heat Transfer, 95C, 416-419, 573.
- Churchill, S.W. and Usagi, R. (1972). A General Expression for the Correlation of Rates of Transfer and Other Phenomena. AIChE J., 18, 1121-8.
- Fair, J.R., Jr. (1955). Ph.D. Thesis. University of Texas, Austin.
- Fristrom, R.M. (1960). Structure of Laminar Flames. Sixth Symposium (International) on Combustion, 96-110. The Combustion Institute, Pittsburgh.
- Gröber, H., Erk, S. and Grigull, R. (1961). Fundamentals of Heat Transfer (3rd Edn.) transl. by Moszynski, J.R. McGraw-Hill, New York.
- Hahn, W.A. and Wendt, J.O.L. (1982). Analysis of the Flat Laminar Opposed Jet Diffusion Flame with Finite Rate Detailed Kinetics. AIChE J., in press.
- Jakob, M. (1957). Heat Transfer, Vol. II, 217f. Wiley, New York.
- Kerker, M., ed. (1963). Electromagnetic Scattering. MacMillan, New York.
- Larkin, B.K. and Churchill, S.W. (1959). Heat Transfer by Radiation through Porous Insulations. AIChE J., 5, 467-74.
- Moyle, M.P., Churchill, S.W. and Morrison, R.B. (1960). Detonation Characteristics of Hydrogen-Oxygen Mixtures, AIChE J., 92-6.
- Nicholls, J.A. (1960). Stabilization of Gaseous Detonation Waves with Emphasis on the Ignition Delay Zone, Ph.D. Thesis, University of Michigan, Ann Arbor.

- Ozoe, H., Sato, N. and Churchill, S.W. (1980). The Effect of Various Parameters on Thermoacoustic Convection. Part I - Vertical One-Dimensional Flow. Chem. Eng. Commun., 5, 203-21.
- Ozoe, H., Yamamoto, K. and Churchill, S.W. (1979). Three-Dimensional Numerical Analysis of Natural Convection in an Inclined Channel with a Square Cross Section. AIChE JI., 25, 709-16.
- Saville, D.P. and Churchill, S.W. (1970). Simultaneous Heat and Mass Transfer in Free Convection Boundary Layers. AIChE JI., 16, 268-73.

**Integration of the Stiff, Boundary Valued ODE'S for
the Laminar, Opposed Jet Diffusion Flame**

by

**W.A. Hahn¹⁾ and J.O.L. Wendt²⁾
Department of Chemical Engineering
University of Arizona
Tucson, Arizona 85721**

1) **Current Address: Exxon Production Research Center
P.O. Box 2189
Houston, TX 77001**

2) **To whom correspondence should be addressed.**

ABSTRACT

Detailed models of the flat, laminar, opposed jet diffusion flame involve the solution of the momentum, energy and species conservation equations coupled with stiff chemical kinetics. The problem has self similar solutions and can be solved through numerical integration of a set of second order, stiff, boundary valued, ordinary differential equations, each with a regular first order turning point arising from convection.

Finite difference discretization (in the spatial domain) and expansion of the reaction rate source terms in a Taylor series about the backward iteration (in the temporal domain), leads to a matrix equation, the solution of which is obtained through LU decomposition. Modification of standard discretization to allow convergence to be obtained with a minimum of grid points was required, and is described in detail.

Predicted profiles of major and minor species provide useful insight into the use of the laminar opposed jet diffusion flame configuration to investigate detailed kinetic mechanisms under a wide range of conditions.

The Problem

The laminar opposed jet diffusion flame, Figure 1 is a combustion configuration that possesses a number of useful attributes. First, it allows a flat flame to be established in space, without direct influences of burners or possible catalytic surfaces; second, it allows investigation kinetic mechanisms occurring in hot fuel rich regions and over a wide range of conditions, which due to flammability limitations, cannot be established in the premixed mode; third, it may be thought to represent a prototype model of laminar flamelets, each of which is strained in its own plane, and which represent the reaction zones important in turbulent diffusion flames. Its usefulness as a tool to test and corroborate detailed kinetic mechanisms rests on its ability to be modeled and on the power of numerical procedures to enable accurate solution of the ensuing stiff ordinary boundary valued differential equations to be achieved.

Unlike previous work (Fendell, 1965; Krishnamurthy and Williams, 1974) which has focused on analytical solutions of models utilizing simplified combustion chemistry with either infinitely rapid chemical reactions or a simple reaction with a very high activation energy (Peters, 1978); this paper focuses on numerical procedures necessary to solve the problem for an arbitrary large set of reactions describing the free radical combustion chemistry pertinent to this flame. The equations describing the laminar opposed jet diffusion flame comprises a severe test of any numerical integration scheme. The problem is a boundary valued problem with concentrations and temperature set at $\pm\infty$. Moreover, this is one of the cases in nature where there is direct coupling between the energy and the momentum balances through the changes in gas density ρ . An accurate prediction of the (non linear) velocity profile is essential since this has a

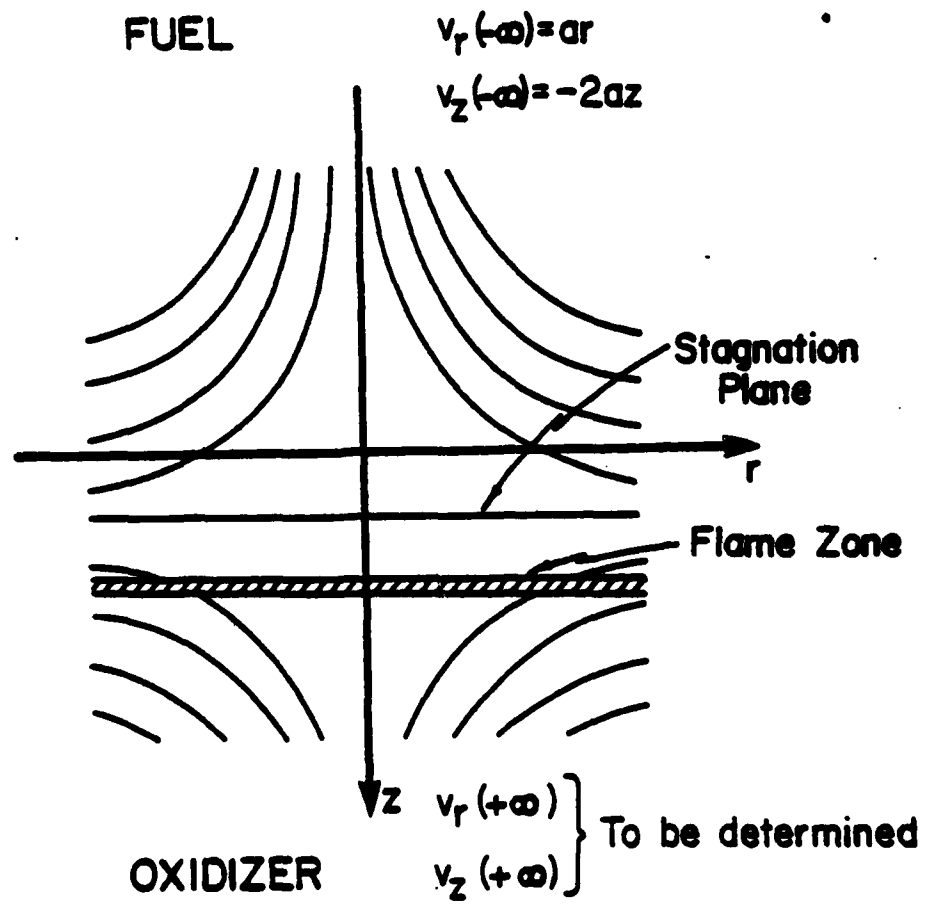


Figure 1. Schematic of the Laminar Opposed Jet Diffusion Flame.

first order influence on all other profiles. Inclusion of the effects of viscosity and other dynamic effects on the overall concentration profiles, together with chemical reaction, distinguishes this problem from that of the one dimensional premixed flat flame, where the velocity profile can be calculated from kinematic considerations alone.

The reaction zone thickness depends on a parameter, ϵ , the rate of stretching, and very steep spatial gradients in concentrations, temperature, reaction rates are the norm rather than the exception. A detailed combustion chemistry mechanism typically consists of many free radical reactions with characteristic times differing by over thirty orders of magnitude at a point in space. This naturally introduces problems associated with stiffness, and special techniques must be implemented to overcome them. The purpose of this paper is first to survey the problem qualitatively in order to present an engineering example of where and how stiffness manifests itself in practice, and second, to present the approaches used to solve this particular problem. These approaches address both the problem of stiffness and that associated with the steep gradients occurring in the spatial domain near the reaction zone.

Self Similar Solution of the Governing Equations

A complete description of the system shown on Figure 1 involves the simultaneous solution of the following equations in cylindrical coordinates:

1. Continuity:

$$\frac{\partial \rho}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (\rho r v_r) + \frac{\partial}{\partial z} (\rho v_z) = 0 \quad (1)$$

2. r momentum:

$$\rho \frac{\partial v_r}{\partial t} + \rho v_r \frac{\partial v_r}{\partial r} + \rho v_z \frac{\partial v_r}{\partial z} = - \frac{\partial P}{\partial r} - \frac{1}{r} \frac{\partial}{\partial r} (r \tau_{rr}) - \frac{\tau_{\theta\theta}}{r} + \frac{\partial \tau_{rz}}{\partial z} \quad (2)$$

3. z momentum:

$$\rho \frac{\partial v_z}{\partial t} + \rho v_r \frac{\partial v_z}{\partial r} + \rho v_z \frac{\partial v_z}{\partial z} = - \frac{\partial P}{\partial z} - \frac{1}{r} \frac{\partial}{\partial r} (r \tau_{rz}) + \frac{\partial \tau_{zz}}{\partial z} \quad (3)$$

4. Energy:

$$c_p C_p \left(\frac{\partial T}{\partial t} + v_r \frac{\partial T}{\partial r} + v_z \frac{\partial T}{\partial z} \right) = + \frac{1}{r} \frac{\partial}{\partial r} (rk \frac{\partial T}{\partial r}) + \frac{\partial}{\partial z} (k \frac{\partial T}{\partial z}) + \sum_{j=1}^{nr} r_j \Delta g_j \quad (4)$$

5. Species:

$$\frac{\partial (cx_A)}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (crx_A v_r) + \frac{\partial}{\partial z} (cx_A v_z) = \frac{1}{r} \frac{\partial}{\partial r} (rcD_A \frac{\partial x_A}{\partial r}) + \frac{\partial}{\partial z} (cD_A \frac{\partial x_A}{\partial z}) + R_A$$

(A = 1 NS)

(5)

6. Ideal Gas Law:

$$P/c = RT$$

In the above equations, pseudo binary diffusion has been assumed and radiative heat losses and thermal diffusion have been neglected following Field et. al. (1967). The time dependence has been retained since we wish to solve these equations from an initial guess at $t = 0$ to the steady state at $t \rightarrow \infty$.

The boundary conditions to equations (1) through (6) are:

$$v_z = -2\epsilon z \quad z = -\infty \text{ for all } r\text{'s} \quad (6)$$

$$v_r = \epsilon r \quad z = \pm\infty \text{ for all } r\text{'s} \quad (7)$$

$$P = P_\infty$$

$$T = T_\infty \quad z = +\infty \text{ for all } r\text{'s} \quad (8)$$

$$T = T_\infty \quad z = -\infty \text{ for all } r\text{'s} \quad (9)$$

$$x_A = x_{A1} \quad z = +\infty \text{ for all } r\text{'s} \quad (10)$$

$$x_A = x_{A2} \quad z = -\infty \text{ for all } r\text{'s} \quad (11)$$

It can be shown that Equations (1) through (5) can be reduced to a system of ordinary differential equations by assuming self similar solutions of the following forms:

$$v_r = \epsilon r \psi(z) ; v_z = v(z) \quad (12)$$

$$\frac{\partial P}{\partial r} = f(r) ; \frac{\partial P}{\partial z} = g(z) \quad (13)$$

$$\rho = \rho(z) ; T = T(z) ; x_A = x_A(z) \quad (14)$$

Furthermore, since

$$\frac{\partial P}{\partial r} = f(r) = \epsilon^2 \rho_\infty r$$

it can be shown (Hahn, 1979), that the proposed self similar solution is consistent with the equations, which now become totally independent of r. It is convenient to define a dimensionless temperature as $\theta = \frac{T - T_0}{T_0}$, to yield the following differential equations to be solved:

Mass:

$$\frac{\partial \rho}{\partial t} = - \frac{\partial}{\partial z} (\rho v) + 2 \epsilon \rho \psi \quad (15)$$

r momentum:

$$\frac{\partial \psi}{\partial t} = \frac{\mu}{\rho} \frac{\partial^2 \psi}{\partial z^2} + \left(\frac{1}{\rho} \frac{\partial \mu}{\partial z} - v \right) \frac{\partial \psi}{\partial z} - \epsilon \psi^2 + \frac{\rho_\infty}{\rho} \epsilon \quad (16)$$

Energy:

$$\frac{\partial \theta}{\partial t} = \frac{k}{c C_p} \frac{\partial^2 \theta}{\partial z^2} + \left(\frac{1}{c C_p} \frac{\partial k}{\partial z} - v \right) \frac{\partial \theta}{\partial z} + \frac{1}{T_0 c C_p} \sum_{j=1}^{nr} r_j \Delta h_j \quad (17)$$

Species:

$$\frac{\partial x_A}{\partial t} = D_A \frac{\partial^2 x_A}{\partial z^2} + \left(\frac{1}{c} \frac{\partial}{\partial z} (c D_A) - v \right) \frac{\partial x_A}{\partial z} + \left(- \frac{1}{c} \frac{\partial c}{\partial t} - 2 a \psi - \frac{1}{c} \frac{\partial (c v)}{\partial z} \right) x_A + \frac{1}{c} R_A$$

(A = 1 NS)

(18)

The z momentum equation plays no role other than to calculate $\frac{\partial P}{\partial z}$.

Of interest is the solution as $t \rightarrow \infty$ when the left hand sides of these equations are equal to zero. This is the desired one dimensional steady state solution. It should be noted that all physical and transport properties are allowed to vary with position. Simplifying assumptions with respect to diffusion can be made, depending on which level of complication for D_A , the pseudo binary diffusion coefficient, is appropriate.

Numerical Solution

The equations (16) through (18) were solved numerically in a time dependent, fully implicit and fully coupled mode. These equations have the form

$$\frac{\partial c_i}{\partial t} = A_i(z) \frac{\partial^2 c_i}{\partial z^2} + B_i(z) \frac{\partial c_i}{\partial z} + D_i(z) C_i + R_i(z) \quad (19)$$

$i = 1 \dots NS+2$

where the coefficients are given:

| | A_i | B_i | D_i | R_i |
|----------|--------------------|--|--|---|
| x_A | D_A | $\frac{1}{c} \frac{\partial}{\partial z} (cD_A) - v$ | $-\frac{1}{c} \frac{\partial c}{\partial t} - 2\epsilon\psi - \frac{1}{c} \frac{\partial(cv)}{\partial z}$ | $\frac{1}{c} R_A$ |
| θ | $\frac{k}{cC_p}$ | $\frac{1}{cC_p} \frac{\partial k}{\partial z} - v$ | 0 | $\frac{1}{T_0 cC_p} \sum r_j \Delta h_j$ |
| ψ | $\frac{\mu}{\rho}$ | $\frac{1}{\rho} \frac{\partial \mu}{\partial z} - v$ | 0 | $-\epsilon\psi^2 + \frac{\rho_\infty}{\rho} \epsilon$ |

(20)

If the domain in which a solution is sought is given by

$$- \ell/2 < z < + \ell/2$$

the boundary conditions for all the x_A and θ values are specified, while for

ψ :

$$\begin{aligned} \psi &= 1 & z &= - \ell/2 \\ \psi &= \sqrt{\frac{\rho - \rho_\infty}{\rho_\infty}} & z &= + \ell/2 \end{aligned}$$

Continuity (Equation 15) is solved separately in the spatial domain

$$\frac{\partial}{\partial z} (\rho v) = - 2\epsilon\rho\psi - \frac{\partial\rho}{\partial t} \quad (21)$$

where

$$\frac{\partial\rho}{\partial t} = \frac{(MW)P}{R} \frac{\partial(\frac{1}{T})}{\partial t} \quad (22)$$

Stiffness in Equation (19) arises through the term $R_1(z)$ which is the forcing function due to detailed combustion chemical reactions. The coefficient $B_1(z)$ changes sign in the neighborhood of the stagnation plane, causing a regular first order turning point.

An explicit approach to solve Equation 19 is to evaluate A_1 , B_1 , and R_1 using values of a previous iteration and solve for C_1 , thus decoupling the system of equations and leading to inversion of a series of tridiagonal matrices. This was attempted but failed. Tyson (1964) developed a technique for solution of systems of stiff initial valued problems in which the term $R_1(z)$ is expanded about the backward time step. This idea was later used by

Wendt et. al. (1979) to solve systems of stiff boundary valued problems and forms the centerpiece for the solution method utilized in this work. Thus

$$R_i^{k+1}(z) = R_i^k + \sum_{j=1}^N \frac{\partial R_i}{\partial C_j}^k (C_j^{k+1} - C_j^k) \quad (23)$$

with $N = NS+2$ and k refers to the time step. This expansion is performed at each grid point p .

Standard Finite Difference Discretization

Both a collapsing and an equally spaced grid were used with standard finite difference discretization. The appropriate formulas are

$$\frac{dC_i}{dz} = \frac{\Delta z_{p-1}}{\Delta z_{p+1}(\Delta z_{p+1} + \Delta z_{p-1})} C_{i,p+1}^{k+1} - \frac{\Delta z_{p-1} - \Delta z_{p+1}}{\Delta z_{p-1} \Delta z_{p+1}} C_{i,p}^{k+1} - \frac{\Delta z_{p+1}}{\Delta z_{p-1}(\Delta z_{p+1} + \Delta z_{p-1})} C_{i,p-1}^{k+1} \quad (24)$$

and

$$\frac{d^2 C_i}{dz^2} = \frac{2}{\Delta z_{p+1} + \Delta z_{p-1}} \left[\frac{C_{i,p+1}^{k+1} - C_{i,p}^{k+1}}{\Delta z_{p+1}} - \frac{C_{i,p}^{k+1} - C_{i,p-1}^{k+1}}{\Delta z_{p-1}} \right] \quad (25)$$

The finite difference form of Equation 19 now becomes

$$\left[\alpha_{i,p}, \beta_{ip}^{-1} \frac{1}{\Delta t}, \gamma_{i,p} \right] \begin{bmatrix} \Delta C_{i,p-1} \\ \Delta C_{i,p} \\ \Delta C_{i,p+1} \end{bmatrix} + \left[\frac{\partial R_i}{\partial C_j} \right]_{p-}^k \begin{bmatrix} \Delta C_{1,p} \\ \Delta C_{2,p} \\ \vdots \\ \Delta C_{N,p} \end{bmatrix} = -F_{i,p}^k \quad (26)$$

with

$$\alpha_{1,p} = \frac{1}{\Delta z_{p-1}(\Delta z_{p+1} + \Delta z_{p-1})} [2 - \Delta z_{p+1} A_{1,p}] \quad (27)$$

$$\beta_{1,p} = - \left[\frac{2}{\Delta z_{p+1} + \Delta z_{p-1}} \left(\frac{1}{\Delta z_{p+1}} + \frac{1}{\Delta z_{p-1}} \right) + \frac{(\Delta z_{p-1} - \Delta z_{p+1}) A_{1,p}}{\Delta z_{p+1} \Delta z_{p-1}} - B_{1,p} \right] \quad (28)$$

$$\gamma_{1,p} = \frac{1}{\Delta z_{p+1}(\Delta z_{p+1} + \Delta z_{p-1})} [2 + \Delta z_{p-1} A_{1,p}] \quad (29)$$

and

$$\Delta C_{j,p} = C_{j,p}^{k+1} - C_{j,p}^k \quad (30)$$

$F_{t,p}^k$ gathers all remaining terms known at the k'th time step and $\alpha_{1,p}$, $\beta_{1,p}$, and $\gamma_{1,p}$ as well as $\frac{\partial R_1}{\partial C_j}$ are also evaluated at the known k'th step. If m is the number of grid points equations (26) represents a set of $(N \cdot m)$ simultaneous algebraic equations for the values of the increments $\Delta C_{1,p}$ for a time step Δt . The system can be rearranged, using matrix notation, as follows (Wendt et. al., 1979):

$$\underline{A} \underline{\Delta C} = - \underline{F} \quad (31)$$

where \underline{A} is a $(m \times N) \times (m \times N)$ block tridiagonal matrix, i.e. an $m \times m$ matrix in which each element is a $N \times N$ matrix. The diagonal $(N \times N)$ blocks are full, nondiagonally dominant (depending on Δt) matrices while the off diagonal blocks are diagonal matrices.

Classical block tridiagonal matrix inversion logic (Wendt et. al., 1979) was not successful in solving the flame problems attacked in this work. However, use was made of the fact that \underline{A} is also a band matrix and therefore

the solution to (31) can be obtained employing LU decomposition, which was found to be very efficient for this type of problem (Peaceman, 1977). This method also saves core storage, since only the band of dimension $2N_x(m \times N)$ need be treated.

With the equally spaced grid, $\Delta z_{p-1} = \Delta z_{p+1}$, and a large number of grid points must be used when the reaction zone is thin and gradients are steep. For example, with a CO flame, 8 species and 7 reactions and with $\epsilon = 10 \text{ sec}^{-1}$, 101 grid points were employed to obtain a solution, while 35 grid points were insufficient for proper resolution of the peaks.

Since systems involving 35 species or more are common in this type of problem, it was clear that a discretization allowing fewer grid points was essential. This forced utilization of a collapsing grid.

The collapsing grid is shown in Figure 2 and was automatically adjusted as follows. The location of the peak in the temperature profile was determined and a grid was created to the left with a specified expansion factor. The expansion factor for the right side was then determined such that both intervals next to the peak temperature location were equal and such that an equal number of grid points were located on each side. This method allowed the CO opposed jet diffusion flame to be solved with 35 grid points, provided that the stretching rate ϵ , was less than 7 sec^{-1} . In these cases the equally spaced grid with 35 grid points was too coarse in the neighborhood of the peak temperature to allow convergence. However, at $\epsilon > 7 \text{ sec}^{-1}$ the collapsing grid was also prone to problems due to nonsymmetric numerical diffusion of errors, and this led to oscillations and an inability to satisfy the boundary conditions. This is described in detail below.

The burner spacing, ℓ , or domain in which the problem was being solved, was automatically adjusted so that there was not net flux of heat or any

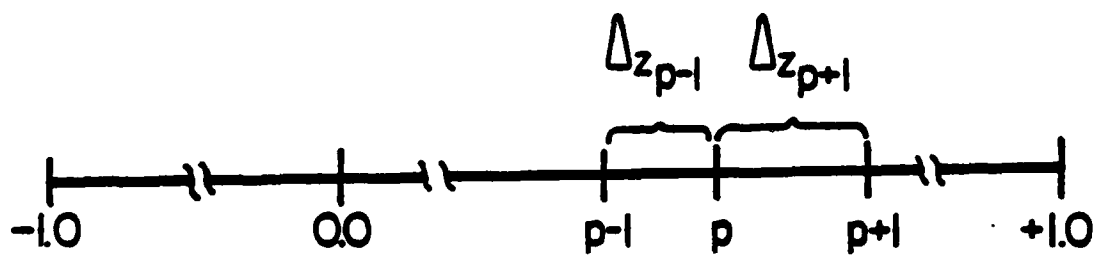


Figure 2. Collapsing Grid Utilized for Finite Difference Solution.

species into either of the burners. Thus the distance l could be increased by 50% if, at any iteration, any net flux had the incorrect sign at

$z = +l$ or $z = -l$. Likewise, l could be automatically decreased by 2/3 if the temperature profile indicated that greater resolution was appropriate. Whenever the grid had to be adjusted, the new values of all variables were obtained by linear interpolation between values at the previous grid points.

The original initial estimate of profiles at $t = 0$ was from the classical Burke Shumann flame with an infinitesimally thin reaction zone, together with arbitrary free radical profiles to facilitate ignition. Much computer time can be saved if intermediate nonconverged or converged results are saved and used as the initial guesses for the next run. The adjustable time step was controlled entirely by the temperature profile, allowing changes of 1% at any point in any time interval. At convergence to steady state, determined by relative changes of all variables, the time step had become very large, greater than 1 sec. This typically occurred, from a 'reasonable' initial guess within 200 to 400 iterations, from an initial $\Delta t = 10^{-6}$ sec.

Modified Central Difference Discretization

When solving the model corresponding to the opposed jet diffusion flame with a high rate of stretching, numerical oscillations were observed in the temperature profile, the ψ profile and the reactant profiles. These undulations occur close to the boundaries of the domain and do not satisfy the boundary conditions and thus prevent convergence. These instabilities originated from the collapsing grid and diffusion of numerical errors towards the boundaries. Yet, a collapsing grid is essential in order to handle the sharp temperature profiles in the center with less than 35 grid points overall. One solution was to use 100 equally spaced grid points and this appeared to allow convergence. However, this large number of grid points

constrained the program to solution of relatively small kinetic problems (due to core storage limitations) and did not allow investigation of hydrocarbon plus fuel nitrogen chemistry together. Since the latter was the key aspect of modeling in this work, these results put us on the horns of a dilemma.

This dilemma was ultimately resolved by developing a modified central difference discretization in which the first and second derivations are given by (for equally spaced grids):

$$\frac{dC_i}{dz} = \frac{W_{1,p+1} C_{i,p+1} - W_{1,p-1} C_{i,p-1}}{2h} \quad (32)$$

$$\frac{d^2 C_i}{dz^2} = \frac{W_{2,p+1} C_{i,p+1} - 2 C_{i,p} + W_{2,p-1} C_{i,p-1}}{h^2} \quad (33)$$

with the weighting functions $W_{1,p+1}$, $W_{1,p-1}$, $W_{2,p+1}$ and $W_{2,p-1}$ calculated as described in the Appendix. The essence of this approach is to approximate the differential equations to be solved within a mesh interval, rather than the solutions to the equations. Thus, this discretization is exact for linear ODE's with constant coefficients and constant forcing functions while the conventional discretization is not. Since the opposed jet diffusion flame equations are certainly not linear, the modified discretization is not exact, but is superior as long as the solutions to the equations behave more like a sum of exponentials than a quadratic function of z .

Implementation of the modified discretization resulted in solution of the opposed jet diffusion flame model, with CO kinetics and with a stretching rate $\epsilon = 10 \text{ sec}^{-1}$, with 35 grid points. This was not possible with standard discretization with either collapsing or uniform grids, in which cases no convergence could be achieved. With uniform standard discretization convergence was obtained with 101 grid points. Therefore the modified

discretization allowed the minimum number of mesh points to be reduced from 101 to 35 and this allowed development of a general purpose laminar opposed jet diffusion flame code.

The program also allowed the laminar opposed jet diffusion flame model for a $\text{CH}_4/\text{O}_2/\text{N}_2$ flame to be solved. This involved 16 species and 39 (reversible) reactions. Core limitations required the use of not more than 35 grid points for this case. Neither the evenly spaced grid nor collapsing grid with standard discretization allowed a convergent solution to be found. The modified discretization did converge, indicating again the necessity of this approach.

General Purpose Code

The general purpose code developed allows an arbitrary reaction set containing up to 150 reactions and 70 species to be interpreted and solved for the laminar opposed jet flame configuration. Diffusion coefficients, thermal conductivities, viscosities and thermochemical properties are supplied in the program. These are calculated from the proper Lennard Jones and Stockmayer intermolecular potential parameters and from the NASA equilibrium tape. Appropriate mixing rules including the bifuraction approximation for multicomponent diffusion are incorporated. Table 1 shows a sample input for a carbon monoxide reaction set utilized in the results presented here. This reaction set was obtained from a recent review of $\text{CO}/\text{H}_2/\text{O}_2$ kinetics by Corley and Bowman (1982). The data can be listed in any order and a free format input is utilized to aid in the use of the program. The program requires the stretching rate as an input parameter and adjusts the burner spacing automatically such that there is no net flux of any species back into either of the burners. Appropriate velocities are calculated and this is used in the design of the experimental configuration.

Table 1 Data Input to LOJDF Code

| | | | | | | | |
|------------|-----|-------------|--------|----------|-------|------|-----|
| STOICH,COR | 1, | CO | + O | + M | = CO2 | + M | |
| FRATCO,COR | 1, | 0.3800E+25, | -3.00, | 6.170, | | | |
| RRATCO,COR | 1, | 0.3819E+31, | -4.00, | 133.551, | | | |
| STOICH,COR | 2, | CH2O | + M | | = HCO | + H | + M |
| FRATCO,COR | 2, | 0.3310E+17, | 0.00, | 81.020, | | | |
| RRATCO,COR | 2, | 0.1929E+12, | 1.00, | -10.552, | | | |
| STOICH,COR | 3, | HCO | + M | | = H | + CO | + M |
| FRATCO,COR | 3, | 0.1550E+15, | 0.00, | 14.675, | | | |
| RRATCO,COR | 3, | 0.7430E+11, | 1.00, | -1.863, | | | |
| STOICH,COR | 4, | H2 | + M | | = H | + H | + M |
| FRATCO,COR | 4, | 0.2080E+16, | 0.07, | 103.830, | | | |
| RRATCO,COR | 4, | 0.2203E+12, | 1.07, | -1.511, | | | |
| STOICH,COR | 5, | H2O | + M | | = OH | + H | + M |
| FRATCO,COR | 5, | 0.2300E+25, | -2.00, | 122.600, | | | |
| RRATCO,COR | 5, | 0.4978E+20, | -1.00, | 1.905, | | | |
| STOICH,COR | 6, | O | + O | + M | = O2 | + M | |
| FRATCO,COR | 6, | 0.1000E+19, | -1.00, | 0.000, | | | |
| RRATCO,COR | 6, | 0.6919E+23, | -2.00, | 119.977, | | | |
| STOICH,COR | 7, | H2O2 | + M | | = OH | + OH | + M |
| FRATCO,COR | 7, | 0.1200E+18, | 0.00, | 45.500, | | | |
| RRATCO,COR | 7, | 0.3993E+12, | 1.00, | -6.505, | | | |
| STOICH,COR | 8, | H | + O2 | + M | = HO2 | + M | |
| FRATCO,COR | 8, | 0.1500E+16, | 0.00, | -1.000, | | | |
| RRATCO,COR | 8, | 0.6035E+19, | -1.00, | 47.352, | | | |
| STOICH,COR | 9, | H | + O2 | | = OH | + O | |
| FRATCO,COR | 9, | 0.4500E+11, | 1.00, | 14.805, | | | |
| RRATCO,COR | 9, | 0.2769E+10, | 1.00, | -1.860, | | | |
| STOICH,COR | 10, | H2 | + O | | = OH | + H | |
| FRATCO,COR | 10, | 0.1800E+11, | 1.00, | 8.900, | | | |
| RRATCO,COR | 10, | 0.8115E+10, | 1.00, | 6.871, | | | |
| STOICH,COR | 11, | OH | + OH | | = H2O | + O | |
| FRATCO,COR | 11, | 0.2140E+10, | 1.11, | 0.000, | | | |
| RRATCO,COR | 11, | 0.2322E+11, | 1.11, | 17.383, | | | |
| STOICH,COR | 12, | OH | + H2 | | = H2O | + H | |
| FRATCO,COR | 12, | 0.1100E+10, | 1.60, | 3.464, | | | |
| RRATCO,COR | 12, | 0.5382E+10, | 1.60, | 18.818, | | | |
| STOICH,COR | 13, | H | + HO2 | | = OH | + OH | |
| FRATCO,COR | 13, | 0.2500E+15, | 0.00, | 1.900, | | | |
| RRATCO,COR | 13, | 0.1628E+14, | 0.00, | 40.194, | | | |
| STOICH,COR | 14, | OH | + HO2 | | = H2O | + O2 | |
| FRATCO,COR | 14, | 0.5000E+14, | 0.00, | 1.000, | | | |
| RRATCO,COR | 14, | 0.5741E+15, | 0.00, | 73.343, | | | |
| STOICH,COR | 15, | O | + HO2 | | = OH | + O2 | |
| FRATCO,COR | 15, | 0.5000E+14, | 0.00, | 1.000, | | | |
| RRATCO,COR | 15, | 0.5290E+14, | 0.00, | 55.960, | | | |
| STOICH,COR | 16, | H | + HO2 | | = O2 | + H2 | |
| FRATCO,COR | 16, | 0.2500E+14, | 0.00, | 0.700, | | | |
| RRATCO,COR | 16, | 0.5867E+14, | 0.00, | 57.689, | | | |
| STOICH,COR | 17, | H2 | + O2 | | = OH | + OH | |

Table 1 continued

| | | | |
|----------------|-------------|-------|--------------|
| FRATCO,COR 17, | 0.2500E+13, | 0.00, | 39.000, |
| RRATCO,COR 17, | 0.6935E+11, | 0.00, | 20.305, |
| STOICH,COR 18, | H02 + H2 | | = H2O2 + H |
| FRATCO,COR 18, | 0.3000E+12, | 0.00, | 18.680, |
| RRATCO,COR 18, | 0.6216E+12, | 0.00, | 3.638, |
| STOICH,COR 19, | H02 + H02 | | = H2O2 + O2 |
| FRATCO,COR 19, | 0.2000E+13, | 0.00, | 0.000, |
| RRATCO,COR 19, | 0.9725E+13, | 0.00, | 41.947, |
| STOICH,COR 20, | H2O2 + H | | = H2O + OH |
| FRATCO,COR 20, | 0.3000E+15, | 0.00, | 9.000, |
| RRATCO,COR 20, | 0.4612E+14, | 0.00, | 77.690, |
| STOICH,COR 21, | H2O2 + OH | | = H2O + H02 |
| FRATCO,COR 21, | 0.1000E+14, | 0.00, | 1.000, |
| RRATCO,COR 21, | 0.2361E+14, | 0.00, | 31.396, |
| STOICH,COR 22, | CO + OH | | = CO2 + H |
| FRATCO,COR 22, | 0.1500E+08, | 1.30, | -0.765, |
| RRATCO,COR 22, | 0.3540E+10, | 1.30, | 23.304, |
| STOICH,COR 23, | CO + H02 | | = CO2 + OH |
| FRATCO,COR 23, | 0.1000E+12, | 0.00, | 10.000, |
| RRATCO,COR 23, | 0.1537E+13, | 0.00, | 72.364, |
| STOICH,COR 24, | CO + O2 | | = CO2 + O |
| FRATCO,COR 24, | 0.6905E+08, | 1.00, | 34.810, |
| RRATCO,COR 24, | 0.1003E+10, | 1.00, | 42.214, |
| STOICH,COR 25, | HCO + H | | = CO + H2 |
| FRATCO,COR 25, | 0.2000E+15, | 0.00, | 0.000, |
| RRATCO,COR 25, | 0.9053E+15, | 0.00, | 88.803, |
| STOICH,COR 26, | HCO + O | | = CO + OH |
| FRATCO,COR 26, | 0.5500E+13, | 0.50, | 0.000, |
| RRATCO,COR 26, | 0.1122E+14, | 0.50, | 86.774, |
| STOICH,COR 27, | HCO + O | | = CO2 + H |
| FRATCO,COR 27, | 0.5500E+13, | 0.50, | 0.000, |
| RRATCO,COR 27, | 0.2649E+16, | 0.50, | 110.843, |
| STOICH,COR 28, | HCO + OH | | = CO + H2O |
| FRATCO,COR 28, | 0.3000E+11, | 1.00, | 0.000, |
| RRATCO,COR 28, | 0.6644E+12, | 1.00, | 104.157, |
| STOICH,COR 29, | HCO + O2 | | = CO + H02 |
| FRATCO,COR 29, | 0.5000E+12, | 0.50, | 0.834, |
| RRATCO,COR 29, | 0.9644E+12, | 0.50, | 32.648, |
| STOICH,COR 30, | HCO + HCO | | = CH2O + CO |
| FRATCO,COR 30, | 0.4000E+14, | 0.00, | 0.000, |
| RRATCO,COR 30, | 0.3290E+16, | 0.00, | 75.034, |
| STOICH,COR 31, | CH2O + H | | = HCO + H2 |
| FRATCO,COR 31, | 0.2500E+11, | 1.00, | 3.200, |
| RRATCO,COR 31, | 0.1376E+10, | 1.00, | 16.969, |
| STOICH,COR 32, | CH2O + OH | | = HCO + H2O |
| FRATCO,COR 32, | 0.5000E+10, | 1.00, | 1.650, |
| RRATCO,COR 32, | 0.1346E+10, | 1.00, | 30.773, |
| STOICH,COR 33, | CH2O + O | | = HCO + OH |
| FRATCO,COR 33, | 0.1300E+11, | 1.00, | 2.050, |
| RRATCO,COR 33, | 0.3225E+09, | 1.00, | 13.790, |
| STOICH,COR 34, | CH2O + H02 | | = HCO + H2O2 |
| FRATCO,COR 34, | 0.1000E+13, | 0.00, | 8.000, |
| RRATCO,COR 34, | 0.1140E+12, | 0.00, | 6.727, |
| DILUEN,HE | | | |

Results

An objective of the model was to determine at the outset if and under which conditions it is possible to obtain a laminar opposed jet diffusion flame with a reaction zone of sufficient thickness to allow for detailed probing for species profiles. The conditions examined correspond to fairly low stretching rates because of limitations in gas flow rates for the experiment proposed. The question addressed is whether it is possible to obtain a sufficient number of sampled points to properly define gradients, and even second derivatives in the reaction zone. This question always arises when flat flame data is utilized to extract net rates of formation or destruction of a particular species.

Figure 3 through 5 show profiles of temperature, velocity and the similarity function ψ , of major and of intermediate species for a laminar opposed jet diffusion flame with $\text{CO}/\text{H}_2\text{O}/\text{N}_2$ as the fuel and O_2/N_2 as the oxidant. The stretching rate was 3.62 sec^{-1} . Note how density changes influence the axial velocity profile and the location of the stagnation place (Figure 3). The flame thickness is predicted to be approximately 1 cm, which is barely sufficient to allow for detailed in flame probing. Figure 5 shows that the free radical profiles are present in only a similarly very narrow region.

These results prompted further predictions to be made to determine how the reaction zone thickness might be broadened. One approach is to operate the flame at subatmospheric pressure. Figure 6 show the effect of decreasing pressure on the reaction zone thickness of the same flame at the same stretching rate of $\epsilon = 3.62 \text{ s}^{-1}$.

It is apparent that the reaction zone thickness is not greatly expanded until the pressure is reduced to 0.2 atmospheres or below. However, sampling

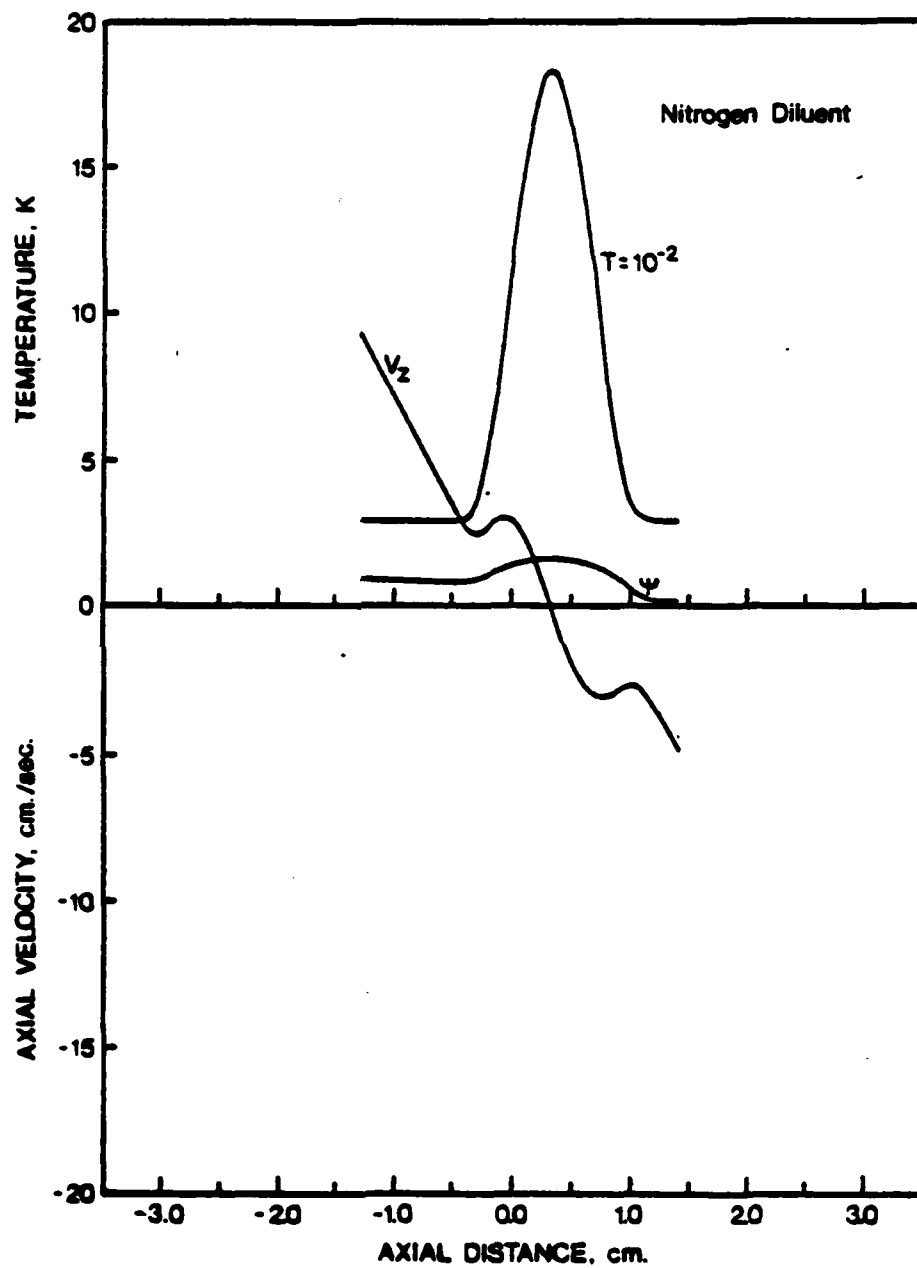


Figure 3. Velocity, Temperature and ψ Profiles for LOJDF with N_2 as Diluent and $\epsilon = 3.62 \text{ s}^{-1}$.

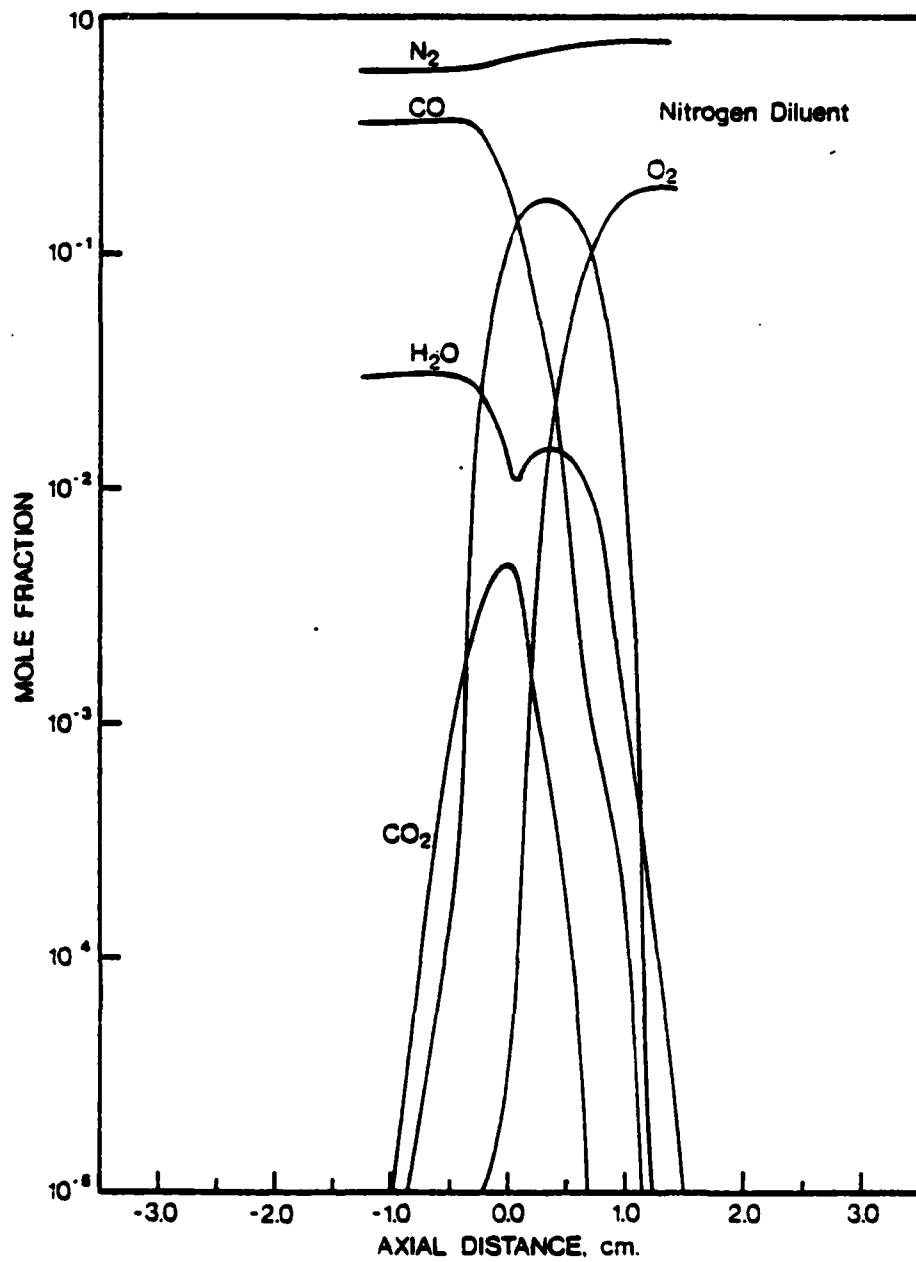


Figure 4. Major Species Profiles ($\epsilon = 3.62 \text{ s}^{-1}$).

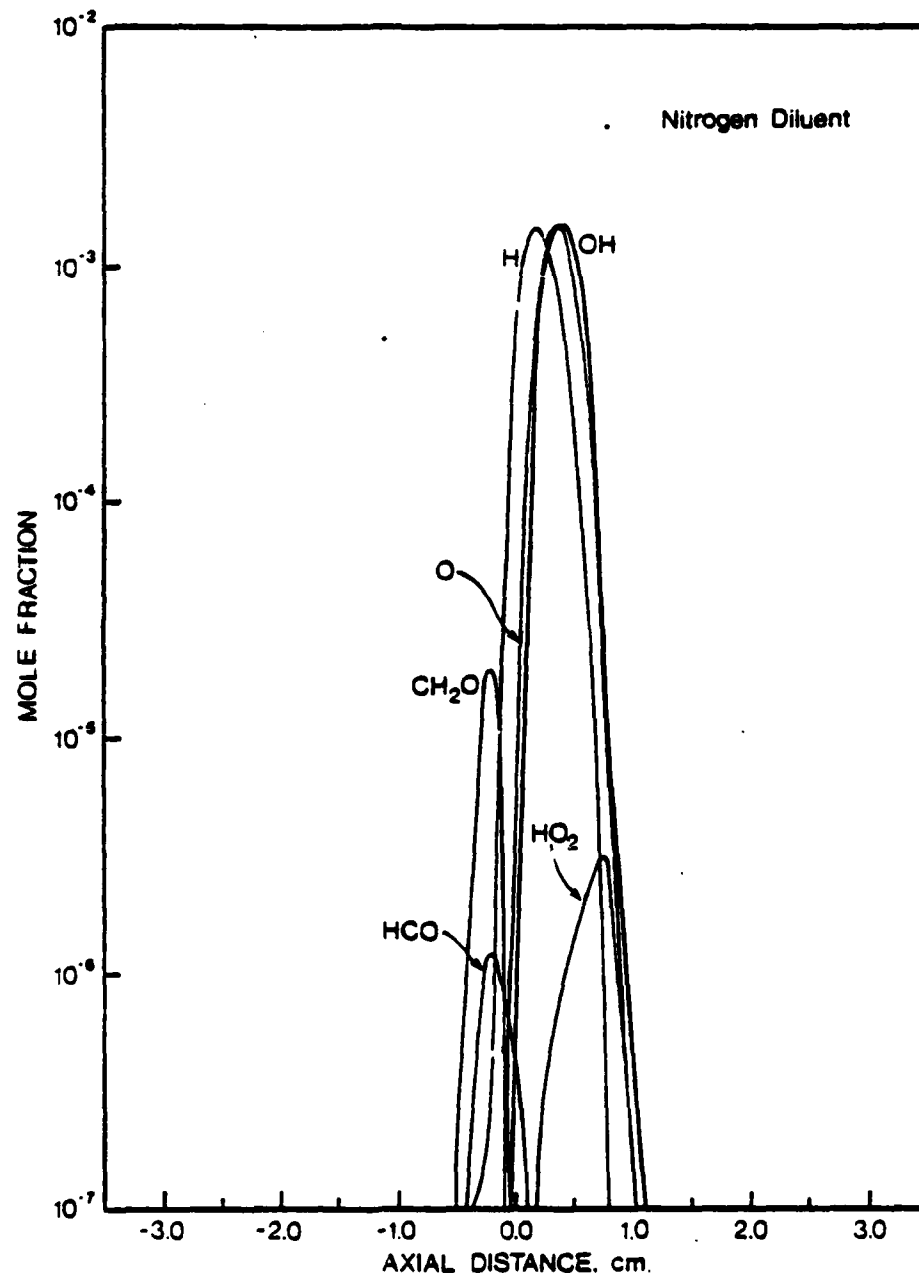


Figure 5. Free Radical Species Profiles ($\epsilon = 3.62 \text{ s}^{-1}$).

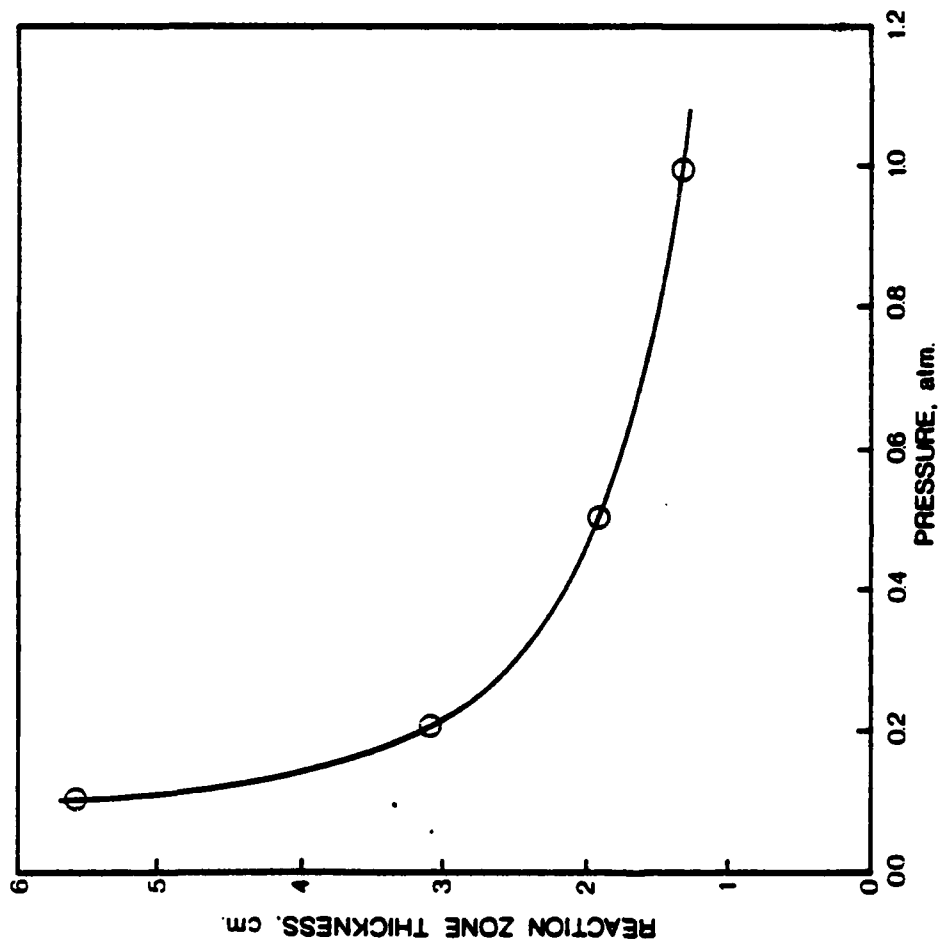


Figure 6. Influence of Pressure on Reaction Zone Thickness.
(N_2 Diluent, $\epsilon = 3.62 \text{ s}^{-1}$)

and analysis problems are more difficult at these low pressures. Therefore another approach to achieve thick reaction zones was attempted.

Because helium has a large diffusion coefficient it was thought that substitution of nitrogen by helium as a diluent should achieve thicker reaction zones. Figure 7 through 9 show results of predictions of a CO/H₂/O₂ diffusion flame in helium. Figure 7 shows a reaction zone thickness of over 2.5 cm, while Figures 8 and 9 indicate that at a stretching rate $\epsilon = 3.62 \text{ s}^{-1}$, the species profiles do change over a sufficiently thick zone to allow proper sampling, even at one atmosphere.

Conclusions

The laminar opposed jet diffusion flame can be modeled in detail and used to corroborate and investigate combustion kinetic mechanisms under a wide range of conditions. The ensuing equations to be solved involve stiffness through the reaction rate terms, very steep spatial gradients because of diffusion, and a velocity which changes sign in the domain of interest. These equations can best be solved numerically in a fully coupled, fully implicit manner, utilizing a modified discretization in the spatial domain that approximates the equation within a mesh interval rather than its solution. This approach may be relevant to many other problems involving multiple reactions, diffusion and convection, and future work might be directed towards improving the discretization to allow better approximations for the forcing function within a mesh interval.

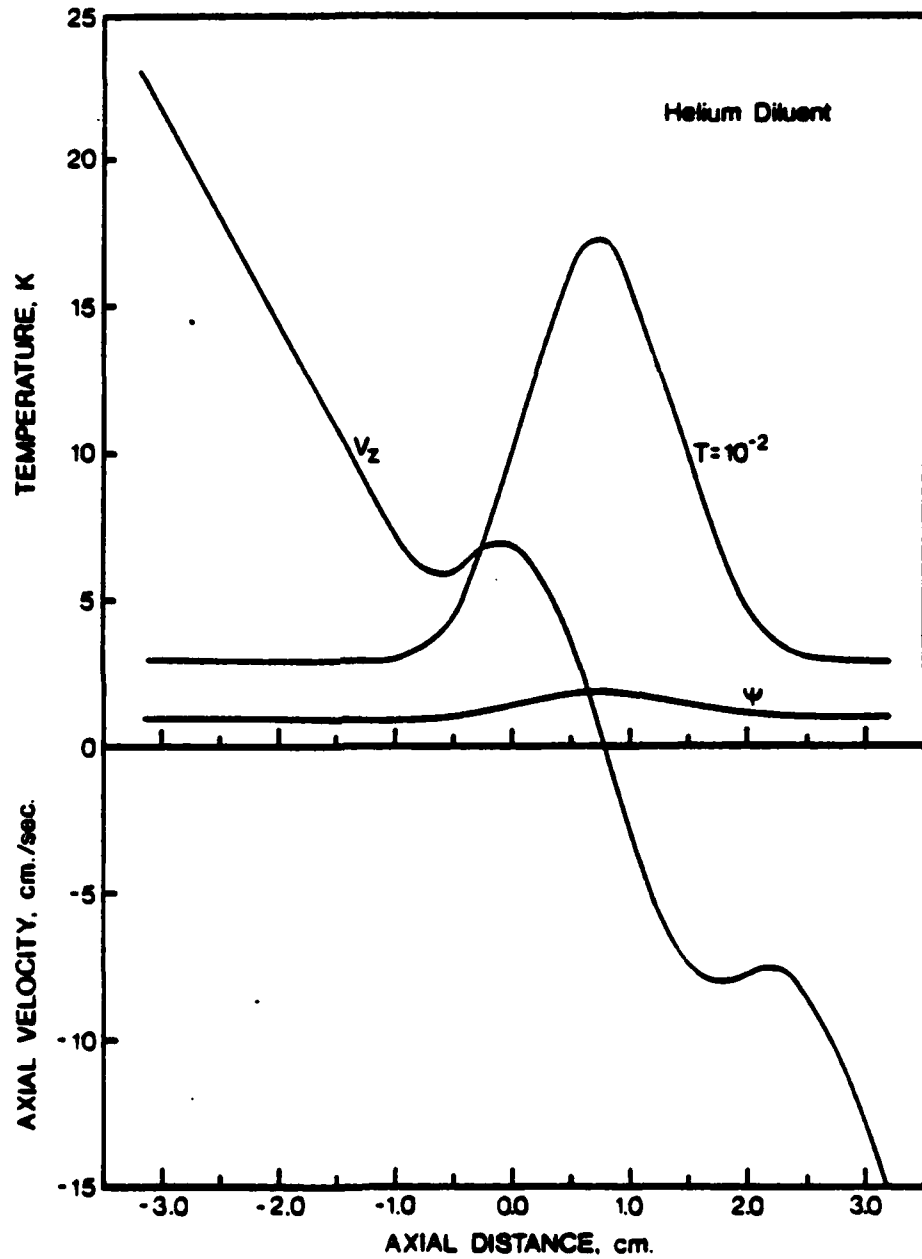


Figure 7. Velocity, Temperature and ψ Profiles for LOJDF with Helium Diluent ($\epsilon = 3.62 \text{ sec}^{-1}$).

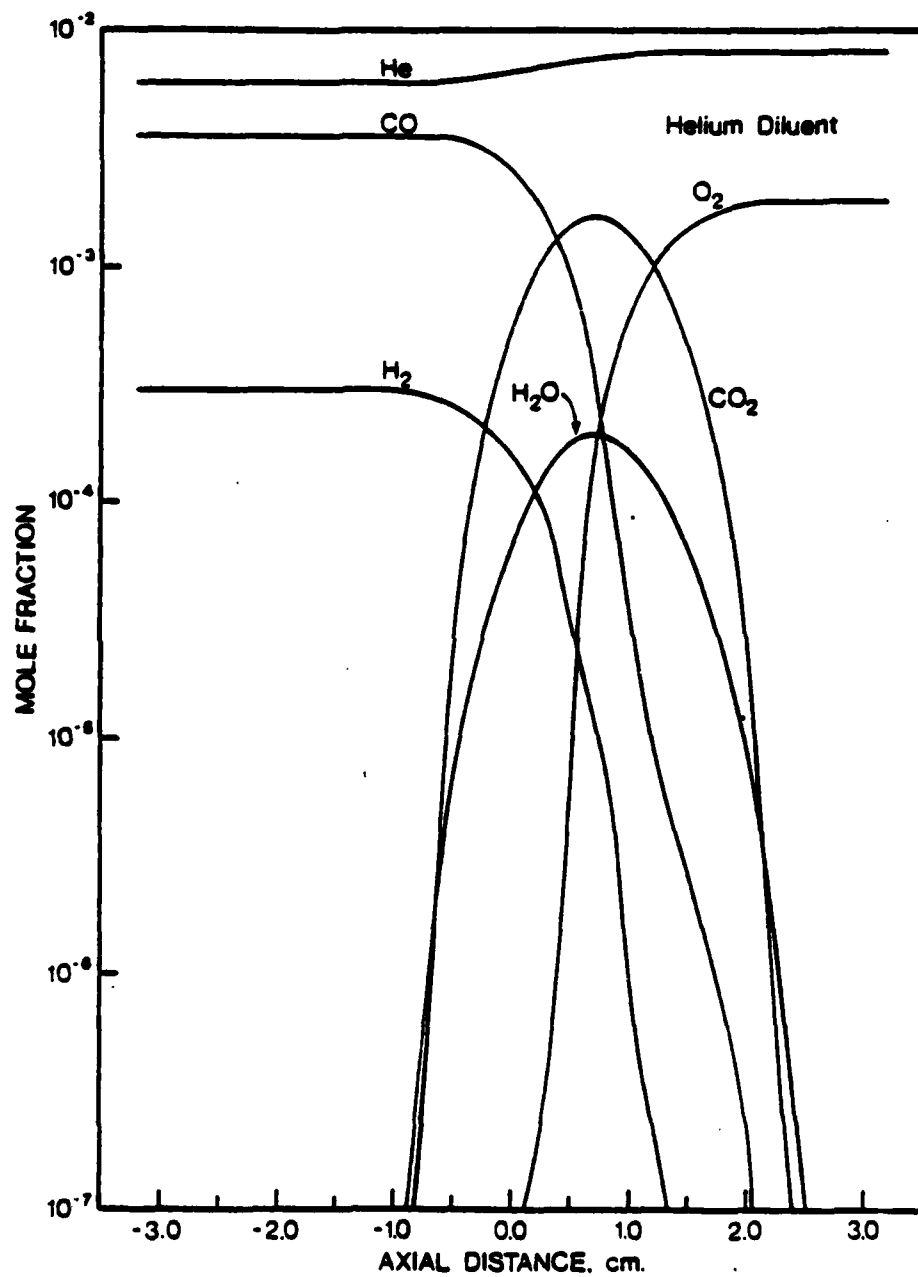


Figure 8. Major Species Profiles: Helium Diluent.

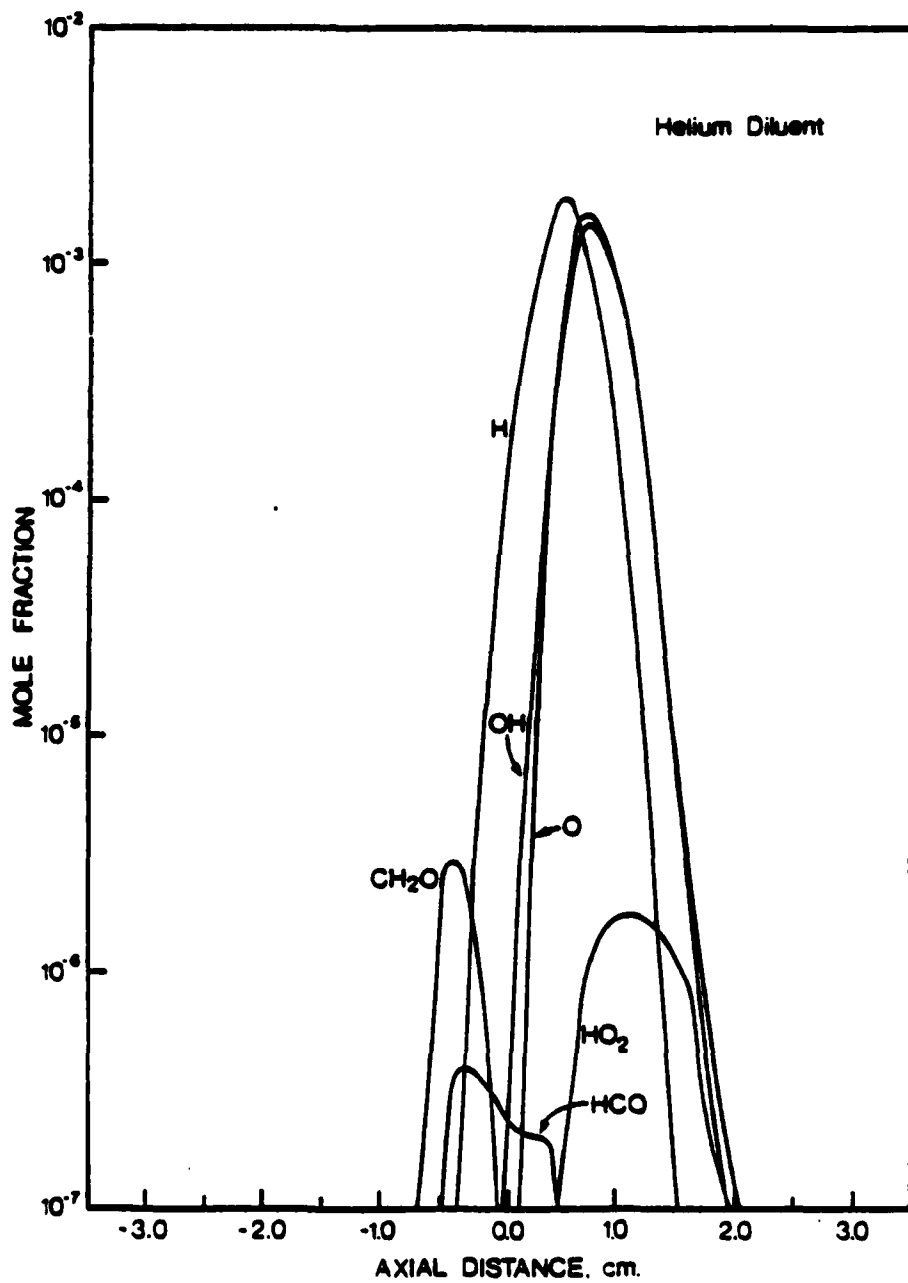


Figure 9. Free Radical Species Profiles: Helium Diluent.

NOMENCLATURE

| | |
|-------------|--|
| A | Block tridiagonal matrix. |
| A_i | Variable coefficient in equation (19). |
| B_i | Variable coefficient in equation (19). |
| c | Molar density (gr-mol/cm ³). |
| C_i | Genrtal dependent variable. |
| C_p | Heat capacity (cal/gr-mol ^o K). |
| D_A | Pseudo binary diffusion coefficient of species A in gaseous mixture. |
| D_i | Variable coefficient in equation (19). |
| $F_{i,p}^k$ | Forcing function evaluated at grid p and time k. (see equation 26). |
| F_{-1} | Block vector of forcing functions (see equation (31)). |
| h | Mesh size for modified discretization (see Appendix). |
| k | Thermal conductivity of gaseous mixture (cal/cm-sec ^o K) Length of the domain in which numerical solution is sought. |
| m | Number of grid points. |
| MW | Molecular weight of mixture. |
| n | Number of gases in mixture. |
| nr | Number of chemical reactions. |
| NS | Number of species in rection set. |
| N | $nsp + 2$. |
| P | Pressure (atm). |
| r | Radial distance (cm). |
| r_j | Rate of reaction j (gr-mol/cm ³ -sec). |
| R_A | Rate of formation of species A (gr-mol/cm ³ -sec). |

| | |
|-----------------|--|
| R_I | Forcing function (see equation (30)). |
| t | Time (sec). |
| T | Temperature ($^{\circ}\text{K}$). |
| v | Axial velocity (cm/sec). |
| \underline{v} | Velocity vector (cm/sec). |
| v_r | Radial component of \underline{v} (cm/sec). |
| v_z | Axial component of \underline{v} (cm/sec). |
| $W_{1,p+1}$ | Weighting factor for modified discretization (see equation 32 and Appendix). |
| $W_{1,p-1}$ | Weighting factor for modified discretization (see equation (32) and Appendix). |
| $W_{2,p+1}$ | Weighting factor for modified discretization (see equation (33) and Appendix). |
| $W_{2,p-1}$ | Weighting factor for modified discretization (see equation (33) and Appendix). |
| x_A | Mole fraction of species A. |
| z | Axial distance (cm). |

Greek Symbols

| | |
|------------------------|---|
| $\alpha_{i,p}$ | Auxiliary variable (see equation 27)). |
| $\beta_{i,p}$ | Auxiliary variable (see equation 28)). |
| $\gamma_{i,p}$ | Auxiliary variable (see equation 29)). |
| $\underline{\Delta C}$ | Block vector of corrections. |
| $\Delta C_{j,p}$ | Correction to variable j at point p . |
| Δh_j | Heat of reaction j (cal/gr-mol). |
| Δz | Increment of grid. |
| c | Stretching rate (sec^{-1}). |

- θ Dimensionless temperature = $\frac{T-298}{298}$.
- μ Viscosity of mixture (g/cm-sec).
- ρ Density (g/cm³).
- ψ Similarity transformation (see equation (12)).

Subscripts

- Refers to distance from flame zone.
- i Refers to dependent variable i.
- p Refers to grid point.

Superscripts

- k Refers to time step.

REFERENCES

- Corley, T.L. and C.T. Bowman in Pulverized Coal Combustion, Pollutant Formation and Control, EPA Project Decade Monograph, Chapter 5, to be published by US Government Printing Office, 1982.
- Fendell, F.E., "Ignition and Extinction in Combustion of Initially Unmixed Reactants", *J. Fluid Mech.*, 21, 281 (1965).
- Field, M.A., D.W. Gill, B.B. Morgan, and P.G.W. Hawksley, "Combustion of Pulverized Coal", BCURA Leatherhead, Cherey & Sons Ltd., Bamberg, England (1967).
- Hahn, W.A., "Pollutant Formation of Flat Laminar Opposed Jet Diffusion Flames", Ph.D. Dissertation, University of Arizona, Tucson. University Microfilms, Ann Arbor, Michigan (1979).
- Krishnamurthy, L., and F.A. Williams, "A Flame Sheet in the Stagnation-Point Boundary Layer of a Condensed Fuel", *Acta Astronautica*, 1, 711 (1974).
- Peaceman, D.W., "Fundamentals of Numerical Reservoir Simulation", Elsevier Scientific Publishing Company, New York (1977).
- Peters, N., "An Asymptotic Analysis of Nitric Oxide Formation in Turbulent Diffusion Flames", *Comb. Sci. and Tech.* 19, 39 (1978).
- Tyson, T.J., "An Implicit Integration Method for Chemical Kinetics", Paper 9840-6002-R000, TRW Space Tech. Lab., Redondo Beach, California (1964).
- Wendt, J.O.L., C.H. Martinez, D.G. Lilley, and T.L. Corley, "Numerical Solution of Stiff Boundary Valued Problems in Kinetics and Diffusion", *Chem. Eng. Sci.*, 34, 527 (1979).

ACKNOWLEDGEMENTS

The authors would like to thank W. Steven Lanier for his suggestion to attempt the modified discretization described in this work. Additional insight contributed by J. Heinrich and C.J. Kau is also much appreciated. This work was supported by the U.S. Environmental Agency under Grant R806685-01 to the University of Arizona.

APPENDIX

Development of Modified Discretization

If the function to be obtained is viewed to approximate the solution of second order ODE with constant coefficients:

$$a \frac{d^2 C}{dz^2} + b \frac{dC}{dz} + C = f \quad (A1)$$

then its form will be

$$C = A_1 e^{\alpha_1 z} + A_2 e^{\alpha_2 z} + f \quad (A2)$$

where α_1 and α_2 may be real and distinct or complex.

From Equation (A2)

$$C_{p+1} = A_1 e^{\alpha_1 z} e^{\alpha_1 h} + A_2 e^{\alpha_2 z} e^{\alpha_2 h} + f \quad (A3)$$

$$C_{p-1} = A_1 e^{\alpha_1 z} e^{-\alpha_1 h} + A_2 e^{\alpha_2 z} e^{-\alpha_2 h} + f \quad (A4)$$

Expanding C forward and backward in a Taylor Series, evaluating all derivatives and not truncating achieves the following results:

$$\frac{dC}{dx} = \frac{C_{p+1} - C_{p-1}}{2h} - \frac{1}{h} A_1 e^{\alpha_1 z} [\sinh(\alpha_1 h) - \alpha_1 h] - \frac{1}{h} A_2 e^{\alpha_2 z} [\sinh(\alpha_2 h) - \alpha_2 h] \quad (A5)$$

and

$$\frac{d^2 C}{dz^2} = \frac{C_{p+1} - 2C_p + C_{p-1}}{h^2} - \frac{2}{h^2} A_1 e^{\alpha_1 z} \left[\cosh(\alpha_1 h) - 1 - \frac{\alpha_1^2 h^2}{2} \right] - \frac{2}{h^2} A_2 e^{\alpha_2 z} \left[\cosh(\alpha_2 h) - 1 - \frac{\alpha_2^2 h^2}{2} \right] \quad (A6)$$

Inserting (A3) and (A4) yields

$$A_1 e^{\alpha_1 z} = \frac{e^{-\alpha_2 h} C_{p+1} - e^{\alpha_2 h} C_{p-1}}{2 \sinh[(\alpha_1 - \alpha_2) h]} \quad (A7)$$

$$A_2 e^{\alpha_2 z} = \frac{e^{-\alpha_1 h} C_{p+1} - e^{\alpha_1 h} C_{p-1}}{2 \sinh[(\alpha_2 - \alpha_1) h]} \quad (A8)$$

which, on substitution in the RHS of A5 and A6 and gathering terms, yields expressions for the weighting functions depending on whether the roots are real and distinct, complex, or real and equal.

a) Real distinct roots α_1, α_2 .

$$W_{1,p+1} = 1 - \frac{e^{-\alpha_2 h} [\sinh(\alpha_1 h) - \alpha_1 h]}{\sinh[(\alpha_1 - \alpha_2) h]} - \frac{e^{-\alpha_1 h} [\sinh(\alpha_2 h) - \alpha_2 h]}{\sinh[(\alpha_2 - \alpha_1) h]} \quad (A9)$$

$$W_{1,p-1} = 1 - \frac{e^{\alpha_2 h} [\sinh(\alpha_1 h) - \alpha_1 h]}{\sinh[(\alpha_1 - \alpha_2) h]} - \frac{e^{\alpha_1 h} [\sinh(\alpha_2 h) - \alpha_2 h]}{\sinh[(\alpha_2 - \alpha_1) h]} \quad (A10)$$

$$W_{2,p+1} = 1 - \frac{e^{-\alpha_2 h} \left[\cosh(\alpha_1 h) - 1 - \frac{\alpha_1^2 h^2}{2} \right]}{\sinh[(\alpha_1 - \alpha_2) h]} - \frac{e^{-\alpha_1 h} \left[\cosh(\alpha_2 h) - 1 - \frac{\alpha_2^2 h^2}{2} \right]}{\sinh[(\alpha_2 - \alpha_1) h]} \quad (A11)$$

$$W_{2,p-1} = 1 + \frac{e^{\alpha_2 h} [\cosh(\alpha_1 h) - 1 - \frac{\alpha_1^2 h^2}{2}]}{\sinh[(\alpha_1 - \alpha_2) h]} + \frac{e^{\alpha_1 h} [\cosh(\alpha_2 h) - 1 - \frac{\alpha_2^2 h^2}{2}]}{\sinh[(\alpha_2 - \alpha_1) h]} \quad (\text{A12})$$

b) Complex roots:

$$\alpha_1 = \alpha_R - i \alpha_I \quad (\text{A13})$$

$$\alpha_2 = \alpha_R - i \alpha_I \quad (\text{A14})$$

The weighting functions are:

$$W_{1,p+1} = 1 - \frac{2e^{-\alpha_R h} (M_1 + M_2)}{\sin(2\alpha_I h)} \quad (\text{A15})$$

$$W_{1,p-1} = 1 + \frac{2e^{\alpha_R h} (M_1 - M_2)}{\sin(2\alpha_I h)} \quad (\text{A16})$$

and

$$M_1 = \cos(\alpha_I h) [\alpha_I h - \cosh(\alpha_R h) \sin(\alpha_I h)] \quad (\text{A17})$$

$$M_2 = \sin(\alpha_I h) [\alpha_R h - \sinh(\alpha_R h) \cos(\alpha_I h)] \quad (\text{A18})$$

While

$$W_{2,p+1} = 1 + \frac{2e^{-\alpha_R h} (N_1 - N_2)}{\sin(2\alpha_I h)} \quad (\text{A19})$$

$$W_{2,p-1} = 1 - \frac{2e^{\alpha_R h} (N_1 + N_2)}{\sin(2\alpha_I h)} \quad (\text{A20})$$

and

$$N_1 = \cos(\alpha_I h) [\alpha_R \alpha_I h^2 - \sinh(\alpha_R h) \sin(\alpha_I h)] \quad (\text{A21})$$

$$N_2 = \sin(\alpha_I h) [\cosh(\alpha_R h) \cos(\alpha_I h) - 2 - \alpha_R^2 h^2 + \alpha_I^2 h^2] \quad (\text{A22})$$

c) Equal roots, α

The solution now is

$$C = A_1 e^{\alpha z} + A_2 z e^{\alpha z} + f \quad (\text{A23})$$

and the weighting functions for Equations (32) and (33) become

$$W_{1,p+1} = 1 - e^{-\alpha h} [\sinh(\alpha h) - \cosh(\alpha h) - \alpha h - 1] \quad (\text{A24})$$

$$W_{1,p-1} = 1 + e^{\alpha h} [\sinh(\alpha h) - \cosh(\alpha h) - \alpha h + 1] \quad (\text{A25})$$

$$W_{2,p+1} = 1 - e^{-\alpha h} [\cosh(\alpha h) + \sinh(\alpha h) - \frac{\alpha^2 h^2}{2} - \alpha h - 1] \quad (\text{A26})$$

$$W_{1,p-1} = 1 - e^{\alpha h} [\cosh(\alpha h) - \sinh(\alpha h) - \frac{\alpha^2 h^2}{2} + \alpha h - 1] \quad (\text{A27})$$

Thus it can be seen that this modified finite difference formulation is exact, with no truncation error, if the general solution for the real problem behaves like that for linear ODE's locally. The roots α_1 and α_2 are calculated from the actual homogeneous equation at position P. The particular integral is also taken to be a constant locally. Since the equations for the opposed jet diffusion flame might fit into this category (compare Equations A1 and 19) it

would appear that the new formulation might allow a drastic reduction in the number of grid points.

Simple Test for Comparison of Standard and Modified Discretization

For the solution of simple ODE's with constant coefficients, with 11 grid points modified discretization was exact to with seven figures while standard discretization was good to with 2 to 4 figures. For the equation:

$$0.5 \frac{d^2 C}{dz^2} + 20z = - 10^6 e^{-\frac{z^2}{0.08}}$$

with

$$\begin{array}{ll} \text{BC: } z = - 1.5 & C = 1.0 \\ z = + 1.5 & C = 0.0 \end{array}$$

the modified discretization was stable using from 5 to 95 grid points. The standard discretization was unstable, with oscillations at the boundaries for less than 35 grid points and thus with total loss of accuracy at the boundary. However, the modified discretization, although stable at the boundaries, was inferior to standard discretization in determining the peak value of C. This was doubtless due to the assumption of a constant forcing function over a mesh interval. Further improvement might be achieved by allowing for a exponential forcing function over a mesh interval and thus deriving a new formulation for discretization.

A Master Equation Study of the Rate and
Mechanism of Vibrational Relaxation and
Dissociation of Molecular Hydrogen by Helium

John E. Dove

Department of Chemistry and Scarborough College,
Lash Miller Chemical Laboratories, University
of Toronto, Toronto, Ontario, Canada M5S 1A1

and Susanne Raynor

Department of Chemistry, Harvard University
12 Oxford Street, Cambridge, Mass. 02138, U.S.A.

Abstract

In order to gain a full understanding of the basis of chemical reactivity, it is necessary to know how individual molecular processes influence or determine the rates of observed chemical reactions. A widely applicable model of chemical kinetic processes in molecular gases is to treat them as a sequence of collision-induced transitions between the quantized energy levels of the molecules. Using this model, and assuming that the rate constants for the transitions between levels are known or can be found, the prediction of the kinetic behavior requires the solution of a set of differential equations. The time-dependent populations of the individual energy levels of the molecules can be calculated by integrating the system of chemical kinetic ordinary differential equations ("master equation") governing those populations. When the populations are known, the other time-dependent properties of the reaction system can be calculated, and the influence of individual molecular processes can be analyzed.

In recent theoretical studies, the H_2 molecule has played an important role. A major reason for this is that H_2 has only about 370 internal (v, J) levels, a number which is small enough for the master equation to be solved directly by modern numerical techniques. The size of the problem can be further reduced by confining attention to, e.g., the 176 levels of para- H_2 .

Two of the most basic chemical kinetic processes are the collision-induced vibrational excitation, and the collision-induced dissociation, of simple molecules. We have studied these processes in para- H_2 with He as collision partner. The H_2 -He system was chosen because very good interaction potentials are available. The calculations simulated shock wave experiments in which the translational temperature of the gas is suddenly raised; the populations of the levels then relax towards equilibrium. The system of differential equations which must be solved is very stiff. Transition rate constants among the various bound levels can differ by fifteen orders of magnitude, and some of the rate constants for the dissociative processes are even smaller than any of those for transitions among the bound levels. Except at high temperatures, the most important processes are those in which the quantum numbers v and J change by only small amounts. The rate constant matrix can therefore be very sparse.

Examples of these calculations will be presented and discussed. Among the factors which have been investigated and elucidated are: the mechanism of vibrational relaxation, especially the relative importance of pure vibrational and simultaneous vibration-rotation transitions; the relative importance of rotational and vibrational energy in the dissociation reaction; the contribution of collision-induced dissociation directly out of relatively low-lying energy levels; departures from a Boltzmann distribution among molecular energy levels during pseudo-steady dissociation; the factors governing the observed "anomalously" low temperature dependence of the dissociation rate; the possibility of deviations from a linear mixture rate law for the overall dissociation reaction.

DYNAMICS OF FIXED BED ADSORBERS

M.Q.Dias, J.C.Lopes, C.A.Costa and A.E.Rodrigues*

Modelling of fixed bed adsorption is described by two different approaches: staged and differential.

Considering first the staged approach (column viewed as a series of perfectly mixed sorbers) we start with the study of adsorption in a CSTR. Even for this simple case highly stiff systems occur, with film mass transfer resistance, and provided the isotherm is nonlinear.

Algorithms for the integration of systems of ODEs (Michelsen, Gearb) are compared.

The differential approach leads to a system of PDEs; after using collocation methods we obtain again stiff systems. Difficulties associated with handling this problem is discussed in detail.

References:

Rodrigues et al, AICHEJ, 25(3), 416 (1979)

Rodrigues et al, CACE'79, Montreux

*Department of Chemical Engineering
University of Porto
PORTO Codex, Portugal

STIFF COMPUTATION: WHERE TO GO?

Informal Communication Prepared for the
International Conference on Stiff Computation
to be held: April 12-14, 1982 at Park City, Utah

by: Dr. Francis E. Collier
Lecturer in Simulation Techniques
Institute for Automatic Control
The Swiss Federal Institute of Technology Zurich
ETH - Zentrum
CH-8092 Zurich
Switzerland

It is the aim of this communication to list some of the more severe shortcomings of currently available stiff computation codes as they have been collected through discussions with the "end users" of such software. These complaints are, however, formulated in terms as they are understandable to mathematicians, that is to the "producers" of integration algorithms. Some suggestions are given on how these deficiencies may be overcome in the future, and where open research fields can still be found.

1. INTRODUCTION

Having a background in Electrical Engineering, stiff computation is certainly not at the very center of my scientific research activities. I have never developed an integration algorithm of my own, and, if my prediction is correct, I shall never do so.

Having read this statement of mine, you may ask yourself what then is the reason for me to sit down at the text editor and to type in this communication. In fact, you may be reminded of the old joke concerning the fellow who once entered the employment office of a larger company. Asked for his desire, he answered that he had come because he had read in the newspaper about this new opening for the post of an engineer. The person in charge of the matter started to ask him questions of all sorts to figure out what level of knowledge this fellow had acquired, and it turned out soon that he was hardly able to answer the simplest questions. Finally, the employment manager turned sad and asked him what had made him thinking that he might be the appropriate person for this job. So, the (rather shy) visitor told him gently that he just had come to tell them that they should not count on him. However, I insist that also this is not the reason for my writing.

Originally coming from the application's side (developing simulation software for the Swiss machine tool industry), I moved more and more away from the "end user" applications, and became meanwhile a "tool maker", that is: a person who develops software (in particular: modelling and simulation software) for other persons' needs. My own field of research these days is basically in Computer Science. By these means, I understand my job as to be kind of an "interpreter" between the

"end users" of the software on the one side, that is: people who have definite needs but who are poorly trained to express their needs in terms condensed enough to make them easily understandable to mathematicians; and between the "producers" of algorithms (among others: integration algorithms) on the other side.

In this communication, I shall try to summarize a few complaints raised by some of my users as concerning stiff computation algorithms and codes. However, I shall try to reformulate these complaints in terms which should be understandable to the producers of these algorithms which I realize shall be present at Park City. That is: my communication shall basically pose questions rather than to present answers. It is my sincere hope that one or the other of the "algorithm producers" may find these questions of sufficient interest and generality to be prepared to looking for answers, that is: to develop new algorithms or amend existing algorithms which I could then integrate into my simulation software so that, in the final run, they would satisfy the needs of my simulation users.

2. DISCONTINUITY HANDLING

It happens frequently in engineering applications that simulation problems are of a discontinuous nature. In terms of a mathematician: given the state space representation of a system (that is: a set of first order ODE's)

$$\dot{x} = f(x, t)$$

together with a set of initial conditions:

$$x(t_0) = x_0$$

to form an initial value problem, and together with a final time

t_f .

Some functions f_i are discontinuous functions in time.

Typical examples of discontinuous elements are:

- a) in mechanical engineering:
 - friction phenomena
 - losses
- b) in electrical engineering:
 - diodes
 - thyristors
 - any combined analog and digital circuitry
- c) in chemical engineering:
 - charging and discharging of batch reactors

We can distinguish basically between two different types of discontinuities:

- a) discontinuities of which we know the time when they are expected to happen. As a typical example of this type of discontinuity, we may mention the output voltage of a square wave generator. Each time, the voltage switches from negative to positive value or vice-versa, a discontinuity takes place. The time instances, when these discontinuities are expected to take place, are precisely known beforehand. These discontinuities are in simulation usually referred to as "time events". Time events are scheduled events in the sense that the time instances of when the discontinuity is to take place may be collected into a "calendar of events".
- b) discontinuities of which we do not beforehand know the time when they are expected to happen. Instead, we know the condition under which the discontinuity takes place, e.g. one state variable (x_i) crossing a prescribed level into positive direction. As a typical example of this type of discontinuity, we may mention the output voltage of a rectifier circuit. Here, we do not know when the output voltage is going to have a discontinuity (in its first time derivative). We just know that, each time the input voltage crosses through zero in either direction, the output voltage has a break. These discontinuities are in simulation usually referred to as "state events". For state events, no scheduling is possible. Instead, we need a mechanism to describe the condition under which the event is going to happen. This is usually referred to as "state condition".

It makes sense to distinguish between these two types of discontinuities for two reasons:

- 1) The user of the code requires two different mechanisms to describe the discontinuities to the program.
- 2) The algorithm requires two different

mechanisms to handle them. Time events may be handled simply by inspection of the calendar of events for the next scheduled event time. When this moment approaches, the step size must be reduced to hit the event time precisely, or the algorithm must interpolate back to the just passed event time, depending on the algorithm in use. State events, on the other hand, must be handled by either iteration or interpolation (depending on the algorithm).

Unfortunately, until very recently, none of the existing codes for numerical integration provided for appropriate discontinuity handling. When I mentioned this problem to Alan Hindmarsh during the Urbana ACM/SIGNUM Conference on Numerical Ordinary Differential Equations (three years ago), he answered that this was merely a trivial problem which should be left to the end user (to be optimally adapted to his personal needs) and should certainly not be of any concern to the numerical mathematician. I did not agree upon this judgment at that time, and I definitely do not agree to it now. Reasons for this are manifold:

- 1) The numerical mathematician usually thinks that, given a particular algorithm, everything is said and clear. However, this is not the case in reality for reasons of communication problems. The end user is, in general, not able to understand an integration algorithm sufficiently well to be able to code it into a properly executing program by himself. This holds equally true for discontinuity handling. Concerning the integration algorithm, the mathematical attitude was originally also that coding of the program would be basically in the user's responsibility, while it has meanwhile been realized that pre- or standardized codes are superior in many respects, even though there are still some people around to call this the "vacuum cleaner" approach (Gene Golub). I suggest that the handling of discontinuities is just a little junger problem, and that, on the long run, also here standardized codes shall replace home tailored software.
- 2) If you look into the work done by T. E. Hull at Toronto, it should become clear that one has no chance whatsoever to compare integration algorithms with each other. What one can do, however, is to compare the aptitude of different integration codes to solve a particular given application problem. Hopefully, one is lucky enough to be able to extend the experiences gained from entire sets of problems to a classification of applications. It should then become clear which algorithm is best for any particular practical application without being forced to try them all out. Hull's results indicate that the step from the integration algorithm once given to the integration code is by no means of a trivial nature. In fact, more computation time is used by the "dirty" surroundings than by the integration algorithm itself. Again, also this applies to the problem of discontinuity

handling as well.

- 3) It is not economical at all to develop larger pieces of software for each application separately. On the contrary, we should concentrate on determining what large classes of application problems have in common, extract this information, and provide for a program which handles all that in a standardized manner. This is time efficient, money efficient, and much safer as the amount of software left to be coded by the end user is minimized while obviously much more care and expertise can flow into the design and production of the standardized part of the software. It is here, where "simulation software" comes into the game. Definitely, discontinuity handling belongs to the part of the code which should be standardized.

To conclude these considerations, let me propose some very concrete steps towards a (according to my view satisfactory) solution. Those among you, who are involved in the business of producing integration codes rather than integration algorithms, shall certainly be aware of the proposals raised by Alan Hindmarsh as concerning a standardized user interface for ODE solvers (ODEPACK) [14]. This is precisely where we ought to go. It is evident that I cannot know in advance under all circumstances which integration code shall be optimally suited for my particular application. It should, thus, be such that I can code my application software entirely independent of the ODE solver. Ideally seen, I would like to be able to replace any ODE solver by any other ODE solver by simply replacing one subroutine name by another subroutine name without being forced to change a single bit of my application program beyond. In fact, I would want to maintain a library of integration codes to have a "remedy against all diseases". An additional advantage of this approach is that I may easily update my ODE solver library when new releases become available as there are no side effects to be expected from simply replacing the old subroutine by its modified version. However, this also means that I should not be forced to change a single line within the ODE solver for any particular application. It is now very easily shown that, when I try to graft the discontinuity handling upon the integration code (which is possible, and which I have done in my simulation software GASP-V [4,5] for precisely the above mentioned reasons), I obtain inefficient code. For this reason, although I fully support Alan Hindmarsh's idea concerning a standardized interface for ODE solvers, I do not agree to his standard proposal. I feel strongly about the need for enhancing the standard proposal by adding on to it an appropriate description of conditional termination criteria (discontinuity functions) as it shall be presented in due course. In addition, the standard should also be expanded in a further sense. If I currently want to maintain a library of ODE solvers, I shall most certainly need a linear system solver in the majority of them. If this (lower end) interface is not standardized; too, I have to maintain any number of basically identical linear system solvers in my ODE library. For this reason, it would make sense

to standardize also this lower end of the ODE code. Different linear system solvers may then reside in a (possibly separate) library (e.g. general LSS and sparse LSS) out of which I may select whatever seems appropriate to go with whatever ODE solver I want to use.

The basic problem here is that numerical mathematicians (that is: the producers of the integration algorithms) are not necessarily equally well trained for computer science. However, the above mentioned considerations are basically those of a computer scientist, and not of a mathematician, and it is sometimes hard to convince mathematicians of the fruitfulness of such considerations.

When I asked G. Wanner after his presentation of a newly developed A-contractive algorithm (that is -- in terms of Germund Dahlquist -- a G-stable algorithm, if I understood Wanner correctly), a presentation which was given during the Rutishauser Symposium in Zurich [21], whether he had already produced any executable code for his algorithm, he looked at me as you may look at a very strange bird and answered that his work was purely theoretical, and that the job of coding would definitely have to be someone else's task. I, however, can assure you that the end user is certainly not willing to try his hands at such an adventure because he is (1) not able to do it, and (2) not able to judge beforehand whether this new algorithm will get him anywhere or not. It has already too often happened (also to myself!) that a good looking new algorithm turned out to be a complete failure after it had been coded into a program. The effort spent on coding this algorithm was tremendous, and it became afterwards never clear whether the problem was really with the algorithm or just with the code!

Even P. Henrici (who is known for his sympathy with "practical" solutions to "real world" problems as opposed to "ivory tower" solutions to "green grass" problems), when I asked him a couple of years ago whether he would agree to join the consulting committee for my PhD thesis which I was to write on "Combined Continuous/Discrete System Simulation" [5], answered that he did not know anything about simulation but that he was certainly willing to learn something about. I then told him that this was beside the point in that I was sure that he knew a lot about simulation. The problem was simply that he did not know that he knew something about! This indeed is a severe problem in that it indicates that the average mathematician, even being a specialist in numerical integration, does not scan the literature for articles on "simulation", even though these articles could be as important to his work as any contribution on "numerical integration".

Coming back to my former issue on discontinuity handling: What has happened since the Urbana meeting? I was glad to realize that there were at least a few mathematicians around who took my comments seriously enough to spend some thoughts on them and think of some remedies. In fact, some results may already be summarized now.

1) For non-stiff problems, the discontinuity handling problem had already been solved prior to the Urbana meeting by some people like Alan Pritsker [18] and myself [5] in a fairly general way. Applying an RK-algorithm, we just have to make sure that the discontinuity takes place at the end of an integration step. Time events are handled by simply reducing the step size if the next event is shortly ahead. State events are handled by iterating back to the unknown event time by any available method. Pritsker uses bi-section [18], whereas I resorted to inverse Hermite interpolation [5]. For obvious numerical reasons, it is important not to code the discontinuity itself in the ODE set but only the state condition (e.g. by means of conditional termination criteria). The discontinuity itself is then expressed by context switching during the execution of the event (after execution of the event, another set of ODE's becomes active). A somewhat more mathematical view of this procedure can be found from Manshardt [16].

2) For stiff problems, one usually wants to apply multi-step integration, for which the step size adjustment at least creates a certain amount of overhead. The now common approach is to maintain two independent clocks, the external simulation clock and the internal integration clock. The integration proceeds with optimized step size and order, whereas the synchronization with the simulation clock is done by interpolating back using the Nordieck vector. This methodology may also be applied to discontinuity handling if the state conditions are formulated as an adjunct set of discontinuity functions

$$g(x, t)$$

with the meaning that the simulation run terminates at either the final time t_f or when the first of the g_i functions crosses through zero in either direction, whatever comes first. According to my knowledge, we owe this formulation to Mike Carver [1].

3) Two of the available GEAR-codes (one by Mike Carver [3], and the other by Kahaner [15]) have meanwhile been upgraded to contain a discontinuity handling mechanism. At least for the Kahaner implementation (which I consider to be the best of the currently available GEAR-codes -- it differs from the Hindmarsh implementation in that the former DIFSUB subroutine has been modularized into about 20 smaller subroutines which are now much easier understandable and avoid all those "dirty" GOTO statements pointing backward in code), the discontinuity mechanism seems to have been partly triggered by my comments at Urbana, in that MackHyman (who was taking part in these discussions at Urbana) produced shortly after that meeting a new version of the Kahaner code which now allows to specify an adjunct set of discontinuity functions by using precisely the mechanism advertised above.

4) I might want to suggest the introduction of an additional flag for indication whether really all crossings are to be detected or whether only positive or only negative crossings are important. This is not really essential, but it is useful in many applications to formulate hysteresis effects (e.g. a heating system is switched on when the temperature falls below 18 degrees centigrad, while it is switched off only after the temperature reached 21 degrees centigrad -- as too frequent switching may damage the switching mechanism). By use of the above mentioned flag, a lot of unnecessary context switching can be avoided which makes the user program shorter.

5) Unfortunately, the above mentioned mechanism does not really solve all problems yet. The reason for this is as follows: After a discontinuity took place, the integration needs to be restarted. In case of the GEAR-codes, this means that the algorithm has to start again at an order of one. If discontinuities occur at frequent rates, the integration algorithm needs to be restarted again and again. In this way, one ends easily up by having an extremely inefficient implementation of the trapezoidal rule, as higher orders get no chance to build up!

6) It is quite frequent in engineering applications that the accuracy requirements are not very severe (e.g. .0001). Then, an efficient low order algorithm may do a better job on the problem.

7) This was realized by Deuffhard who developed a new low order code for stiff integration which looks very promising [10]. For such an algorithm, also the discontinuity handling becomes easier. (I have, however, not yet found the time to try my own hands at this code, and so I cannot give any final judgment yet.)

8) If higher accuracy requirements are important, a higher order algorithm has its advantages. A typical engineering application for this would be the simulation of a combined analog and digital circuitry containing memory elements (flip flops). In such applications, it is extremely important to know whether spikes are around, and what the maximum transient voltages in some parts of the analog circuit are. The answer to these questions is very sensitive to parameter variations and also to event timing. For this reason, the system usually must be simulated with a relative accuracy of 10^{-5} .. 10^{-6} .

9) After I mentioned this problem to Bill Geer, he realized that his algorithm could be substantially improved if the "warming up" period could be made more efficient, that is, if we can avoid to restart after discontinuities at order one. For this purpose, he developed in the meantime RK-starters for GEAR-codes [13]. Although I have not yet found the time to implement such an algorithm personally, I am fully convinced that this amendment shall make the code substantially faster

when frequent discontinuities occur.

- 10) I would definitely recommend to the LosAlamos group (MacNyma) to consider the incorporation of such an algorithm into their code.

In fact, it were these reactions to my comments raised at Urbana which were basically responsible for encouraging me to sit down now and to write this statement, let me discuss the following "application problem" which was invented by me some time ago as a new benchmark problem for simulation software [5]:

Even if a good amount of progress has been achieved since 1979, the question of discontinuity handling is by no means settled yet. To illustrate this statement, let me discuss the following "application problem" which was invented by me some time ago as a new benchmark problem for simulation software [5]:

Given a set of domino stones from a domino game (usually 55, but any number will do). Place these stones in a distance "d" from each other. If now the first of these stones is pushed, all stones fall flat. The question to be answered is, at which distance "d" between two consecutive stones the chain velocity is maximized.

This problem is, as I hope, of a sufficiently "green grass" nature to refresh even the heart of a mathematician. However, the problem is by no means academical as precisely the same simulation problem arises in many "practical" applications, e.g. the heating of steel ingots in a steel soaking pit and slabbing mill, or chemical batch reactions with charging and discharging of batch reactors, or the traffic flow around an intersection where each car may be modelled by a set of (discontinuous) ODE's whereby new cars may enter the considered area at any time while old cars disappear from the region after they stayed in the system for some time.

What is common to all these applications? Obviously, whenever a discontinuity arises, the entire structure of the problem may change, and even the number of ODE's is varying with time. We call these problems "variable structure problems". Each domino stone in the above presented problem has to obey Newton's law, that is, is represented by a second order system or by a set of two first order ODE's. Taking the 55 stones of the game, we obtain altogether a 110th order model. However, it makes little sense to program the model in this way as only a very small number of stones are moving simultaneously. Some stones may have fallen already down while others are still untouched. Moreover, the physical law, governing the motion of any stone in the system, is the same. It, therefore, makes much more sense to code an entity to represent any "model" stone (we could talk of a "stone type"), and allow new stones of the prescribed type to be generated at event times while others may be destroyed at event times. It is evident that there shall exist a (possibly even continuous) interaction between falling stones which has to be taken into account. Any ODE solver, as they are currently marketed, is theoretically able to handle this situation as

long as the code provides for appropriate discontinuity handling mechanisms in the previously discussed sense. However, the portion of the program which remains to be user coded will still be substantial. We feel that, again, the user should be relieved of that part of the coding which is common to all the above mentioned examples. Software for this type of applications (that is: variable structure simulation) is currently under development at our group, but it is not yet finalized. This software shall consist of a FORTRAN coded subroutine package GASP-VI [20], an extension to the existing package GASP-V [4,5,6], together with a PASCAL coded preprocessor (front end) COSY [5,7,8] to make the user interface a little more convenient and less error prone.

3. SPARSE LINEAR SYSTEM SOLVER

Looking into the history of GEAR-codes, the original code proposed by Bill Gear himself suffered from numerically "dirty" programming, in that it happened frequently during the execution of a problem that the program suddenly died due to division by zero (a problem which I never could convince my computer to handle in a convenient way!) or due to similarly unpleasant effects.

The next "generation" of the GEAR-code, implemented by Alan Hindmarsh, was considerably better, in that:

- a) the code was numerically cleaned up -- no division by zero occurred thereafter, and
- b) the code executed considerably faster, as the originally used (very primitive) linear system solver was replaced by a far superior code.

In particular, this latter improvement and the convenient availability of the code made this implementation very famous and widely used. (In fact, it is still widely used.)

Meanwhile, this second implementation has seen (at least) two successors, namely the previously mentioned implementations by Mike Carver [3] and the implementation by Dave Kahaner [15]. Although I like the Kahaner implementation best (for reasons of programming cleanliness) the Carver implementation enjoys an important numerical advantage.

Remember that the Hindmarsh implementation basically differed from the original one in that the linear system solver had been replaced, that is, that portion of the code had been replaced in which most of the computation time elapses. Mike Carver again modified this portion of the program in that now, in addition to the formerly used linear system solver, also a SPARCS version is available making use of the Reid sparse matrix routines [19]. The user may decide (through an additional switch) whether he wants to use the sparse version of the linear system solver or whether he wants to stay with the conventional solver.

Taking, for instance, the Collatz equation [9] with 33 equidistantly spaced discretization points (a problem which is by the way *not* stiff -- the stiff algorithms within the GEAR-code does a terrible job on this set of 66 ODE's!), we have compared the Adams algorithm (within the GEAR-code) once using sparse matrix techniques and once using the conventional algorithm. The sparse version was by a factor of 3 faster than the conventional version. (This problem was tried out by use of the FORSIM-VI software [3].)

Similar results can be reproduced from almost *any* example (be it stiff or non-stiff), as soon as a PDE is involved, or -- soon as the order of the system is larger than about 20. (In most publications, a break even point of about $n=50$ can be found, but this figure is derived primarily from storage allocation considerations -- a problem which is getting less important with the advent of modern virtual memory systems. It is now the execution time which plays the key role, and, here, sparse matrix solutions become attractive even at lower orders.)

This improvement is fairly easy to achieve, and it is, therefore, surprising to me that most of the available codes do ignore this possibility. This is presumably the way to gain efficiency with very little effort! We would strongly recommend to the Loadcase group that they should consider this improvement for their next release.

4. AUTOMATED PARTITIONING

It is quite common to many applications (e.g. in chemical engineering) that some portions of the model are considerably faster than others (e.g. a chemically reacting system consisting of fast and of slow reactions). It seems intriguing to try to reduce the overhead involved in the numerical integration of the slow subsystem by splitting the system into a fast and into a slow portion and by using different step sizes (and possibly even different algorithms) for the two of them [17]. This works quite well for some applications, e.g. for selftuning regulators with a (fast) inner loop and a (considerably slower) outer adaptation loop. (We tried this out by use of the SIMNON software [11,12].)

However, this partitioning scheme is not always easily done. It requires quite some expertise to "master" such pieces of software. Moreover, it is not guaranteed that such a partitioning scheme even exists for a particular stiff system. In fact, if the system is nonlinear, the eigenvalues of the Jacobian may move around with time freely, and it may well happen that some modes of the system are "fast" during some period of time, while they are "slow" during other periods. Again, it was Mike Carver who came up with a brilliant (because extremely simple) idea for automated partitioning which he presented in 1980 during an International Conference on Simulation held at Interlaken, Switzerland [2]. His method requires one single additional parameter to be user tuned, a parameter which even has some physical meaning

assigned to it, which makes the adjustment reasonably easy even for engineers. Beside from this parameter, the partitioning is fully automated and even adaptive, in that it may vary with time.

Obviously, the last word is not yet said about automated partitioning, but the approach taken by Carver is definitely a good step forward and shows where future research can still be done.

REFERENCES

- [1] Carver, M. B., (1977) "Efficient Handling of Discontinuities and Time Delays in Ordinary Differential Equation Systems". Proceedings of the International Conference SIMULATION'77 held at Montreux, Switzerland, June 22-24, 1977. (M. Hamza, ed.). pp. 153-158. Acta Press, P.O.Box 354, CH-8053 Zurich.
- [2] Carver, M. B. and S. R. MacEwan, (1980) "Automatic Partitioning in Ordinary Differential Equation Integration". Progress in Modelling and Simulation, Academic Press, London, 1982. (P. E. Collier, ed.).
- [3] Carver M. B., D. G. Stewart, J. M. Blair and W. H. Selander, (1978) "The FORSIM VI Simulation Package for the Automated Solution of Arbitrarily Defined Partial and/or Ordinary Differential Equation Systems". Report No. AECL-5821. Available from: Atomic Energy of Canada Ltd, Chalk River Nuclear Laboratories, Chalk River, Ontario K0J 1J0. 154 pp.
- [4] Collier F. E., (1978) "The GASP-V Users' Manual". Available from the author.
- [5] Collier F. E., (1979) "Combined Continuous/Discrete System Simulation by Use of Digital Computers: Techniques and Tools". PhD Thesis. Report No. Diss ETH No 6483. Swiss Federal Institute of Technology Zurich. Accepted for publication as a book by Academic Press, London.
- [6] Collier F. E. and A. E. Blits, (1976) "GASP-V: A Universal Simulation Package". Proceedings of the 8th AICA Congress on Simulation of Systems, Delft, The Netherlands, August 23-28, 1976. North-Holland Publishing Company. (L. Dekker, ed.) pp. 391-402.
- [7] Collier F. E. and A. P. Bonguicini, (1979) "The COST Simulation Language". Proceedings of the 9th IMACS Congress on Simulation of Systems, Sorrento, Italy, September 23-27, 1979. North-Holland Publishing Company. (L. Dekker, G. Savastano and S. C. Venetocchietto, eds.) pp. 271-281.

- [8] Cellier F. E., M. Rinvall and A. P. Bongulielmi, (1981) "Discrete Processes in COSY". Proceedings of the International Workshop on Modelling and Simulation Methodology held at Cosenza, Italy, April 9-11, 1981.
- [9] Collatz L., (1960) "The Numerical Treatment of Differential Equations". Springer Verlag, Berlin, pp. 323-329.
- [10] Deuflihard P., G. Bader and U. Nowak, (1980) "LARKIN - A Software Package for the Numerical Simulation of LARGE Systems Arising in Chemical Reaction KINetics". Proceedings of the Workshop on Modelling of Chemical Reaction Systems held at Heidelberg, FRG, Sept. 1-5, 1980.
- [11] Elmqvist H., (1977) "SIMNON - An Interactive Simulation Program for Nonlinear Systems". Proceedings of the International Conference SIMULATION'77 held at Montreux, Switzerland. (M. Hamza, ed.) pp. 85-89. Acta Press, P.O.Box 354, CH-8053 Zurich.
- [12] Elmqvist H., (1975) "SIMNON: An Interactive Simulation Program for Nonlinear Systems. User's Manual". Report No 7502, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- [13] Gear C. W., (1980) "Runge-Kutta Starters for Multistep Methods". ACM Transactions on Mathematical Software, Vol. 6, No. 3, September 1980, pp. 263-279.
- [14] Hindmarsh A. C., (1978) "A Tentative User Interface Standard for ODEPACK". Report No. UCID-17954. To be obtained from: Technical Information Dept., Lawrence Livermore Laboratory, University of California, Livermore CA 94550.
- [15] Kahaner D. (1979) "A New Implementation of the Gear Algorithm for Stiff Systems". Unpublished private communication. For further detail contact: Dr. MacHyman, University of California, LosAlamos Scientific Research Laboratory, Contract W-7405-ENG-36, P.O.Box 1663, LosAlamos NM 87545, U.S.A.
- [16] Mannhardt R., (1978) "One-Step Methods of Any Order for Ordinary Differential Equations with Discontinuous Right-Hand Sides". Numerische Mathematik, Vol. 31, pp. 131-152.
- [17] Paluszinski O. A. and J. V. Wait, (1977) "Simulation of Partitioned Linear/Nonlinear Systems". Proceedings of the International Conference SIMULATION'77 held at Montreux, Switzerland, June 22-24, 1977. (M. Hamza, ed.) pp. 134-139. Acta Press, P.O.Box 354, CH-8053 Zurich.
- [18] Pritsker A. A. B., (1974) "The GASP-IV Simulation Language". John Wiley.
- [19] Reid J. K., "Sparse Matrix Routines". For further detail contact: Dr. J. K. Reid, Atomic Energy Research Establishment (AERE), Harwell, United Kingdom.
- [20] Rinvall M. and F. E. Cellier, (1982) "The GASP-VI Simulation Package for Process-Oriented Combined Continuous and Discrete System Simulation". Proceedings of the 10th IMACS Congress on Simulation of Systems. Montreal, Canada, August 8-13, 1982.
- [21] Wanner G. (1980) Presentation given during the Rutishauser Symposium. (Unpublished.) For further detail contact: Dr. G. Wanner, Section de Mathématiques, Université de Genève, P.O.Box 124, CH-1211 Genève, Switzerland.

END

FILMED

1-83

DTIC