STIFF DIFFERENTIAL SYSTEMS
THE IBM RESEARCH SYMPOSIA SERIES

1971: Computational Methods in Band Theory
      Editors: P.M. Marcus, J. F. Janak, and A.R. Williams

1972: Computational Solid State Physics
      Editors: F. Herman, N. W. Dalton, and T. R. Koehler

1972: Sparse Matrices and Their Applications
      Editors: D. J. Rose and R. A. Willoughby

1972: Complexity of Computer Computations
      Editors: R. E. Miller and J. W. Thatcher
      Associate Editor: J. D. Bohlinger

1973: Computational Methods for Large Molecules
      and Localized States in Solids
      Editors: F. Herman, A. D. McLean, and R. K. Nesbet

1973: Ion Implantation in Semiconductors
      and Other Materials
      Editor: Billy L. Crowder

1974: Stiff Differential Systems
      Editor: Ralph A. Willoughby

A Continuation Order Plan is available for this series. A continuation order will bring
delivery of each new volume immediately upon publication. Volumes are billed only upon
actual shipment. For further information please contact the publisher.
Proceedings of the International Symposium on Stiff Differential Systems
organized and sponsored by the Thomas J. Watson Research Center and IBM
Germany, held at Wildbad, Germany, October 4-6, 1973

Softcover reprint of the hardcover 1st edition 1974
A Division of Plenum Publishing Corporation
227 West 17th Street, New York, N.Y. 10011
A Division of Plenum Publishing Company, Ltd.
4a Lower John Street, London W1R 3PD, England

All rights reserved
No part of this book may be reproduced, stored in a retrieval system, or transmitted,
in any form or by any means, electronic, mechanical, photocopying, microfilming,
recording, or otherwise, without written permission from the Publisher
The papers in these proceedings were presented at an International Symposium on Stiff Differential Systems, which was held at the Hotel Quellenhof, Wildbad, Federal Republic of Germany, October 4-6, 1973. The symposium was organized by IBM Germany and sponsored by the IBM World Trade Corporation.

On behalf of all the participants we wish to express our appreciation to the sponsors and organizers for their generous support, particularly to Dr. G. Hübner, representing Scientific Relations, IBM Germany, and Dr. G. Kozak, representing IBM World Trade Headquarters, New York.

The purpose of the conference was to provide an intensive treatment of all aspects of a difficult problem class, stiff differential systems. Some major fields of interest of attendees and contributors are: 1) Modeling and problem solving in scientific and technological applications, 2) Qualitative theory of stiff systems, 3) Numerical Analysis, including design, validation, and comparison of algorithms, as well as error and stability analysis, and 4) Computer Science, in particular problem-oriented programming languages, program packages, and applications-oriented computer architecture.

The papers in these proceedings are not intended for the novice in the initial value field, but the very extensive bibliography will provide an excellent background for those readers not already familiar with the concepts. In particular, each of three recent books [Gear (1971 D), Lapidus and Seinfeld (1971), and Lambert (1973 A)] devotes a chapter to stiff systems. While another book [Stetter (1973)] does not directly address itself to the stiff system question, it does represent a fundamental treatise on the qualitative aspects of the numerical initial value problem. Dr. P. J. van der Houwen, Mathematical Center, 2e Boerhavestraat 49, Amsterdam, Netherlands, as well as a number of the
authors in these proceedings are the principal investigators on research projects in stiff systems. It is suggested that the interested reader contact these professionals to receive copies of the department reports on this subject.

Dr. R. A. Willoughby
Symposium Chairman

Dr. P. Schweitzer
Symposium Manager
CONTENTS

International Symposium on Stiff Differential Systems: Introduction .......................... 1
Ralph A. Willoughby

Composite Multistep Methods and Stiff Stability ............................. 21
Theodore A. Bickart and William B. Rubin

Numerical A-Stability for Difference-Differential Systems ......................... 37
R. K. Brayton

Computation of Real-Time-Control in Aerospace Applications by Multiple Shooting Procedures: Extended Abstract ....................... 49
R. Bulirsch and H.-W. Branca

Simulation of Chemical Kinetics Transport in the Stratosphere .................... 51
J. S. Chang, A. C. Hindmarsh, and N. K. Madsen

The Sets of Smooth Solutions of Differential and Difference Equations ........... 67
Germund Dahlquist

Integration Package for Chemical Kinetics .......................................... 81
Lennart Edsberg

Optimal Second Derivative Methods for Stiff Systems .................................. 95
W. H. Enright

The Stability of Automatic Programs for Numerical Problems ........................ 111
C. W. Gear, K. W. Tu, and D. S. Watanabe
An Implementation of Gear's Algorithm for CSMP III
A. R. Gourlay and H. D. D. Watson .......................... 123

A Pseudodynamic Method for Solving Nonlinear Algebraic Equations
G. Hachtel and M. Mack ........................................ 135

The Validation and Comparison of Programs for Stiff Systems
T. E. Hull ..................................................... 151

Numerical Solution of Singular Perturbation Problems
Heinz-Otto Kreiss ............................................ 165

Two Unconventional Classes of Methods for Stiff Systems
J. D. Lambert .................................................. 171

The Occurrence and Numerical Solution of Physical and Chemical Systems Having Widely Varying Time Constants
L. Lapidus, R. C. Aiken, and Y. A. Liu ....................... 187

A Stiff System Package Based on the Implicit Midpoint Method with Smoothing and Extrapolation
Bengt Lindberg .................................................. 201

Construction of a Family of Second Order A-Stable K-Step Formulas Depending on the Maximum Number, 2K-2, of Parameters
Werner Liniger and Thierry Gagnebin ......................... 217

Stiffness and Accuracy in the Method of Lines Integration of Partial Differential Equations
A. M. Loeb and W. E. Schiesser ............................... 229

Economical Global Error Estimation
Hans J. Stetter .................................................. 245

Consistency and Stability for One-Step Discretizations of Stiff Differential Equations
M. van Veldhuizen .............................................. 259
CONTENTS

Bibliography ........................................ 271
Author List ......................................... 307
Index for Papers and Bibliography ................. 319
SYMPOSIUM PARTICIPANTS, SESSION CHAIRMEN, SPEAKERS, COAUTHORS

R. Aiken (Coauthor with Lapidus), Princeton University; L. Alsop, IBM Research; P. Banderet, University of Neuchâtel; R. Baumann, Technical University of Munich; T. A. Bickart (Speaker), Syracuse University; A. Blaser, IBM Heidelberg Science Center; Branca, H.-W. (Coauthor with Bulirsch), University of Köln; R. K. Brayton (Speaker), IBM Research; R. Bulirsch (Speaker), Technical University of Munich; T. Chambers, Central Electricity Generating Board London; J. S. Chang (Speaker), Lawrence Livermore Laboratory; G. Dahlquist (Speaker), Royal Institute of Technology Stockholm; B. Dejon, University of Erlangen-Nuremberg; J. Descloux, Federal Polytechnical University of Lausanne; L. Edsberg (Speaker), Royal Institute of Technology Stockholm; W. H. Enright (Speaker), University of Toronto; S. Fillippi (Session Chairman), University of Giessen; L. Fox (Session Chairman), Oxford University; R. Frank, Technical University of Vienna; T. Gagnebin (Coauthor with Liniger), University of Neuchâtel; C. W. Gear (Speaker), University of Illinois; Y. Genin, Electronic Research Laboratory Brussels; F. Göbber, University of Tübingen; A. R. Gourlay (Speaker), Loughborough University of Technology; R. D. Grigoriiff, Technical University of Berlin; G. D. Hachtel (Speaker), IBM Research; K. P. Hadeler, University of Tübingen; J. Halliday, British Aircraft Corporation Preston; M. L. Henrici, ETH Zurich; P. Henrici (Session Chairman), ETH Zurich; A. C. Hindmarsh (Coauthor with Chang), Lawrence Livermore Laboratory; P. J. van der Houwen (Session Chairman), Mathematics Center Amsterdam; G. Hübler, IBM Germany Sindelfingen; T. E. Hull (Speaker), University of Toronto; H.-O. Kreiss (Speaker), University of Uppsala; J. D. Lambert (Speaker), University of Dundee; L. Lapidus (Speaker), Princeton University; B. Lindberg (Speaker), Royal Institute of Technology Stockholm; W. Liniger (Speaker), IBM Research; Y. A. Liu (Coauthor with Lapidus), SUNY at Buffalo; A. Loeb (Coauthor with Schiesser), Naval Ship Engineering Center Philadelphia; M. H. Mack (Coauthor with Hachtel), IBM Research; N. K. Madsen (Coauthor with Chang), Lawrence Livermore Laboratory; M. Mäkelä, Helsinki University of Technology; R. E. A. Mason, IBM Canada Toronto; B. Meister, IBM Heidelberg Science Center; A. Morreew, University of Paris; J. P. Morreew, University of Paris; K. Nickel, University of Karlsruhe; W. B. Rubin (Coauthor with Bickart), IBM Poughkeepsie; E. Shauer IBM Heidelberg Science Center; W. E. Schiesser (Speaker), Lehigh University; P. Schweitzer (Symposium Manager), IBM Germany Sindelfingen; A. van der Sluis, University of Utrecht; M. N. Spijker, Technical University of Leiden; H. J. Stetter (Speaker), Technical University of Vienna; J. Stoer, The University of Würzburg; K. W. Tu (Coauthor with Gear), Rutgers University; M. van Veldhuizen (Speaker), University of Utrecht; D. S. Watanabe (Coauthor with Gear), University of Illinois; H. D. D. Watson (Coauthor with Gourlay), IBM Peterlee Science Center; H. Werner (Session Chairman), University of Westfälische Wilhelms; G. W. T. White, Cambridge University; R. A. Willoughby, IBM Research.
INTRODUCTION

During the last decade there has been an extensive development of numerical integration methods designed to be efficient for stiff systems of ordinary differential equations. The purpose of the symposium was to focus attention on this difficult problem class and to bring together for mutual interaction many of the leading experts. The papers in the proceedings together with the unified bibliography should provide an adequate basis of information for persons interested in contributing to this field of research.

The references in the main bibliography are limited to those which have appeared or are about to appear in periodicals, books, and proceedings with unlimited availability. Department reports and PhD theses are referred to in this introduction and in individual papers. The subject index for the papers in the proceedings also includes a subject classification for articles referenced in the main bibliography.

Numerical Analysis research relative to topics such as stiff systems is concerned with finding, understanding, and providing reliable algorithms for well posed problems which are a priori expressed in certain canonical forms. This is quite proper since without this this type of discipline, the computing field and problem solvers would be forced to depend on a large number of unrelated ad hoc devices and schemes whose reliability and limits of applicability are unknown.

A design feature which is very important for general purpose algo-
The algorithms is that they have an optimality* relative to computing cost for a given specified accuracy [see e. g. T. E. Hull, "A search for optimum methods for the numerical integration of ordinary differential equations," SIAM Rev. 2(1967), pp. 647-654; Hull et al (1972); Krogh (1973B); B. L. Ehle, Mathematics Report, University of Victoria, Canada (1972); P. Fox(1972)].** Hull's paper in this proceedings is concerned with validation and comparison of programs for the numerical integration of stiff systems. One of the difficult questions in any comparison project is how to choose the test problems. Each practical system has special features which are unique to that problem. This makes global classification of system characteristics a very fuzzy procedure. However, a reasonable mix of textbook problems with canonical problems from various applied fields seems to be a sensible approach to this question. The Lapidus-Aiken-Liu paper gives the reader a good insight into how stiff systems arise in physical and chemical applications. Most of the papers have several examples, but an Information Processing Department Report by G. Bjurel, G. Dahlquist, B. Lindberg, S. Linde, and L. Oden, "Survey of stiff ordinary differential equations," NA 70.11, Royal Institute of Technology, Stockholm (1970) contains an excellent spectrum of examples. A highly readable historical discussion of stiff systems is given in [Dahlquist (1973)]. One should also see the special chapter on stiff systems in each of the recent books on the numerical initial value problem [Lambert (1973A); Gear (1971D); Lapidus and Seinfeld (1971)].

Bickart, Dahlquist, Gear, Hull, and Lapidus as well as a number of other speakers, session chairmen, and participants have research project in the field of stiff differential systems. Their department reports contain a great deal of information, and the reader should contact these professionals directly to obtain these reports.

During the period 1966-68 there was a considerable amount of independent research being conducted on general purpose approaches to the integration of stiff differential systems, and after the IFIP Congress in Edinburgh [Morrell (1968)] it was recognized throughout the numerical analysis community that this was an important and difficult problem area. The awareness of the numerical pathologies associated with these systems has a much longer history dating back at least to [Fox and Goodwin (1949)]. The Curtiss-Hirschfelder 1952 paper in the Proceedings of the National Academy of Science U. S. A. clearly pointed out the importance of stiff systems, especially in the area of Chemical Kinetics. These early insights were largely overlooked by numerical analysts for over fifteen years. This was probably due to the feeling that stiffness was a very isolated situation which could best be

---

*Optimality does not mean merely better than other existing methods.
**The brackets indicate bibliographic references. Citations in the main bibliography are made via author(s) last name(s) and year.
INTRODUCTION

handled on a system by system basis with special purpose techniques.

According to [Hirschfelder (1963), p. 368] the term "stiff" is used because such systems correspond to tight coupling between the driver and driven in servo-mechanisms. The particular class of stiff systems which were studied by Curtiss and Hirschfelder can be written as a pair of first order autonomous differential equations

\[ \dot{x} = a(x,y) \]  \hspace{1cm} (1a)

\[ \dot{y} = y - G(x) \]  \hspace{1cm} (1b)

where \( \dot{x} = (dx/dt) \) and \( x, y, t \) are scalar variables. In their analysis they dealt with \( dy/dx \) and treated \( y \) as a function of \( x \). Given that \( \Delta x \) is the desired sampling interval for \( x \), then system (1) is stiff provided \( |a(x,y)/\Delta x| \ll 1 \) and \( G(x) \) varies with \( x \) considerably more slowly than does \( \exp(x/a(x,G(x))) \). Figure 1 on page 236 of their article shows a tangent vector field in the \((x,y)\)-plane. This field streams in to a smooth particular curve \( y = Y(x) \) which is a special solution to system (1).

A pseudo steady state approach to system (1) consists of setting \( (dy/dt) = 0 \) and integrating the equation \( (dx/dt) = a(x,G(x)) \) with an appropriate initial condition for \( x \). The papers by Chang-Hindmarsh-Madsen, Dahlquist, Edsberg, and Lapidus-Aiken-Liu all discuss this method of asymptotic approximation. Kreiss and van Veldhuizen are also concerned with asymptotic expansions as is Stetter in his discussion of economical global error estimation. It is important to realize in dealing numerically with stiff systems that there are two small parameters, the step size \( h \) and the stiffness measure \( \varepsilon \), and that except for boundary layer transitions one has \( h \gg \varepsilon \).

Dahlquist used the pseudo steady state approach in 1951 on a stiff system which arose in the design of a control system ["Mathematical and computational studies of the two-dimensional motion of RB 321" (Swedish), Research Institute of National Defense, Stockholm (1952)]. In the mid sixties Dahlquist and his students analyzed and tested an extension of this approach which is called SAPS, the Smooth Approximate Particular Solution method [Dahlquist (1968)]. The Dahlquist and the van Veldhuizen papers are both concerned with an asymptotic analysis as \( \varepsilon \to 0 \) (infinite stiffness) of the decomposition of the solution into a smooth component and a rapidly decaying component. The later's paper is a highly condensed summary of his recent PhD thesis under Professor Dr. van der Sluis at the University of Utrecht, the Netherlands.

Kreiss' paper is a summary of joint work with L. R. Abrahamson and H. B. Keller. A more complete version of this work is contained in the Computer Science Department Report, Uppsala, Sweden (1973),
"Difference approximations for singular perturbations of systems of ordinary differential equations." The boundary value problem which is analyzed is the vector system:

$$\epsilon y''(x) + A(x)y'(x) + B(x)y(x) = F(x), \ y(0) = \alpha, \ y(1) = \beta.$$ 

It is shown how to derive asymptotic expansions in powers of $\epsilon$ using solutions of $A(x)y'(x) + B(x)y(x) = F(x)$, which is the reduced equation, and of equations with constant coefficients

$$\epsilon y''(x) + A(i)y'(x) = G_i(x), \ i = 0,1.$$ 

Three difference methods are analyzed and one of these does not require the system to be in partitioned form. The later method more readily applies to nonlinear problems.

The relationships among stiff systems, pseudo steady state approximations, and asymptotic methods relative to singular perturbations are discussed in some detail in the Lapidus-Aiken-Liu paper [see also Miranker(1974)].

Just as in the analysis presented by Kreiss, it is always desirable when possible to reduce the problem or parts of the problem to linear systems with constant coefficients. This approach has been studied for the case of quadratic initial value problems [P. D. Gerber, "Linearization of $\dot{x} = xx + Bx + c$ by $x = f(t,u,\bar{u})$, $u = B\bar{u} + C\bar{u} + d$," IBM Research Report RC4528 (1973)].

Let the stiff system be denoted by

$$\dot{\bar{u}} = F(u), \ u_0 = u(t_0) \quad (2a)$$

with

$$\frac{\partial F}{\partial u} = \left( \frac{\partial F_i}{\partial u_j} \right) = J \text{ (say)} \quad (2b)$$

as Jacobian matrix and

$$L = \max_{u;\Delta u \neq 0} \frac{||F(u+\Delta u) - F(u)||}{||\Delta u||} \leq \max_{u} ||\frac{\partial F}{\partial u}|| \quad (2c)$$

as the Lipschitz constant. The stiffness parameter $\epsilon$ introduced earlier is essentially $1/L$. A common way of describing stiff systems is via the spread and nature of the local eigenvalues, especially of their real part. If in (2c) one fixes $u$ and restricts $||\Delta u||$ to the open interval $(0,\delta)$ then the resulting local Lipschitz constant is essentially $L(u) = ||J(u)||$ where the matrix norm is subordinate to the vector norm for $u$. The spectral radius for $J$ is a lower bound for $L(u)$; i.e. $\rho(J) = \text{spectral radius} = \max |\lambda(J)|$ and $\rho(J) \leq L(u)$. Another scalar measure for $J$ is the logarithmic norm [Dahlquist (1958); Ström (1972); C. A. Desoer and H. Haneda, IEEE Trans. CT-19 (1972), pp. 480-486.]
\[ \nu(J) = \lim_{\varepsilon \to 0^+} \varepsilon^{-1} \left( \|I + \varepsilon J\| - 1\right) = \lim_{\varepsilon \to 0^+} \varepsilon^{-1} \ln \|\exp(\varepsilon J)\| \geq \nu(J) \]

where \( \nu(J) = \) spectral abscissa = \( \max \Re(\lambda(J)) \). These scalar measures played a basic role in Dahlquist's analysis in the early and mid 1950's of the stability, convergence, and error bounds for numerical integration methods for non-stiff systems. He and his students have extended some aspects of this analysis to the characterization of smooth particular solutions of stiff systems.

In the design of non-stiff integration methods one is willing to deal with a restriction of the form

\[ Lh < C \]

where \( h \) is the time step size, \( L \) is the global Lipschitz constant and \( C \) is a constant of order one which depends on the fixed-\( h \) stability of the method or on the convergence criterion of the corrector iteration. As stated before one seeks accurate solutions with \( Lh \gg 1 \) in the case of stiff systems. Of course if one is interested in the short duration boundary layer transitions, then for these excursions one must deal with the restriction \( Lh < C \). However one would like the average step size over the entire interval of integration to be independent of \( L \) so that the limit \( L^{-1} \to 0 \) can be a benefit rather than a disaster.

The boundary layer transition may be of no interest in itself and also be so brief that one treats it as a discontinuous adjustment of the initial condition to the target smooth particular solution. Lack of smoothness in the differential system and also step size changes in the numerical integration can reintroduce the need to consider boundary layer transitions. The two related papers by Dahlquist and by Lindberg consider the question of smoothing in relation to the implicit midpoint rule.

This readjustment to the correct smooth particular solution is also a critical aspect of the multiple shooting approach to real time control problems in aerospace applications. This area of research is summarized in the extended abstract by Bulirsch-Branca. Their full manuscript will be published later in a periodical, and the two recent PhD theses at the Technical University of Cologne by H. W. Branca and by U. Zimmermann also deal with numerical methods for this class of control problems. The latter both did their thesis work under Professor Dr. Bulirsch, who is now a member of the Mathematics Institute of the Technical University of Munich.

The research on stiff systems has had a number of recent contributions in the area of one step methods. These include two papers by researchers at Royal Dutch Shell Research, A. Prothero and A. Robinson, which will appear in [Watson (1973)] and in Math. Comp. J. S. Rosenbaum has a part of his PhD thesis to appear in BIT.
THREE ELEMENTARY STIFF SYSTEMS. This editor has had practical numerical experience with the following three examples of stiff systems. Some of the physical details have been simplified for purposes of exposition, but they should help to give the reader a better feeling for stiffness and they also form excellent "excruciating problems" for numerical analysis courses.

System I. Nuclear Reactor Control Rod Problem*.

\[ \begin{align*}
\dot{x} &= a(y - x) \quad (3a) \\
\dot{y} &= (b - d)x - (c - d)y + at(y + 1) \quad (3b)
\end{align*} \]

\( x(0) = y(0) = 0, \quad 0 < \alpha \ll 1, \quad \alpha b = \alpha d, \quad 0 < d < a < b < c. \) The eigenvalues of the Jacobian at \( t = 0 \) are \( \lambda = -(a - d), \quad -c; \) at \( t = T = (c - b)/\alpha \) the Jacobian is singular; and for \( t > T \) the reactor is supercritical (i.e. the solution grows exponentially). The purpose of this study was to examine the nature of a class of exponential excursions. System (3) would be integrated from \( t = 0 \) to \( t = t_1 > T. \) The system would then be altered \((\alpha < 0)\) to reflect the inward motion of the control rod, and this new system would be integrated for an interval of time \( t_1 \leq t \leq t_2 \) with the solution of the original system at \( t = t_1 \) as initial conditions. The concern in the subsequent integration focused on whether the exponential excursion remained within allowable limits.

A practical set of values is \( \alpha = 1/8, \quad d = 1/30, \quad a = 1/5, \quad b = 10, \quad c = 60. \) One has \( t_1 > (c - b)/\alpha = 400 \) and \( \lambda = -1/6, \quad -60 \) at \( t = 0. \) System (3) was integrated via certain specifications of the following three parameter family of formulas**[Liniger and Willoughby (1970)]

\[ u_+ - h(1 - \mu)\dot{u}_+ + h^2\xi\ddot{u}_+ = u + hu\dot{u} + h^2\nu\ddot{u} \quad (4) \]

In the original 1957 study* it was found that the backward Euler formula \((\mu = \nu = \xi = 0)\) coupled with \(2 \times 2\) matrix inversion was an effective integration procedure. The papers by Fox-Goodwin (1949) and by Curtiss-Hirschfelder (1952) had also concluded earlier that it is desirable to combine a globally stable implicit formula with matrix inversion for stiff systems.

The second diagonal Padé formula [Gragg (1972)] in (4) is given by \( \mu = 1/2, \quad \nu = \xi = 1/12. \) Liniger and Willoughby found that a step size of \( h = 5 \) could be used for the time interval \([0,400]\) with an accuracy of at least five significant digits. Note that \( \max |\lambda h| = 300. \)

A generalization of the backward Euler formula which is quite different from (4) is given by the class of backward differentiation

*Timlake and Willoughby (1957), unpublished.

**For simplicity of notation \( u = u_n = u(t_n), \quad u_+ = u_{n+1}, \) etc. and for system (3) \( u_+ = (x,y). \)
formulas \[\text{[Dahlquist (1973)]}\]
\[
\mathbf{h}_u = \nu u_+ + (1/2)\nu^2 u_+ + \cdots + (1/k)\nu^k u_+ \tag{5a}
\]
where
\[
\nu^{i+1}u_j = \nu^i u_j - \nu^i u_{j-1}. \tag{5b}
\]
This class of formulas along with many other classes was studied \((k<6)\) relative to fixed-\(h\) stability \([W. \ Liniger, \text{"On the stability of numerical integration methods for differential equations"} \text{ (German), PhD thesis, University of Lausanne (1956)}\] and was shown to be stable for any \(\lambda < 0\). Gear showed that this class is stiffly stable \((k<7)\) in a \((1967)\) Computer Science Report, University of Illinois, Urbana. Gear's work was presented at the \((1968)\) IFIP Congress. The proceedings of this conference is the reference \([\text{Morrell (1968)}]\).

**System II. Tunnel Diode Switching Problem**

\[
\begin{align*}
\dot{L}I &= E - V - RI \tag{6a} \\
\dot{C}V &= I - F(V) \tag{6b} \\
F(V) &= A_3(V^3 - 3A_2V^2 - 3A_1V) \tag{6c} \\
E &= E_0 + S(T) \Delta E \tag{6d}
\end{align*}
\]

Physical Units: \(E, V \text{ (VOLTS); } I, F(V) \text{ (10}^{-3}\text{ AMPS); } R \text{ (10}^{-3}\text{ OHMS); } L \text{ (10}^{-6}\text{ HENRIES); } C \text{ (10}^{-12}\text{ FARADS); } T \text{ (10}^{-9}\text{ SECS).}

---

*\(M. \ S. \ Axelrod, \ A. \ S. \ Farber, \ D. \ E. \ Rosenheim, \text{ IBM J. R&D } \(6\)(1962), pp. 158-169; and L. Esaki, \text{ ibid.}, pp. 170-178.*
For sake of simplicity $S(T)$ is assumed to be piecewise constant (i.e. $S = 0, \pm 1$). The case $S = 0$ represents the quiescent situation from an electrical engineering point of view. This is also a case of classical bistability [J. Moser, IBM J. R&D 5(1961), pp. 226-240]. The voltage $E_0$ is chosen here so that the point of inflection of the diode characteristic $P_0 = (A_2, F(A_2))$ is an intersection point of $I = F(V)$ with the load line $I = (E_0 - V)/R$. The point $P_0$ is a saddle point and the other two intersections $P_-$ and $P_+$ are stable points. The separatrix curve consists of the two singular solutions which approach $P_0$ as $T \to \infty$. This curve separates the plane into a pair of regions which are respectively attraction regions for $P_-$ and $P_+$.

A simplified view of switching is as follows: FORWARD SWITCHING
(a) Assume that the circuit is at the low voltage equilibrium point $P_-$ with $E = E_0$;
(b) One applies a step increase $E = E_0 + \Delta E$ such that the load line has only a high voltage intersection with the diode characteristic;
(c) Once the trajectory has entered the attraction region for $P_+$, $E$ is reset to $E_0$ and then $P \to P_+$ as $T \to \infty$.

The backward switching is completely analogous. R. Brayton studied tunnel diode switching via a Lyapunov function and R. Miranker analyzed this switching via singular perturbation theory.

The diode characteristic is much more complicated than is indicated in (6c) (e.g. the peak region is narrower than the valley region). Also the capacity is not a constant but is instead a function of $V$

$$C = C_0(1 - V/G)^{-1/2}$$

where $G = 0.6$ (say). These are technical matters which are not necessary to introduce into an idealized model problem for students.

The tunneling phenomenon in Quantum Physics accounts for the negative resistance region $F'(V) < 0$ [L. Esaki, Phys. Rev. 109(1958), pp. 603-604]. The peak to valley ratio of the current $\eta = \text{IMAX}/\text{IMIN}$ is an important physical parameter as is $\omega = \mu R - 1$ where $\mu = -F'(A_2)$. The stiffness may be varied by choosing $L$ and $C$ in the interval $[.01, 10]$. A convenient set of parameter values is as follows:

$$A_1 = .0167, A_2 = .148, A_3 = 6650$$
$$R = .017, E_0 = .245, \Delta E = .022$$

Then one has $P_{\text{MAX}} = (.0755, 10.44)$, $P_0 = (.148, 6.1)$, $P_{\text{MIN}} = (.22, 1)$, $\eta = 10.44$, $\mu = 100$, $\omega = 0.7$. 
INTRODUCTION

System III. Reversible Enzyme Kinetics Problem*

\[
\dot{X} = -AE + PW \tag{7a}
\]

\[
\dot{Y} = E - QW \tag{7b}
\]

where \(X = Y = 0\) at \(T = 0\), \(P = A + B(D - X)(1 - X)\), \(Q = 1 + CY^2\), and \(W = Y - X + E\). The parameter ranges are

\[
.01 < A, B < 10, \quad 0 < C < 1, \quad 1 < D < 1000, \quad .01 < E < 0.1.
\]

Since \(\dot{W} = (1 + A)E - (P + Q)W\), the sign of \(P + Q\) is a basic mechanism in the time evolution of \(X\) and \(Y\). Let \(A = B = C = 1\) and assume that \(0 < E \ll 1 \ll D\). In this case

\[
P + Q = -(D - 1)(X - 1) + (X - 1)^2 + (2 + Y^2)
\]

and \(P + Q = 0\) for \(X = 1 + (D - 1)(S + S^2 + 2S^3 + 5S^4 + \cdots)\) where \(S = (2 + Y^2)/(D - 1)^2\).

The reversible enzyme kinetics is indicated in the following diagram

**Diagram**

The problem here is one of estimation of the reaction rates. One performs a series of chemical experiments and observes the time evolution of certain compounds. The aim is to choose the parameters in the mathematical model in such a way that the numerical simulation reproduces essentially the same time plots. The papers by Edsberg and by Lapidus-Aiken-Liu discuss this type of problem [see also Curtis and Chance (1972); Hemker (1972B)].

Approximation problems of this character can be cast into the framework of optimization in a nonlinear least squares sense. This is of only marginal help in a global setting since sensitivities as well as correct measures for closeness of similar plots are typically known only after a large amount of experimentation and simulation. Of course, if one has a first guess which is within \(\varepsilon\), then the estimate within \(\varepsilon^2\) is often easy to obtain.

Problem modeling usually involves more than parameter estimation. There is also the determination of the mathematical structure of the governing equations and/or inequalities. The modeler has only partial information based on observation of natural phenomenon or of physical experiments. He or she has an inverse problem which is not necessarily well posed. There are some outputs and one seeks a (least complex) mathematical model which satisfies the known physical laws for the situation and which will yield close approximations to the

*T. T. Wu, Cornell Medical School, New York City (1967), notes.
given outputs under specified class of inputs. If the model is of a
dynamic nature, then one wants assurance that the simulations pro­
vide time plots which represent the character of the mathematical
model rather than subtle peculiarities of the numerical integration
procedure.

Control is the underlying concept in all aspects of problem modeling
and simulation. In numerical analysis* it concerns the control of the
numerical accuracy within a specified limit to the computational
complexity; in problem modeling it concerns restricting the degrees
of freedom in the experimental environment and also concerns the
precision of the measurements; and in the real world it is the
controlling mechanisms which the mathematical model is aimed at
revealing.

The efficiency associated with stiff system integration methods has
two quite separate aspects:

(a) The global stability (as $t \to \infty$), smoothing, step-size con­
trol, combining of approximate solutions, local and global
error estimation; and

(b) The matrix algebra required for each Newton (like) update
associated with solving the possibly nonlinear implicit
system for $u_\ast = u(t + h)$.

Stetter's paper unifies many of the features in (a), at least for
non-stiff systems, and also presents ideas on economical global
error estimation which were partly motivated by the work of the
astronomer Zadunaisky presented at the 1973 Dundee conference [the
proceedings reference is Watson (1973); see also Zadunaisky (1970)].
A nonstandard approach to estimating the local discretization error
is given via a smooth function $\bar{u}(t)$ which satisfies

$$
(\frac{d\bar{u}(t)}{dt}) - F(\bar{u}(t)) = h^p \phi + O(h^{p+1})
$$

where $\phi$ is the principal error function for the given $p^{th}$ order
method [Henrici (1962), p. 77]. The function $\bar{u}$ can be obtained for
example by (local) interpolation of the approximate numerical solu­
tion. One obtains the global error estimate via integration of the
variational equation

$$
\dot{\bar{w}} - J(t)\bar{w} = \phi
$$

where $J(t)$ is the Jacobian matrix ($\frac{\partial F}{\partial u}$) evaluated along the approx­
imate solution $u = \bar{u}(t)$.

*The primary controls in the numerical integration of ordinary dif­
ferential equations are the choice of the formula and of the step-size.
The close analogy between this new "differential corrections" technique and the well-known difference correction approach of L. Fox is discussed in Stetter's paper. The later procedure has been analyzed and demonstrated by Pereyra [see Author List] in a series of papers. It is the ability to iterate these procedures that removes the dilemma of obtaining an improved solution but losing the error estimate for the new approximation.

Interpolation methods can be used to extend the discrete values \( \hat{u}(t_i), 0 \leq i \leq N \), to a \( p \)-times continuously differentiable function \( \hat{u}(t) \) which satisfies (8) on the interval \( t_0 \leq t \leq t_N \). The function \( \hat{d}(t) = (d\hat{u}(t)/dt) - F(\hat{u}(t)) \) is called a "defect function" and the smallness of \( \hat{d}(t) \) shows that one is finding the exact solution of a nearby problem in the sense of backward error analysis.

Spline functions provide a systematic basis for the interpolation. H. Werner, who was a session chairman at the symposium, and his student, R. Runge, have made recent contributions in this area [H. Werner, "An application of regular splines to interpolation problems and initial value problems," (preprint) to appear as Mitteilungen der GMD, Bonn-Berlinkhoeven (1973); R. Runge, "Solution of initial value problems with the aid of a class of nonlinear spline functions," PhD thesis, University of Münster (1972)].

Aspect (b) of efficiency, which was defined on the previous page, aims at minimizing the computational complexity for each Newton update. In the case of large systems the exploiting of the fixed sparseness structure of the Jacobian matrix \( (\partial F/\partial u) \) for the system \( \dot{u} = F(u) \) has helped to achieve remarkable efficiencies for this update process [see Sparse Matrices in the Subject Index]. The numerical procedures in the Newton update are:

(i) Generation of the Jacobian elements;

(ii) Triangular Factorization of the coefficient matrix; and

(iii) Forward and Backward Substitutions.

A common shortcut is to treat the Jacobian as piecewise constant in \( u \) and to resort to steps (i - ii) only when the Newton convergence breaks down. Another simplification consists of evaluating the Jacobian elements via divided differences

\[
\frac{\partial F_i}{\partial u_j} = \frac{(F_i(u + \delta e_j) - F_i(u))}{\delta}
\]

where \( e_j \) is the \( j \)-th unit vector. This approach to (i) greatly reduces the programming effort, but it also introduces a noise amplification effect which is not well understood.
The development of a programming package for efficient integration of stiff differential systems is a large undertaking but a number of such packages now exist. Gear’s is the widest used and best known. The IBM program CSMP*, which was developed originally by R. Brennan has been extended a number of times. The paper by Gourlay-Watson describes their inclusion of a version of Gear’s integration method in the program package CSMP III.

Enright in his paper reports on a package which is based on his class of second derivative methods. These methods will be discussed later. The Lindberg paper describes a package based on the implicit midpoint rule

\[ hF(v) = u_+ - u, \ v = .5(u_+ + u) \tag{9} \]

with smoothing and extrapolation (to \( h \to 0 \)). Edsberg’s paper presents his package for Chemical Kinetics which is based on the above method. The paper by Dahlquist provides the underlying analysis for the theory of the smooth particular solution and for the implicit midpoint rule with smoothing and extrapolation.

Another large application area is that of Electronic Circuits, and the Hachtel-Mack paper concerns a special Davidenko-type parameter stepping method for solving nonlinear equations which arise in steady state device characterization problems. An integration method [Hachtel, Brayton, and Gustavson (1971)] which is an extension of Gear’s backward differentiation approach forms the basis of the parameter stepping technique. Brayton’s paper also relates to electronic circuits and this paper will be discussed in connection with fixed-h stability.

The Chang-Hindmarsh-Madsen paper presents a simulation of chemical kinetics transport in the stratosphere. One is dealing here with a stiff partial differential system. The size of the Jacobian is very large (\( N \times N \) where \( 10^4 \leq N \leq 10^5 \)) because of the spatial discretization. The Newton update in this case involves a multidimensional PDE sparseness structure for the Jacobian so a block relaxation scheme is used. Figure 1 in their paper shows an unusual difficulty occurring in step size control.

Another paper involving the "Methods of Lines" integration of partial differential equations is that of Loeb-Scheissner. If one considers a linear PDE with constant coefficients such as \( u_t = u_{xx} \) then the method of lines yields a class of initial value systems of the form \( \dot{u} = Au \). The size and form of \( A \) depends on how one approximates the spatial derivatives and on the number of mesh points. The authors make a detailed study of the eigenvalue character of \( A \) and of the stiffness ratio \( \lambda_{\text{MAX}}/\lambda_{\text{MIN}} \) as \( \Delta x \) and the difference formulas vary.

*See [Brennan (1968)] for a survey of CSMP-like simulation.
There has been very little research relative to $t \to \infty$ stability for variable step schemes. This is due to the inherent difficulties in the asymptotic analysis as $n \to \infty$ of linear difference equations with variable coefficients. Just as in the case of the asymptotic analysis of the smooth particular solution, one is faced with more than one parameter (e.g., the average step size $h$ and some measure of the variation of $h(\partial F/\partial u)$ in the neighborhood of the desired trajectory). A recent contribution in the area of variable-$h$, $t \to \infty$ stability is given in [Brayton and Conley (1972)]. It is hoped that more results of this type will be forthcoming in the future.

The research area of fixed-$h$ stability on the other hand is highly developed. In [Genin (1973)] certain basic concepts from electrical network theory are related in an elegant way to the algebraic aspects of fixed-$h$ stability, and the latter plays a basic role in the design of stiff system integration formulas. Fixed-$h$ stability analysis of an integration procedure consists of studying the nature of the dominant roots of a characteristic polynomial which arises when the procedure is applied to the scalar equation

$$\dot{z} + \lambda z = 0, \quad \text{Re}(\lambda) > 0. \quad (10a)$$

The exact recurrence associated with (10a) is

$$z(t + h) = e^{-qz(t)} \quad (10b)$$

where $q = \lambda h$. Since $|e^{-q}| < 1$ for Re($q$) > 0, one is interested in the region in the complex $q$-plane where the roots of the associated characteristic polynomial are in modulus less than one. This is the region of fixed-$h$ stability for the given method.

If one applies formula (4) with $u = z$ to (10a) then the resulting recurrence formula is $z = R(q)z$ where

$$R(q) = [1 + (1 - \mu)q + \xi q^2]^{-1}[1 - \mu q + \nu q^2]. \quad (10c)$$

If the region of fixed-$h$ stability includes Re($q$) > 0 then the formula is called $A$-stable [Dahlquist (1963A)]. In (10c) and (4) let

$$\mu = (1 - \alpha)/2, \quad \xi = (\beta + \alpha)/4, \quad \nu = (\beta - \alpha)/4,$$

then the resulting scheme is $A$-stable if and only if $\alpha, \beta \geq 0$. For the case $\xi = \nu = 0$ formula (4) has $0 \leq \mu \leq .5$ as the $A$-stability condition.

There are two weaker forms of global fixed-$h$ stability which are also important in the design of stiff integration procedures: (a) Stiff stability [Gear (1967), Computer Science Report, University of Illinois]; and (b) $A(\alpha)$-stability [Widlund (1967)]. All three of these types of global fixed-$h$ stability are characterized in the
Bickart-Rubin paper. They deal with a general class of composite multistep methods which are ultra-implicit. The methods involve \( n > 1 \) future points and \( m \) past points at each (composite) time step. Only \( k < n \) of the future points are retained to propagate the solution.

The Bickart-Rubin paper describes certain features in Rubin's PhD thesis under Bickart ["A-stability and composite multistep methods"; Syracuse University, New York (1973)]. In particular Rubin has developed a computer program which provides an exact yes-no test of A-stability for composite multistep methods with integer coefficients. The test is also extended to cover the case of stiff stability. The composite methods allow high order A-stable as well as stiffly stable methods to be synthesized.

The papers by Liniger-Gagnebin and by Enright also concern the synthesis of classes of A-stable and stiffly stable methods. Liniger has for a long time advocated fixed step approaches to stiff system integration. He has applied this idea in a novel way to achieve higher order approximate solutions from the combining of lower order approximate solutions in such a way as to knock out the leading error terms. In the usual Richardson extrapolation to the limit \( (h \to 0) \), one achieves the higher accuracy by using \( h \) and \( h/2 \) with the same formula and then combining these approximate solutions. An alternate procedure was proposed by Liniger and was systematically analyzed in a joint paper [Liniger and Odeh (1972)]. In this approach one deals with a multiparameter family of integration formulas each with the same step size and step number. The formulas are all A-stable and of second order for specified ranges of the parameters. The coefficients of the \((u, \dot{u}, \text{etc.})\) terms in the multistep formula are functions of one or more parameters. Several specific sets of parameter values are used to generate the low order approximate solutions and these are then combined to create a high order approximate solution. This more accurate solution is not propagated but is used for output purposes.

The Liniger-Gagnebin paper is an extension of the 1972 paper, and gives for an arbitrarily large \( k \), an explicit construction of a \((2k - 2)\)-parameter family of second order, A-stable \( k \)-step methods. The parameters in these formulas may be used, for example, for exponential fitting, to achieve strong damping of the stiff components, or to produce solutions of order \( p > 2 \) as weighted averages of second-order solutions.

Enright* has structured a class of variable order second derivative

---

methods. These are written below in a form which suggests the use of a modified Newton approach [Liniger and Willoughby (1970)] for solving the implicit difference system for \( u_{n+1} \) (note: \( Du = \dot{u} \)).

\[
P(hD)u_{n+1} - R_n = 0 \tag{11a}
\]

\[
P(Q) = |\gamma_0|Q^2 - |\beta_0|Q + 1 \tag{11b}
\]

\[
R_n = u_n + h \sum_{s=1}^{k} \beta_s u_{n+1-s} \tag{11c}
\]

Two schemes for determining the coefficients \( \gamma_0, \beta_0, \beta_1, \ldots, \beta_k \) are presented together with tables of their exact values and a graph of the stability region for each \( k \). The two criteria other than stiff stability which were involved in the determination of the parameters are: (a) Maximum Order; and (b) Factorization of \( P(Q) \). Here one has

\[
Q = hJ = h(\partial F/\partial u), \quad u = F(u).
\]

For the first class of methods

\[
P(Q) = |\gamma_0|(Q - rI)(Q - \bar{rI}) \tag{12a}
\]

\[
P(Q)\Delta u = b \tag{12b}
\]

\[
u_+ = u_+ + \Delta u \quad \text{(Newton Update).} \tag{12c}
\]

\( Q \) is a real matrix and \( b \) is a real vector, but \( r \) is a complex scalar and \( \bar{r} = \text{complex conjugate of } r \). Thus the solution to (12b) is

\[
\Delta u = \text{Im}(w)/\text{Im}(r) \tag{13a}
\]

where \( w \) is the complex vector which is the solution to

\[
|\gamma_0|(Q - rI)w = b. \tag{13b}
\]

In the second class of methods the condition \( |\beta_0|^2 = 4|\gamma_0| \) is imposed and thus \( P(Q) \) is a perfect square (i.e. \( r = \bar{r} = 2/|\beta_0| \)).

Methods such as Gear's backward differentiation approach only require for each Newton update the solution of (13b) for \( w \) where \( r \) is real and \( \Delta u = w \). Since there is roughly twice the computational complexity for each Newton update for the second derivative methods, one wants to be able to have an average step size which is at least twice that used in Gear-like methods. Of course, such trade-offs are both problem dependent and accuracy dependent.

The paper by Brayton also deals with fixed-h stability, but in the more complicated context of difference-differential equations. Modeling lossless transmission lines as part of an electronic circuit forms a class of practical problems of this type. Certain variables
are evaluated not only at time $t$ in the system but also at $t - \tau_k$ where the $\tau_k$'s are specified delays which are independent of $t$. Since the times $t - \tau_k$ may occur at points which are not part of the $h$-grid, the integration procedure must also include an interpolation scheme.

The electrical engineering concept of passivity (i.e. of being dissipative) is introduced and is related to the algebra of fixed-$h$ stability. Passive interpolation is also defined using ideas developed in [Strang (1962)]. The main result in the Brayton paper is that if one combines an $A$-stable multistep formula with passive interpolation, then there is fixed-$h$ stability for all $h > 0$ provided the difference-differential system is strictly passive. The latter condition is somewhat stronger than that of differential stability as $t \to \infty$.

Convergence as $h \to 0$ for one step formulas and for linear multistep formulas has an excellent and basically complete theory associated with it. The early fundamental research is given in [Dahlquist (1956, 1959); Henrici (1962, 1963)], and this work has been refined and extended over the years [see e.g. Stetter (1973)].

Convergence theory for variable order-variable step schemes is being developed by Gear and his students as well as by a number of other researchers. The Gear-Tu-Watanabe paper is a summary of this work. The stiff system case using backward differentiation formulas is not considered in this paper, but many of the tools of analysis which are used also apply to the stiff case. Recall that for stiff systems one has one (or more) stiffness parameter $\varepsilon$ and that the assumption involved in the design of efficient stiff system integration formulas is that $0 < \varepsilon \ll h$.

Multistep formulas can be analyzed via the one step framework by introducing composite vectors [Richtmyer and Morton (1967)]. The error propagation formula then assumes the simple form

$$
\varepsilon_{n+1} = T_n \varepsilon_n + d_n
$$

(14)

where $T_n$ is the error propagation matrix. If one can show that

$$
||d_n|| = o(h_n)
$$

(15a)

and that for

$$
\sigma(m,n) = ||S(m,n)||, \quad S(m,n) = T_{m-1} \cdots T_{n+1} T_n,
$$

(15b)

$\sigma(m,n)$ is uniformly bounded for all $m > n$, then one has

$$
||\varepsilon_N|| \to 0 \text{ for all } t_N \in [0,T] \text{ as } \max(h_n) \to 0, \quad h_n = t_{n+1} - t_n.
$$
Both the variable step technique and the interpolation technique [Nordsieck (1962)] are analyzed in the above context. Basically the results are: (a) For interpolation methods each step size change should be followed by a number of fixed $h$ steps—the number being comparable to the order of the formula; and (b) For variable step methods it is sufficient to have $(h_{n+1}/h_n) = 1 + O(h)$ where $\text{MAX}(h_n) \leq h$ and $h \to 0$.

If one imposes the periodicity condition $T_{n+p} = T_n$ for fixed $p > 1$ and for all $n \geq 0$, then one can analyze $t \to \infty$ stability of (14) via the spectral radius of $S' = S(p,0)$. Instability has been exhibited by Tu in his PhD thesis under Gear for the fourth order Adam's method with interpolation and with $h_{2n} = 10h_{2n+1} = h_{2n+2}$ as the step size rule (i.e. $p = 2$, $S' = T_{1T_0}$).

Lambert presents in his paper two classes of matricial schemes which he calls unconventional. The motivation for the study is the reduction of the computational complexity associated with the use of Newton's method at each time step. A number of generalizations of fixed-$h$ stability are defined to provide analysis tools for dealing with the nonscalar nature of the schemes.

In the first class of methods the formulas are only linearly implicit so exactly one system of the form (13b) with $r$ real has to be solved at each time step. In performance these schemes are like the stiffly stable linear multistep methods with a single Newton update at each time step.

The second Lambert class of formulas are nonlinear, and this means that there is a quite different stability situation in the system case than in the scalar case. This difference is nontrivial even for constant coefficient systems $\dot{u} = Au$. Also this second class of schemes does not have a global concept of order in the usual sense of truncation error as $h \to 0$ via an expansion in powers of $h$.

Exponentials play a fundamental role in the constant coefficient system. In fact, one can write

$$\dot{u} + Au = G(t,u)$$

in the form

$$u(t+h) = Mu(t) + h \int_0^1 M^{1-\theta} K(\theta) d\theta$$

where $M = \exp(-Ah)$ and $K(\theta) = G(t+\theta h, u(t+\theta h))$. No papers at the conference were directly concerned with computational use of the matrix $M$, but there are a number of articles in the literature that deal with the case where $A$ is a diagonal matrix [see e.g. Flatt (1968); Certaine (1960); E. F. Sarkany and W. Liniger, "Exponential fitting of matricial multistep methods for ordinary differential equations," to appear in Math. Comp.]. A special method involving
local exponential fitting at the component level was developed for certain classes of electronic circuits [Fowler and Warten (1967); Sedore (1967); Bowers and Sedore (1971)]. The reader should also consult the Subject Index heading Exponentials for other references.

There is a large amount of special purpose simulation in which the integration method is designed in a way analogous to the design of linear time invariant control systems. That is, the system is represented by a flow diagram with transfer function boxes and feedback loops. Special purpose difference equations are generated from this diagram with the aid of root locus analysis [see e. g. Fowler (1963, 1965)]. This piecing together of special purpose formulas can be very effective if designed by someone familiar with both the mathematics and the details of the physical situation. However, it really represents an ad hoc problem-by-problem approach [Daniel (1966)]. The journal Simulation contains many articles devoted to special purpose simulation techniques.

As the articles in this proceedings clearly demonstrate, there has been excellent progress in the field of stiff system research. There are a number of open questions and certain areas still need further development. It is hoped that Stetter will soon extend the methods of economical global error estimation to include the stiff system case. Gear and his colleagues are actively working towards a theory of convergence for variable order-variable step methods for stiff systems.

Some of the basic topics which need further research are

(a) Multiparameter Asymptotics;

(b) Variable Order-Variable Step $t \to \infty$ Stability;

(c) Local Estimations of variability, exponential growth and decay, and error to provide better local control on choice of formula and step size;

(d) Noisy and/or Unsmooth Systems and Solutions including dynamically induced boundary layer transitions.

Adaptive control of the choice of formula and step size can be enhanced in a number of ways. For example, let $\hat{u}(t)$ be a smooth interpolant for the problem

$$
\dot{u} = F(u, p), \quad u(t_0) = u_0, \quad t_0 \leq t \leq t_N
$$

where $p$ is a parameter vector. Then the numerical solution of the nearby problem
\[
\dot{v} = F(v, p+\Delta p), \quad v(t_0) = u_0 + \Delta u_0, \quad t_0 \leq t \leq t_N
\]

should have a more efficient adaptive control than the \( \dot{u} \) problem provided the change in \( p \) and \( u_0 \) does not cause drastic qualitative changes in the \( v \)-trajectory. An area of research which could be very useful for the design of algorithms is the qualitative theory for ordinary differential equations [see e. g. C. Conley, "The gradient structure of a flow: I," IBM Research Report RC3932 (1972)].

In the neighborhood of the smooth particular solution the direction of the tangent vector \( \dot{u} \) can be a sensitive function of position. This can also mean severe numerical cancellation in the calculation of \( F(u) \) and of \( J(u) \). The normal assumption is that round-off error is negligible compared to truncation error, but an analysis should be made of the role of numerical cancellation and of round-off error for stiff system integration methods.

The subject of nonlinear oscillations was not a part of the symposium and the field has in general developed quite independently of the research area of the numerical initial value problem. One could have stiffness and a limit cycle behavior combined.

Difference equations can be fitted to decaying exponentials and also to sinusoidal oscillations where the frequency is known a priori. The reader should check the Subject List heading Trigonometric Methods to find references in the main bibliography. See also [Zaudunisky (1970)] for a discussion of accuracy of orbit calculations.


This proceedings is dedicated to the late Dr. Sullivan G. Campbell, who had a keen insight into the whole spectrum of Computer and Information Sciences. He has left the professionals in these fields a challenge for the future in his summation lecture, "The Generation Gap," at a UCLA Symposium [pp. 169-177 in Fourth Generation Computers: User Requirements and Transition, F. Gruenberger (Editor), Prentice-Hall, Englewood Cliffs, New Jersey].
A-stability [Dahlquist (1963A)] and its weaker associates, **
A[\alpha]-stability [Widlund (1967A)] and stiff stability [Gear(1971D)],
have become generally accepted as appropriate properties of numerical methods suitable for solving a stiff initial value problem, as described by a first order vector ordinary differential equation
\[ \dot{x}(t) = f(x(t), t) \]  \hspace{1cm} (1)
with initial condition
\[ x(t_0) = x_0. \]  \hspace{1cm} (2)
The Dahlquist bound of two on the order of A-stable multistep methods was the imperative to propound the above cited weaker stability properties, which are not subject to the Dahlquist bound. An alternative approach for circumventing Dahlquist's bound is to modify the class of methods, rather than the property. One suitable class of methods, which subsumes the class of multistep methods, consists of the composite multistep methods [Sloate and

* The research reported herein was supported in part by the National Science Foundation under Grant GK-23010.

Bickart (1973A)]. This larger class also subsumes the class of
composite one-step methods, which, as reported in [Watts and
Shampine (1972A)] and [Bickart, Burgess, and Sloate (1971A)]*,
contains A-stable methods of order greater than two.

A test to establish the A-stability property was developed by
the second author [Rubin (1973A)]*. Though motivated by a
study of composite multistep methods, the test is applicable to
a much larger class of methods. The significant feature of this
test is that it yields, after a finite number of arithmetic oper­
ations, a definitive answerto the question: Is the tested method
A-stable?

Herein, a precise definition of a composite multistep method
will be supplied and, with respect to the characteristic polynomial
—a polynomial in two variables—associated with the method, the
A-stability test will be presented and illustrated. An additional
and new result being described is a test for stiff stability which
is based on the test for A-stability. The task that must be faced
in devising a test for A[a]-stability will be discussed. The use
of the tests to establish new composite multistep methods, each
with an acceptable stability property, is examined. This applica­
tion of the tests invokes a parameterization, to be described, of
a composite multistep method and of its characteristic polynomial.

As will be evident, stability properties of importance in
solving a stiff equation are examined in terms of the characteris­
tic polynomial. In particular the characteristic polynomial is
the basis for dividing the treatment of stability into two parts.
In part one, a criterion on the characteristic polynomial is estab­
lished and shown to reflect the stability property sought in a
method. In part two, attributes of the characteristic polynomial
needed to satisfy the criterion created in part one are devised.
As a characteristic polynomial can be defined for most methods, not
just the composite multistep methods, this division of the stabil­
ity problem has universality. Furthermore, the attributes test
of part two is then immediately applicable where other methods are
concerned.

COMPOSITE MULTISTEP METHODS

Suppose the differential equation (1) with the initial con­
dition (2) possesses a unique solution \( \chi(t) \) for \( t \in [t_0, \infty) \). Now, the
task in numerical solution of (1) with (2) is: Compute a monotonic
sequence \( \{t_i : t_i \in [t_0, \infty) \text{ for } i=0,1,2,... \text{ and } t_i \to \infty \text{ as } i \to \infty \} \) and a

*See end of paper for the following references: [Bickart, Burgess,
and Sloate (1971A)], [Rubin (1973)], and [Sloate (1971)].
vector-valued sequence \( \{x_i\} \) such that \( x_i \) approximates \( \chi(t_i) \).

The class of methods herein considered for computing \( \{x_i\} \) with \( \{t_i\} = \{t_0 + ih : h > 0\} \) is the class of composite multistep methods. Now, a composite multistep method is defined as an ordered pair \((R, k)\), where \( R \)—the composite matrix—is an \( n \times 2(m+n) \) matrix of real numbers \( \alpha_{ij} \) and \( \beta_{ij} \),

\[
R = \begin{bmatrix}
\alpha_{10} & \cdots & \alpha_{1(m+n-1)} & \beta_{10} & \cdots & \beta_{1(m+n-1)} \\
\vdots & & \vdots & \vdots & & \vdots \\
\alpha_{m0} & \cdots & \alpha_{m(m+n-1)} & \beta_{m0} & \cdots & \beta_{m(m+n-1)}
\end{bmatrix}
\]

and \( k \), the number of retained points, is a positive integer not greater than \( n \). The positive integers \( m \) and \( n \) are, respectively, the number of past points and the number of future points. The sequence \( \{x_i\} \) is generated by \((R, k)\) for a given constant step-size \( h \) as follows: Put \( x_0 = x_0 \), and compute \( x_1, \ldots, x_{m-1} \) using an external starting procedure. Set \( y_i = x_i \) for \( i = 0, \ldots, m-1 \). With \( \ell = 0 \) solve the \( n \) algebraic equations

\[
\sum_{j=0}^{m+n-1} \alpha_{ij} y_j - h \sum_{j=0}^{m+n-1} \beta_{ij} f[y_j, t_0 + (k\ell + j)h] = 0
\]

for \( y_m, \ldots, y_{m+n-1} \); retain the first \( k \) elements of this result as the values of \( x_m, \ldots, x_{m+k-1} \), discarding the remaining \( n-k \). The solution is thus advanced \( k \) points at the 0-th iteration. In general, the \( \ell \)-th step of the iteration, \( \ell = 0, 1, 2, \ldots \), proceeds as follows:

1. Set \( y_j = x_{k\ell + j} \), \( j = 0, \ldots, m-1 \)
2. Solve (3) for \( y_j \), \( j = m, \ldots, m+n-1 \)
3. Set \( x_{k\ell + j} = y_j \), \( j = m, \ldots, m+k-1 \)

The sequence \( \{x_i\} \) is thus advanced a "block" of \( k \) points at each iteration by using the \( n \) equations in \( m \) past points to solve for \( n \) future points, but retaining only the first \( k \).

Now consider the case in which (1) is the scalar, linear equation

\[
\dot{\chi}(t) = q\chi(t).
\]

A careful examination will disclose that (3) then possesses a unique solution for \( y_m, \ldots, y_{m+n-1} \) if and only if, with \( \lambda = qh \),

\[
\delta(\lambda) = \det \{A_f - \lambda B_f\}
\]

is not zero, where \( A_f \) and \( B_f \) are submatrices of \( R \) partitioned as

\[
R = [A_p \ A_f \ B_p \ B_f],
\]
with $A$ and $B$ being the $n \times m$ matrices of "past" $\alpha$'s and $\beta$'s and with $A_p^f$ and $B_p^f$ being the $n \times n$ matrices of "future" $\alpha$'s and $\beta$'s. This implies the method possesses a unique solution for $x_{k+m+1}$, $x_{k+2m+1}$, ..., $x_{k+m+k-1}$ if $\delta(\lambda) \neq 0$. Thus the zeros of $\delta$, to be known as the poles of the method, are of importance. For future use, let $\Lambda$ denote the set of poles; that is, let

$$\Lambda = \{\lambda : \delta(\lambda) = 0\}.$$ 

Note: In all cases of importance $\Lambda$ contains $n$ or fewer points.

Without giving formal expression to the fact, it is evident that the method applied to (4) defines a linear difference equation for $\{x_i\}$, which for $\lambda \notin \Lambda$ is unique. The eigenvalues $\zeta$, which depend on $\lambda$, of this difference equation determine the asymptotic behavior of the solution $\{x_i\}$. In particular, if for a given $\lambda \notin \Lambda$ the eigenvalues $\zeta$ are all contained in the open unit disk, then $\{x_i\}$ approaches the origin asymptotically for all initial conditions (2) and all starting procedures. Under these conditions the solution $\{x_i\}$ is said to be stable.

The eigenvalues $\zeta$ for any given $\lambda$ are the zeros of $P(\lambda, \cdot)$, where $P$, a polynomial in the two variables $\lambda, \zeta$, is called the characteristic polynomial and expressed as:

$$P(\lambda, \zeta) = \det \{R L(\lambda) Z(\zeta)\},$$ \hspace{1cm} (5)

with

$$L(\lambda) = \begin{bmatrix} I_{m+n} \\ -\lambda I_{m+n} \end{bmatrix}$$

and

$$Z(\zeta) = \begin{bmatrix} I_k & \zeta I_k & \cdots & \zeta^{M-1} I_k & \zeta^M J & 0 \\ 0 & 0 & \cdots & 0 & 0 & I_{n-k} \end{bmatrix}^T,$$

where

$$J = \begin{bmatrix} I_{m-k(M-1)} \\ 0 \end{bmatrix}$$

and $M$ is the smallest integer such that $M \geq m/k$ [Rubin (1973A); pp. 25-26, 82].

Thus, stability questions concerning $(R, k)$ can be referred to the characteristic polynomial $P$ of (5). That is, for given values of $q$ and $h$, with $qh = \lambda \notin \Lambda$, $\{x_i\}$ is stable if and only if all zeros of $P(\lambda, \cdot)$ lie in the open unit disk. This result can be stated directly in terms of a set of $\lambda$ values as follows:
THEOREM 1: Let \( P \) be a subset of the complex plane, with \( \Lambda \cap P \) empty. The solution \( \{x_i\} \) corresponding to \((R,k)\) is stable for all complex \( q \) and positive \( h \), with \( qh = \lambda \in P \), if and only if all zeros of \( P(\lambda,\cdot) \) lie in the open unit disk for all \( \lambda \in P \).

A-STABILITY

The A-stability property focuses on the asymptotic behavior of the sequence \( \{x_i\} \) approximating the solution of the linear differential equation (4), as set forth in

DEFINITION 1: Let \( L \) denote the open left half plane. Then, a method* is said to be A-stable if in solving (4), for every fixed positive step-size \( h \), the solution \( \{x_i\} \) is stable whenever \( qh = \lambda \in L \).

Clearly, a composite multistep method is A-stable when the conditions of THEOREM 1 hold with \( P = L \). Observe that the critical fact of the theorem is the necessary and sufficient condition on the characteristic polynomial. This observation motivates

DEFINITION 2: A polynomial \( P \) in \((\lambda,\zeta)\) is said to satisfy the A-stability criterion if for every \( \lambda \in L \) the zeros of \( P(\lambda,\cdot) \) lie in the open unit disk.

The focus is now on the A-stability criterion for which the second author has derived an algebraic test based on a transformed polynomial \( P' \) and auxiliary polynomials \( \nabla_i \) defined as follows [Rubin (1973A); Chap.3]: Let

\[
P'(\omega, z) = (z-j)^m P(j\omega, z+j),
\]

where \( m \) is the degree of \( P \) in \( \zeta \). Let \( m' \) be the degree of \( P' \) in \( z \). \( P' \) is a polynomial in two variables with complex coefficients; it can thus be written in the form

\[
P'(\omega, z) = [1 \ j] \tau(\omega) \ [1 \ z \ldots \ z^{m'}]^T
\]

where \( \tau \) is a \( 2 \times (m' + 1) \) real polynomial matrix. From \( \tau \), define the \( m' \times m' \) Bezoutian matrix \( C \), whose elements \( c_{ij} \) are sums of \( 2 \times 2 \) minors of \( \tau \), as follows:

\[
y^{m'-\max\{i,j\}} \sum_{s=\max\{0,m'+1-i-j\}}^{2m'+1-i-j-s} \tau^{1,2}_{2m'+1-i-j-s,s}.
\]

Let \( \nabla_i \), \( i=1,\ldots,m' \), be the nested leading principal minors of \( C \); that is,

\[
\nabla_i = C(1,2,\ldots,i)\begin{vmatrix} 1,2,\ldots,i \end{vmatrix}.
\]

* That is, a numerical method for the solution of ordinary differential equations.
Each \( \mathcal{V}'_i \) can be shown to be an even polynomial in \( \omega \) of maximum degree \( 2ni \). Let \( \mathcal{V}''_i \) be the polynomial in \( \omega \), of maximum degree \( ni \), formed from the even coefficients of \( \mathcal{V}'_i \). The auxiliary polynomial \( \mathcal{V}_i \) is \( \mathcal{V}''_i \) after removing any zero at the origin.

Now, the test for the A-stability criterion can be expressed as in

THEOREM 2: The polynomial \( \mathcal{P}_i \), with \( \mathcal{V}_m \) not identically zero, satisfies the A-stability criterion if and only if

(a) all zeros of \( \mathcal{P}(\cdot, \infty) \) lie in the closed right half plane,
(b) \( m' = m \),
(c) \( \mathcal{V}_i(0) > 0 \) for all \( i = 1, 2, \ldots, m' \), and
(d) \( \mathcal{V}_i \) has no positive real zeros of odd multiplicity, for all \( i = 1, 2, \ldots, m' \).

Note that (a) is easily checked with the Hurwitz criterion or one of its equivalents, (b) and (c) are trivial to verify, and (d) may be confirmed by methods based on the classical results of Sturm. An even easier test is imposed by the following alternative to condition (d):

\[(d') \mathcal{V}_m \text{ has no positive real zeros.}\]

As modified the resulting theorem provides sufficient conditions which are "almost" necessary for A-stability. In fact, if, in addition, condition (a) is strengthened somewhat by replacing "closed" by "open," then the conditions of the modified theorem are necessary and sufficient for strong A-stability [Rubin (1973A); pp. 63-67]. In this sense is the term almost necessary used.

To illustrate the A-stability test, consider the 5-th order composite multistep method (R,2) with

\[
R = \begin{bmatrix}
53280 & -149040 & 95760 & 0 & 0 \\
2960 & -8280 & -3060 & 400 & 0 \\
& -3060 & 400 & 0 & 0 & 2660 \\
& & -21101 & -1786 & 80184 & -17686 & 2869 \\
& & -1091 & -646 & 7824 & 1574 & 19 \\
& & & 967 & 4142 & 1272 & 3842 & 817
\end{bmatrix}.
\]

* This hypothesis, with \( \mathcal{V}_m \) not identically zero, excludes only pathological cases which are of little interest in the present context. A completely general treatment is given in [Rubin (1973A)].

Note that \( n=3 \) and \( m=2 \). By (5) the characteristic polynomial is

\[
P(\lambda, \xi) = [1 \ \lambda \ \lambda^2 \ \lambda^3] \begin{bmatrix} 0 & -15960 & 15960 \\ 1252 & -10184 & -22988 \\ 727 & 2300 & 13533 \\ 0 & 0 & -3420 \end{bmatrix} \begin{bmatrix} \xi \\ \xi^2 \end{bmatrix}.
\]

By the Hurwitz criterion

\[
P(\lambda, \omega) = 15960 - 22988\lambda + 13533\lambda^2 - 3420\lambda^3
\]

has all its zeros \( \lambda \) in the closed right half plane. Thus (a) of THEOREM 2 is satisfied. Using (6), \( m'=m=2 \); therefore, (b) is established. By way of (7), (8), and (9) the auxiliary polynomials can be computed to be

\[
V_1(\omega) = 37240 + 1462\omega + 855\omega^2
\]

\[
V_2(\omega) = 38580640 + 23002112\omega + 25571970\omega^2 + 6579225\omega^3.
\]

Note that \( V_1 = V_2 \) is not the zero polynomial. Now, \( V_1(0)>0 \) and \( V_2(0)>0 \); thus, condition (c) is verified. By inspection \( V_1 \) and \( V_2 \) have no positive real zeros; hence, condition (d) — also condition (d') — is satisfied. It follows that \( P \) satisfies the \( A \)-stability criterion. Furthermore, it can be shown that the poles of \( R \) are exactly the zeros of the polynomial (10), which are in the closed right half plane. Therefore, this composite multistep method is \( A \)-stable by THEOREM 1 with \( P=L \).

**STIFF STABILITY**

The weaker stiff stability property also focuses on the asymptotic behavior of the approximating sequence, but not throughout the entire open left half plane as was the case for \( A \)-stability. To give this statement precision, consider

**DEFINITION 3:** Let \( S \) be a connected open subset of the left half plane which contains a half plane and has the origin as a limit point. Then, a method is said to be stiffly stable relative to \( S \) if in solving (4), for every fixed positive step-size \( h \), the solution \( \{x_i\} \) is stable whenever \( \omega h = \lambda \in S \).

One possible domain \( S \) is shown in Figure 1; it consists of the union of an open half plane and three open disks.

Just as in the case of \( A \)-stability, THEOREM 1 is invoked, here to say: A composite multistep method is stiffly stable relative to \( S \) when THEOREM 1 holds with \( P=S \). Since \( S \), as in the illustration, may be composed as the union of elementary domains \( P \), the extracted and key condition on the characteristic polynomial is as expressed in
DEFINITION 4: A polynomial \( P \) in \((\lambda, \xi)\) is said to satisfy the \( P \)-stability criterion if for every \( \lambda \in P \), the zeros of \( P(\lambda, \cdot) \) lie in the open unit disk.

An algebraic characterization for stiff stability relative to a domain such as is illustrated in Figure 1 can be constructed by applying linear and bilinear transformations to the characteristic polynomial and then applying the algebraic characterization of the previous section. Thus, consider the linear transformation

\[
\lambda = \sigma + \eta,
\]

which, for real values of the parameter \( \sigma \), conformally maps the open left half \( \eta \)-plane onto the open half \( \lambda \)-plane \( H = \{ \lambda : \text{Re } \lambda < \sigma \} \).

Letting

\[
H(\eta, \xi) = P(\sigma + \eta, \xi),
\]

it is obvious that \( P \) satisfies the \( H \)-stability criterion if and only if \( H \) satisfies the \( A \)-stability criterion.

Similarly, consider the bilinear transformation

\[
\lambda = \frac{(\gamma + \rho) - (\gamma - \rho) \eta}{1 - \eta},
\]

which, for real \( \gamma \) and positive real \( \rho \), conformally maps the open left half \( \eta \)-plane onto the open disk \( D \) with center \( \gamma \) and radius \( \rho \).
in the $\lambda$-plane. Define the polynomial $D$, in two variables, by

$$D(\eta, \zeta) = (1-\eta)^n p \left( \frac{(\gamma+\rho)-(\gamma-\rho)n}{1-\eta}, \zeta \right),$$

(12)

where $n$ is the degree of $P$ in $\lambda$. As before, $P$ satisfies the $D$-stability criterion if and only if $D$ satisfies the $A$-stability criterion.

The implication of the above developed facts is made explicit in

**THEOREM 3:** Let $S$ be the union of an open half plane $H = \{\lambda: \text{Re } \lambda < 0\}$ with a finite number of open disks $D_i$, $i=1, \ldots, d$, centered on the negative real axis. Let $H_i$ be the polynomial defined as in (11) [as in (12)] and associated with $H$ [with $D_i$]. Then $P$ satisfies the $S$-stability criterion if and only if $H$ and $D_i$, $i=1, \ldots, d$, all satisfy the $A$-stability criterion.

To illustrate this domain dependent stiff stability test, consider the 3-rd order backward differentiation method $(R, 1)$, where

$$R = [-2 \ 9 \ -18 \ 11 \ 0 \ 0 \ 0 \ 6].$$

Note that $n=1$ and $m=3$. Now, it must be determined whether

$$P(\lambda, \zeta) = \begin{bmatrix} 1 & \lambda \end{bmatrix} \begin{bmatrix} 2 & -9 & 18 & -11 \ 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ \zeta \\ \zeta^2 \\ \zeta^3 \end{bmatrix}$$

satisfies the $S$-stability criterion, with $S$ here taken to be the union of the open half plane $H = \{\lambda: \text{Re } \lambda < -0.084\}$ and of the open disk centered at $-7$ with radius $7$. Equivalently, it must be determined whether

$$H(\eta, \zeta) = \begin{bmatrix} 1 & \eta \end{bmatrix} \begin{bmatrix} 250 & -1125 & 2250 & -1438 \ 0 & 0 & 0 & 750 \end{bmatrix} \begin{bmatrix} 1 \\ \zeta \\ \zeta^2 \\ \zeta^3 \end{bmatrix},$$

created as in (11) with $\sigma = -0.084$, and

*As for the $A$-stability test, there is an APL implementation of these computations. The additions to the APL program for the $A$-Stability test needed to realize the stiff stability test are described in: Rubin, W.B. and Bickart, T.A., "APL Algorithms for a Stiff Stability Test on Composite Multistep Methods," Memorandum to appear, Elec. and Comp. Engr. Dept., Syracuse University.*
\[ D(n,\zeta) = [1 \eta] \begin{bmatrix} 2 & -9 & 18 & -11 \\ -2 & 9 & -18 & 95 \end{bmatrix} \begin{bmatrix} 1 \\ \zeta \\ \zeta^2 \\ \zeta^3 \end{bmatrix}, \]

created as in (12) with \( \gamma = -7 \) and \( \rho = 7 \), both satisfy the A-stability criterion. The fact that they do can be verified as in the previous illustration. Lastly, to verify that \((R,1)\) is stiffly stable relative to \( S \), THEOREM 1 requires that \( \Lambda \cap S \) be shown to be empty. This fact is established as follows: The poles of this method are the zeros of

\[ \delta(\lambda) = -11 + 6\lambda \]

which by inspection are in the closed right half plane; therefore, \( \Lambda \cap S \) is empty.

To illuminate this result, the domain \( S \) is shown in Figure 2 together with the actual region within which the solution \( \{x_i\} \) is stable. Note: That region is bounded by the locus of points \( \lambda \) such that \( P(\lambda,\zeta) \mid \zeta = 1 = 0 \).

Figure 2. Stiff Stability Domain for 3-rd Order Backward Differentiation Method
Of considerable significance is the fact that the algebraic tests for the A-stability and S-stability criteria are important tools in the quest for A-stable and stiffly stable methods. Classes of methods and their characteristic polynomials can be parameterized, and the algebraic test applied as a function of these parameters. Solving the resultant sets of inequalities will yield particular methods with guaranteed stability properties. The nature of the method and characteristic polynomial parameterizations is presented in this section for composite multistep methods.

Consider the class $R_1(p,n,m)$ of composite multistep methods with elements of $R$ being integers, with non-singular $A_1$, with $n$ future and $m$ past points, and of order $p$. These restrictions on $R$ and $A_1$ can be shown to exclude no cases of potential interest. The following facts are proved in [Rubin (1973A); pp.104-111]: With $\mu = 2m + n-p-1$ assumed to be non-negative, there exists a $\mu \times 2(m+n)$ matrix $S$ of integers partitioned after the $m$-th and $(m+n)$-th columns as

$$S = [S_p S_f S_B],$$

such that

- $S_f = 0$
- $\text{rank } S = \mu$
- $S \Pi (m+n)p = 0,$

where

$$\Pi = [\pi_{i0} \pi_{i1} \cdots \pi_{ip}],$$

with the $\pi_{ij}$ being $2i \times 1$ matrices, defined as follows:

$$\pi_{i0} = [1 1 \cdots 1 0 \cdots 0]^T,$$

$$\pi_{ij} = [j^i \cdots (i-1)^i \cdots -j \cdots -j(i-1)^{i-1}]^T, \quad j=1,\ldots,p.$$  

It can then be established that, if $(R,k) \in R_1(p,n,m)$, then $(xR+XS,k) \in R_1(p,n,m)$, where $x$ is a non-zero integer and $X$ is an $n \times \mu$ matrix of integers. In fact, every $(R,k) \in R_1(p,n,m)$ can be expressed in this manner. That is, every element of the class $R_1(p,n,m)$ can be represented in the parameterized form $(xR + XS,k)$, where $R$ and $S$ are constant matrices, and $X$ is a matrix of free parameters. The scalar $x$ acts as a normalizing parameter.

The characteristic polynomial (5) associated with $(xR + XS,k)$ can be expressed in parametric form as
THEODORE A. BICKART AND WILLIAM B. RUBIN

\[ p = \sum_{j_1} x_{j_1} \left( \begin{array}{cccc} i_1', & i_2', & \ldots, & i_{n-j_1}' \\ i_1+j_1-n, & i_2+j_1-n, & \ldots, & i_n-n \end{array} \right) p_1, \]

where summation is over all integral sequences \( i = i_1, i_2, \ldots, i_n \) satisfying \( 1 \leq i_1 < i_2 < \ldots < i_n \leq n + \mu \), where \( \mu = \max \{0, n - \mu\} \), \( j_1 \) denotes the largest integer such that \( i_1, i_2, \ldots, i_{j_1} \) is the complementary sequence* of \( i \), and where \( p_1 \) is the characteristic polynomial associated with \( (R_i k) \). \( R_i \) is the composite matrix consisting of rows \( i_1, i_2, \ldots, i_n \) of \( i[R_i] \).

EXAMPLES

It was conjectured in [Sloate and Bickart (1973A)]** that: The order of an A-stable composite multistep method cannot exceed \( 2n \). This generalization of the Dahlquist bound was further refined in [Bickart, Burgess, and Sloate (1971A)] to be: The order of an A-stable composite multistep method cannot exceed \( 2 \times \text{rank } B_i \). With \( p = 2n \) the conjecture was reached or, as might be the case, invalidated by example, a search for A-stable methods was conducted. No methods were found to violate the conjecture; on the other hand, only some methods from the subclass \( R_i(4,2,2) \) with \( k = 1 \) were found to reach the conjectured bound. Of the many methods examined, those which were found to be A-stable came from the subclasses \( R_i(4,2,2), R_i(5,3,2), R_i(6,4,2) \) and \( R_i(7,5,2) \). In the first case \( \mu = 0 \) and the characteristic polynomials for each value of \( k \)—namely, 1 thru 4—did not depend on any free parameters; in the other four cases \( \mu = 1 \) and the characteristic polynomials were linearly dependent on the free parameters.

To illustrate these discovered results, all of which are reported in [Rubin (1973A); pp. 111-125] the \( p = 2n \) method with \( k = n \) from \( R_i(4,2,2) \)*** having

\[
R = \begin{bmatrix} 0 & 24 & -24 & 0 & 1 & -13 & -13 \\ 56 & -72 & 0 & 16 & -21 & -39 & 33 \\ \end{bmatrix}
\]

and the \( k = n \) method with \( p < 2n \) from \( R_i(7,5,2) \) having

* The complementary sequence of \( i \) is the monotonically increasing sequence of all those integers from 1 through \( n + \mu \) not contained in \( i \).

** First reported at 1971 SIAM National Meeting.

*** This method along with its implementation, was first reported in [Sloate (1971A)].
were found to be A-stable. Their actual stability region boundaries are illustrated in Figure 3. The first method is significant because it reaches the conjectured maximum order; the second is significant because the number of retained points is maximum, an important factor in creating an efficient numerical integration process based on the method.

The illustration of the previous section is of the p = 3 backward differentiation method. To further illustrate the stiff stability test reported there, consider the p = 4 backward differentiation method (R,1) with

\[ R = \begin{bmatrix} 60480 & -181440 & 120960 & 0 & 0 & 0 & 0 \\ 0 & -3780 & 0 & 3780 & 0 & 0 & 0 \\ -83160 & -280000 & 0 & 0 & 363160 & 0 & 0 \\ -3592512 & -12096000 & 0 & 0 & 0 & 15688512 & 0 \\ 31752000 & -63129024 & 0 & 0 & 0 & 0 & 31377024 \\ -20813 & -14856 & 140439 & -70016 & 34809 & -10488 & 1405 \\ -37 & 1398 & 4863 & 1328 & 33 & -30 & 5 \\ 21543 & 233496 & 362667 & 404352 & 160437 & -10776 & 921 \\ 1000955 & 9545400 & 17560575 & 13251200 & 19119825 & 6052680 & -184075 \\ -10734025 & -19553640 & 5454725 & 21814400 & 27147525 & 41821800 & 1089785 \end{bmatrix} \]

The actual stability region boundary is shown in Figure 4, together
with the constructed domain in relation to which the method can be shown to be stiffly stable according to the test.

![Diagram of Stiff Stability Domain](image)

**FIGURE 4.** Stiff Stability Domain For 4-th Order Backward Differentiation Method

A[\alpha] - STABILITY

In the introduction, the A[\alpha]-stability property, as well as the stiff stability property, was cited as a weaker associate of the A-stability property. To give proper attention to A[\alpha]-stability first consider

**DEFINITION 5:** Let \( W \) be the open (wedge shaped) domain 
\( \{ \lambda: -\alpha < \arg(-\lambda) < \alpha, 0 < \alpha \leq \pi/2 \} \). Then, a method is said to be A[\alpha]-stable if in solving (4), for every fixed positive step-size \( h \), the solution \( \{ x_n \} \) is stable whenever \( qh = \lambda \in W \).

It therefore follows that a composite multistep method is A[\alpha]-stable if THEOREM 1 holds with \( P = W \). Note that the key condition of the theorem is, in light of DEFINITION 4: The characteristic polynomial must satisfy the \( W \)-stability criterion.

Observe, as illustrated in Figure 5, that \( W \) is the union over \( \gamma \in (-\infty, 0) \) of the open disks

\[
P(\gamma) = \{ \lambda: |\lambda - \gamma| < \rho, \rho = |\gamma| \sin \alpha, \gamma < 0 \};
\]

That is,

\[
\hat{W} = \bigcup_{\gamma \in (-\infty, 0)} P(\gamma).
\]
If \( D(\eta, \xi, \gamma) \) denotes the polynomial obtained as in (12) with 
\( \rho = |\gamma| \sin \alpha \), then it follows from the results of that section that:

**THEOREM 4:** \( P \) satisfies the \( \mathcal{W} \)-stability criterion if and only if \( D(\cdot, \cdot, \gamma) \) satisfies the \( A \)-stability criterion for all \( \gamma \in (-\infty, 0) \).

This means that each of the elements (a) thru (d) of the \( A \)-stability criterion test must hold for all \( \gamma \in (-\infty, 0) \). For example, (c) becomes: 
\( \nabla_i(0, \gamma) > 0 \) for all \( i=1, \ldots, m' \) and all \( \gamma \in (-\infty, 0) \). In addition to verifying that \( \nabla_i(0, \gamma) > 0 \) for all \( i=1, \ldots, m' \) and some one \( \gamma \in (-\infty, 0) \), each \( \nabla_i(0, \cdot) \) must be shown to have no negative real zero—a Sturm test must be invoked. It is anticipated that a test, similar to that for the \( A \)-stability criterion, can be constructed for this criterion.

**CONCLUDING DISCUSSION**

Herein a criterion on the characteristic polynomial of a numerical method for solution of the initial value problem was shown to reflect the stability properties of the method. The key result reported was the test for the \( A \)-stability criterion, as that test is also the primary ingredient of the reported test for stiff stability and is the basis for the described test for \( A[\alpha] \)-stability. Of significant practical value are the cited APL programs which implement the \( A \)-stability and stiff stability tests.

Parameterization of a class of methods as a step in the quest for methods with a specified stability property, and parameterization
of a stability domain in the search for a test for a given stability criterion, were introduced. Both situations resulted in a parameterization of the conditions of the test for the A-stability criterion. In the former case, the task is: Find values of the parameters for which the conditions hold. In the latter case, the task is: Validate the conditions for a given set of parameter values.

REFERENCES

1973


1971


NUMERICAL A-STABILITY FOR DIFFERENCE-DIFFERENTIAL SYSTEMS

R. K. Brayton, Mathematical Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

INTRODUCTION

In recent years, general purpose programs have been created for numerically integrating systems of ordinary differential equations. Many of these packages use A-stable or stiffly stable numerical methods. It is quite natural to attempt to extend the applicability of these packages to include delay terms. This allows, in electrical engineering, lossless transmission lines to be used and in economic models, it is quite natural to have delays. From a programming standpoint, there is little difficulty in adding these extra terms since one only has to store the past values of the delayed variables and interpolate.

Of course, the A-stable and stiffly stable methods were not designed to preserve stability for systems of difference-differential equations. It is the purpose of this paper to investigate this point. The final result hoped for, which we do not have yet (and which may not be true), would be that A-stable numerical methods applied to stable systems of difference-differential equations results in a system of difference equations which is stable for all time steps \( h \geq 0 \).

A reasonably general form for the equations to be studied is

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + B(y)(t-\tau) \\
y(t) &= C x(t) + D(y)(t-\tau)
\end{align*}
\]  

(1.1)

where \( x \in \mathbb{R}^n \), \( y, \tau \in \mathbb{R}^m \), \( A \in \mathbb{R}^{nxn} \), \( B, C \in \mathbb{R}^{nxm} \), \( D \in \mathbb{R}^{m \times m} \). The notation \( (y)(t-\tau) = (y_1(t-\tau_1), y_2(t-\tau_2), \ldots, y_m(t-\tau_m))^T \) is used here. Linear electrical networks with coupled lossless transmission lines
fall within this form [Brayton (1968)], [Ho (1973)]. One can view (1.1) as an input-output system with a linear delayed feedback term. The stability of (1.1) depends on the roots \( \{ S_r \} \) of its characteristic equation

\[
\det \begin{pmatrix}
  sI - A & -B e^{-sT} \\
  -C & I - De^{-sT}
\end{pmatrix} = 0
\] (1.2)

where \( T = \text{diag} (\tau_1, \ldots, \tau_m) \). If \( \text{Re}\{S_r\} \leq -\delta < 0 \) then (1.1) is stable (see [Brayton (1968)], theorem A4). However since there are generally an infinite number of roots of (1.2), it is possible even with \( \text{Re}\{S_r\} < 0 \), that the system (1.1) is unstable. If \( |S_r| \to \infty \) implies that \( \text{Re} S_r \to -\infty \), then the system is called retarded [Bellman and Cooke (1963)]. Generally (1.1) is neutral (see [Brayton (1968)], theorem A1), i.e. there are roots \( |S_r| \to \infty \) with \( |\text{Re} S_r| \) bounded. We shall use the term stable in this paper to refer to the fact that all the roots of the characteristic equation of a system (either difference, or difference-differential equations) lie in a left-half plane \( \text{Re}(s) \leq -\delta < 0 \). The content of this paper is a number of theorems that establish the stability of certain systems of difference-differential equations and their discrete analogs.

A linear multistep (LMS) formula used with (1.1) would result in the following system of difference equations

\[
\begin{align*}
  w(t) &= Ax(t) + B(y)(t-1) \\
  y(t) &= C^* x(t) + D(y)(t-1) \\
  0 &= \sum_{i=0}^{k} a_i x(t-ih) - h \sum_{i=0}^{k} \beta_i w(t-ih). \\
  t &\geq 0
\end{align*}
\] (1.3)

Since generally \( \tau_j \neq h \cdot k \), this is a difference equation for \( t \) continuous, not discrete. Interpolation for the \( y_j(t-\tau_j) \) terms would be required to make this a discrete time equation. In the last part of section III, we discuss passive interpolation and a stable method of interpolation is given.

The characteristic equation for (1.3) is

\[
\det \begin{pmatrix}
  -A & -B e^{-sT} & +I \\
  -C & I - De^{-sT} & 0 \\
  \rho(\text{sh})I & 0 & -h\sigma(\text{sh})I
\end{pmatrix} = 0
\] (1.4)

NUMERICAL A-STABILITY FOR DIFFERENCE-DIFFERENTIAL SYSTEMS

where \( \rho(sh) = \sum_{i=0}^{k} a_i e^{-ish} \), \( \sigma(sh) = \sum_{i=0}^{k} \beta_i e^{-ish} \).

A-stability of the multistep formula means that

\[
\text{Re} \left( \frac{\rho(z)}{\sigma(z)} \right) \geq 0 \quad \text{all } z, \text{ Re}(z) \geq 0 .
\]  

(1.5)

Eliminating \( w \) we can obtain the equivalent characteristic equation

\[
\det \left( \begin{array}{cc}
\frac{\rho(sh)}{\sigma(sh)} I - \rho & -Be^{-sT} \\
-C & I - De^{-sT}
\end{array} \right) = 0
\]

(1.6)

which is comparable with (1.2) except \( sI \) has been replaced by \( \frac{\rho(sh)}{\sigma(sh)} I \).

Now the question is, under what conditions does the stability of (1.2) imply the stability of (1.6)? We know of course if the \( T_1 = 0 \), then A-stability of \( (\rho, \sigma) \) is sufficient.

In [Brayton and Willoughby (1967)], the system

\[ x(t) + AX(t-T) + Bx(t) + CX(t-T) = 0 \]

(1.7)

(\( A, B, C \) symmetric, \( T \) scalar) was studied in conjunction with the multistep method

\[ \mu \dot{x}(t) + (1-\mu) \dot{x}(t-h) = \frac{x(t) - x(t-h)}{h} \]

which is A-stable for \( \frac{1}{2} \leq \mu \leq 1 \). It was shown under the conditions \( I + A, B + C > 0 \) (which imply stability of (1.7)), that the system

\[ \mu w(t) + A w(t-T) + B x(t) + C x(t-T) = 0 \]

\[ \mu w(t) + (1-\mu) w(t-h) = \frac{1}{h} (x(t) - x(t-h)) \]

(1.8)

is stable for all \( h \geq 0 \) when \( \frac{1}{2} \leq \mu \leq 1 \). The characteristic equation for (1.8) is

\[
\det \left( \begin{array}{cc}
I + Ae^{-ST} & B + C e^{-ST} \\
h\sigma(sh)I & -\rho(sh)I
\end{array} \right) = 0 .
\]

(1.9)

In [Brayton and Willoughby (1967)] the essence of the proof was to show for \( \text{Re } s \geq 0 \) and \( I + A, B + C > 0 \), that

\[ (I + Ae^{-ST})x + (B + C e^{-ST})y = 0 \]
implies that \( \text{Re} (x^* y) < 0 \), \( x, y \neq 0 \). Since \( y^* x = \frac{\rho(z)}{\text{ho}(z)} x^* y \), then 
\( \text{Re} x^* y = \text{Re} \frac{\rho}{\text{ho}} \geq 0 \) (for A-stable methods). This shows that (1.9) cannot have a root with \( \text{Re} s \geq 0 \).

This simple idea is exploited more fully in this paper.

**THE FIELD OF VALUES AND PASSIVITY**

It is useful to generalize slightly the notion of the field of values. Normally \( F(A) \) denotes the set \( \{ x^* A x \} \) for square matrices \( A \). Let \( A, B \) be matrices of the same dimension. The field of values of the pair \( (A, B) \) is defined as

\[
F(A, B) = \left\{ \frac{x^* y}{x^* x + y^* y} \right\} \quad \text{where} \ A x + B y = 0.
\]

Since \( F(A, B) = \left\{ \frac{x^* J x}{x^* x} \right\} \) where \( J = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \), then

\( F(A, B) \) is the field of values of \( J \) when \( x \) is confined to a linear subspace orthogonal to \( (A, B) \). Hence it inherits many of the properties of the normal field of values, e.g. \( F(A, B) \) is convex.

The pair \( (A, B) \) is passive if

\[ \text{Re} (F(A, B)) \leq 0. \]

If \( \text{Re} (F(A, B)) < 0 \), then \( (A, B) \) is strictly passive. In these terms, A-stability is equivalent to \((\rho(z), \sigma(z))\) being passive when \( \text{Re}(s) \geq 0 \).

If \( A \) and \( B \) are square matrices and \( B = -I \), then \( y = A x \) and \( x y = x A x \) so that

\[
F(A, -I) = \left\{ \frac{x^* A x}{x^* x + A^* A x} \right\} \quad \text{Hence}
\]

\[
\left( \frac{1}{1 + \| A \|^2} \right) |F(A)| \leq |F(A, -I)| \leq |F(A)|.
\]

If \( (A, -I) \) is passive (strictly passive), we say that \( A \) is passive (strictly passive). The term passive comes from the electrical engineering literature [Belevitch (1968)]. For example if \( x \) is the vector of currents and \( y \) vector of the voltages of an \( n \)-port, then passivity means that the \( n \)-port can only absorb power. Another term sometimes used is dissipative.
The use of passivity in stability theory is a special case of the following observation. If \( F(A, B) \cap F(C, D) = \emptyset \) then \( \begin{pmatrix} A & B \\ C & D \end{pmatrix} \) has full rank. The proof of the next theorem is essentially based on this.

The next two lemmas help us establish the existence of characteristic roots in a region \( \text{Re}(s) \leq -\delta < 0 \), i.e. bounded away from the imaginary axis.

Let \( S(\eta, A, B) = \bigcup_{\text{Re}(s) \geq \eta} F(A, B) \) where generally \( A, B \) may depend on the complex parameter \( s \). For two sets \( S_1, S_2 \),

\[ \rho(S_1, S_2) = \sup_{s_1 \in S_1} \inf_{s_2 \in S_2} |x - y| \] 

\( S(\eta, A, B) \) is continuous in \( \eta \) if given \( \varepsilon > 0 \), \( \exists \delta > 0 \) such that \( \rho(S(\eta_1, A, B), S(\eta_2, A, B)) \leq \varepsilon \) whenever \( |\eta_1 - \eta_2| < \delta \). \( S(\eta, A) \) will denote \( S(\eta, A, -I) \).

**Lemma 1** If \( (I - D e^{-sT}) \) is nonsingular for \( \text{Re}(s) \geq -\delta_1 \), some \( \delta_1 > 0 \), then \( S(\eta, A + B e^{-sT} (I - D e^{-sT})^{-1} C^*) \) is continuous for \( \eta \geq -\delta_1 \).

**Proof:** Since \( (I - D e^{-sT}) \) is nonsingular for \( \text{Re}(s) \geq -\delta_1 \), then

\[
\frac{x^*(A + B e^{-sT} (I - D e^{-sT})^{-1} C^*) x}{x^* x + y^* y}
\]

is continuous on the bounded set \( x^* x = 1, |x| \leq e^{\delta_1} \). Thus \( f(\zeta, x) \)

is uniformly continuous. It follows that \( S(\eta, A + B e^{-sT} (I - D e^{-sT})^{-1} C^*) \)

is continuous in \( \eta \) for \( \eta \geq -\delta_1 \) since

\[
\rho(S(\eta_1, A + B e^{-sT} (I - D e^{-sT})^{-1} C^*), S(\eta_2, A + B e^{-sT} (I - D e^{-sT})^{-1} C^*))
\]

\[
= \sup_{x_1, \zeta_1} \inf_{x_2, \zeta_2} |f(x_1, \zeta_1) - f(x_2, \zeta_2)|
\]

\[
\leq \frac{\eta_1 - \eta_2}{\varepsilon}
\]

**Lemma 2** Let \( \sigma(sh), \rho(sh) \) be polynomials in \( e^{-sh} \). If \( \sigma(sh) \) has no roots in \( \text{Re}(sh) \geq -\delta_2 \), then \( \text{Re}(S(\eta, -h\sigma(sh), \rho(sh))) \) is continuous in \( \eta \) for \( \text{Re}(s) \geq -\delta_2 \).
Proof: We note that
\[ \text{Re} \left( \frac{x^* y}{x^* x + y^* y} \right) = \frac{\text{Re}(z) h}{|z|^2 + h^2} , \quad hx = \frac{\rho(sh)}{\sigma(sh)} y \equiv z(sh)y. \]

Since \( \rho \) and \( \sigma \) are polynomials in \( e^{-sh} \), then \( z(sh) \) is uniformly continuous in \( sh \) for \( \text{Re}(sh) \geq -\delta_2 \). Following the same argument used in Lemma 1, \( S(\eta, -\sigma(sh), \rho(sh)) \) is continuous in \( \eta \) for \( h\eta \geq -\delta_2 \).

**Theorem 1** Assume that \( \det (I - De^{-sT}) = 0 \) has no roots in \( \text{Re}(s) > -\delta_1 \), for some \( \delta_1 > 0 \). If \( \text{Re}(F(A+Be^{-sT}(I - De^{-sT})^{-1}C*) ) \leq -\varepsilon < 0 \) for \( \text{Re}(s) > 0 \), then (1.1) is stable. If in addition \( (\rho(sh), \sigma(sh)) \) is passive and \( \sigma(sh) = 0 \) has no roots for \( \text{Re}(sh) \geq 0 \) (i.e. A-stable), then (1.3) is stable for all \( h > 0 \).

Proof: The characteristic equations (1.2) and (1.4) can be put in the same form
\[ \det \begin{pmatrix} -I & A+Be^{-sT}(I - De^{-sT})^{-1}C* \\ hI & -zI \end{pmatrix} = 0 \quad (2.1) \]
where \( z = sh \) for (1.2) and \( z = \frac{\rho(sh)}{\sigma(sh)} \) for (1.4). In both cases, \( z \) has the property \( \text{Re}(z) \geq 0 \) for \( \text{Re}(s) \geq 0 \) and \( S(\eta, h, -z) \) is continuous for \( \eta \geq -\delta_2 \) for some \( \delta_2 > 0 \) using Lemma 2. Also by Lemma 1, \( S(\eta, A+Be^{-sT}(I - De^{-sT})^{-1}C*) \) is continuous for \( \eta \geq -\delta_1 \). Since \( \text{Re}(S(0, A+Be^{-sT}(I - De^{-sT})^{-1}C*) ) \leq -\varepsilon < 0 \) and \( \text{Re}(S(0, h, -z)) \geq 0 \), then using the continuity in \( \eta \) of both of these sets, it is clear that they remain disjoint for some \( -\eta_1 < 0 \). Hence there can be no roots of (2.1) in \( \text{Re}(s) \geq -\eta_1 \).

The next theorem shows that if one can confine the field of values to a sector, then almost A-stable methods can also give stability.

**Theorem 2** Assume \( \det (I - De^{-sT}) = 0 \) has no roots in \( \text{Re}(s) > -\delta_1 \), for some \( \delta_1 > 0 \). If for \( \theta > 0 \)
\[ a) \quad \text{Re} \left( \bigcup_{\theta \geq 0} F(e^{i\theta}(A+Be^{-sT}(I - De^{-sT})^{-1}C*)) \right) \leq -\varepsilon < 0, \quad \text{and} \quad \text{Re}(s) \geq 0 \]
\[ b) \quad |\arg(\frac{\rho}{\sigma})| < \frac{\pi}{2} + \theta \quad \text{for} \quad \text{Re}(s) \geq 0, \]
then (1.3) is stable for all \( h > 0 \).
Proof: The proof is similar to theorem 1, using lemma's 1 and 2 to bound the roots away from the imaginary axis.

It is interesting, because of the expected neutrality of the equations, that although one cannot confine the eigenvalues of the system (1.1) (i.e., the roots of \( \det (sI - (A + Be^{-sT}(I - De^{-sT})^{-1}C)) = 0 \)) to a sector in the left half plane of angle less than \( \pi \), it is possible (under the assumption, \( \det(I - De^{-sT}) = 0 \) has no roots in \( \Re(s) > -\delta \)) to confine \( S(0, A + Be^{-sT}(I - De^{-sT})^{-1}C) \) to such a sector. The distinction here between differential equations and difference-differential equations can be emphasized by letting \( T = 0 \) in the first case. The statement:

\[
\bigcup \ F(A + Be^{-sT}(I - De^{-sT})^{-1}C) \text{ contains the roots of } \Re(s) > 0
\]
\[
\det(sI - A - Be^{-sT}(I - De^{-sT})^{-1}C) = 0
\]

is valid for \( T = 0 \) but is generally invalid for \( T \neq 0 \).

STABILITY FOR ELECTRICAL NETWORKS WITH LOSSLESS TRANSMISSION LINES

Generally it would be difficult, without additional information, to test or prove that \( \Re(\bigcup F(A + Be^{-sT}(I - De^{-sT})^{-1}C) \leq -\varepsilon < 0 \) as \( \Re(s) > 0 \) required in theorem 1. In this section we want to show how to obtain equivalent results by examining the component parts of a system. Thus we will express the system in an unreduced form as algebraic, differential, and delay equations.

The systems to be studied are those of linear electrical networks with the delay terms due to lossless transmission lines,

\[
Ax + By = 0
\]
\[
L \frac{dx_1}{dt} + y_1 = 0
\]
\[
x_2 + Cy_2 = 0
\]
\[
(y_3 - Rx_3)(t) = E [E^{-1}J(y_3 + Rx_3)](t - \tau)
\]

where \( x = (x_1, x_2, x_3)^T, y = (y_1, y_2, y_3)^T, x_1, y_1 \) are vectors of dimension \( n_1 \), and \( n_3 \) is even. \( R, E \) are related by \( R = E D E^{-1} \) where \( D = \text{diag}(d_1, \ldots, d_{n_3}, d_1, \ldots, d_{n_3}) \). Further \( J = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \) where \( I \)}
is the identity matrix of size \( n_3/2 \) and \( EJ = JE \). Again

\[
\tau = (\tau_1, \ldots, \tau_{n_3})^T = (\tau_1, \ldots, \tau_{n_3/2}, \tau_1, \ldots, \tau_{n_3/2})^T.
\]

The general form of the last equation of (3.1) is derived by solving the partial differential equations of lossless coupled transmission lines and expressing the solution in terms of delay equations [Ho (1973)].

In (3.1), the physical interpretation is that \( x \) is a vector of currents, \( y \) a corresponding vector of voltages, \( L \) an inductance matrix, \( C \) a capacitance matrix. The first equation of (3.1) is the relation between currents and voltages imposed by an \( n \)-port given by \((A,B)\). The starting assumption will be that \((A,B)\) is passive, a fact that may be readily obtained in some electrical network problems.

The characteristic equation for (3.1) is

\[
\begin{vmatrix}
A_1 & A_2 & A_3 & B_1 & B_2 & B_3 \\
Lz(s) & 0 & 0 & -hI & 0 & 0 \\
0 & -hI & 0 & 0 & Cz(s) & 0 \\
0 & 0 & (I+EW(s)E^{-1}J) & 0 & 0 & -(I-EW(s)E^{-1}J)
\end{vmatrix} = 0.
\]

(3.2)

Here again we have used the convenience of the symbols \( z, W \) in order to put a number of stability results into one form. In the case of \( (3.1) \) \( z = sh \) and \( W = \text{diag} (e^{-s_1}, \ldots, e^{-s_{n_3}}) \).

In the next theorem, we only need the following properties of \( z(s) \) and \( W(s) \):

a) \( \Re(S(0,z(s))) \geq 0 \) and \( S(\eta,z(s)) \) is continuous in \( \eta \) at \( \eta = 0 \).

b) \( W(s) = \text{diag} (w_1(s), \ldots, w_{n_3}(s), w_1(s), \ldots, w_{n_3}(s)) \) where each \( w_i \) satisfies \( |w_i(s)| \leq 1 \) for \( \Re(s) \geq 0 \) and \( |w_i(s)| \) continuous in \( \eta = \Re(s) \) at \( \eta = 0 \).

**Theorem 3** Suppose \( \Re F(A,B) < 0 \) and \( L, C \) are positive definite symmetric matrices. If a) and b) hold, then the roots of (3.2) lie in a left-half plane \( \Re(s) \leq -\delta < 0 \).

**Proof:** We note that for arbitrary square matrices \( M_1, M_2, M_3, M_4 \)

\[
F \left( \begin{pmatrix} M_1 & 0 \\ 0 & M_3 \end{pmatrix}, \begin{pmatrix} M_2 & 0 \\ 0 & M_4 \end{pmatrix} \right) = CO \left( F(M_1, M_2), F(M_3, M_4) \right)
\]
(CO stands for convex hull). Therefore it is enough to show that each of the pair of matrices in the last three rows of (3.2) satisfy for $n \geq 0$.

1) $\text{Re } S(\eta, Lz(sh), -hJ) \geq 0$

2) $\text{Re } S(\eta, -hJ, Cz(sh)) \geq 0$

3) $\text{Re } S(\eta, (I+EW(s)E^{-1}J)R, -(I-EW(s)E^{-1}J)) \geq 0$.

Then if each of these three sets is continuous in $\eta$, the appropriate fields of values will remain disjoint for $\text{Re}(s) \geq -\delta$ for some $\delta > 0$. Hence (3.2) will be nonsingular for $\text{Re}(s) \geq -\delta$ and there can be no roots there.

Clearly 1) and 2) hold since $L$ and $C$ are positive definite and $z(s)$ satisfies a) above.

To prove 3) we note that with the transformation $\tilde{x} = E^T x$, $\tilde{y} = E^{-1} y$, the equations become

\[
(I+EW^{-1}J)R E^{-T} \tilde{x} - (I-EW^{-1}J)E \tilde{y} = 0
\]

or

\[
(I + WJ)D \tilde{x} - (I - WJ)\tilde{y} = 0.
\]

The field of values, except for rescaling in magnitude, is unchanged since $x^T y = \tilde{x}^T \tilde{y}$. The equation (3.3), if rearranged, can be put in block diagonal form where the $i$th block of equations is $2 \times 2$ and has the form

\[
d \begin{pmatrix}
  1 & w \\
  w & 1
\end{pmatrix}
\begin{pmatrix}
  x_1 \\
  x_2
\end{pmatrix}
- \begin{pmatrix}
  1 & -w \\
  -w & 1
\end{pmatrix}
\begin{pmatrix}
  y_1 \\
  y_2
\end{pmatrix}
= 0.
\]

Let $U = \begin{pmatrix}
  1 & \frac{1}{\sqrt{2}} \\
  -\frac{1}{\sqrt{2}} & 1
\end{pmatrix}$ and $x = U \tilde{x}$, $y = U \tilde{y}$. Since $U^* \begin{pmatrix}
  1 & +w \\
  +w & 1
\end{pmatrix} U$

\[
= \begin{pmatrix}
  1 & +w \\
  0 & 1 - w
\end{pmatrix},
\]

the equations become

\[
d(1-w)\tilde{x}_1 - (1+w)\tilde{y}_1 = 0
\]

\[
d(1+w)\tilde{x}_2 - (1-w)\tilde{y}_2 = 0.
\]

Again $(x)^* y = (x) y$.

Using the fact that $d > 0$, a little computation gives

\[
\frac{\tilde{x}_j \tilde{y}_j}{|x_j|^2 + |y_j|^2} = \frac{d \ (1 + w) \ (1 + \bar{w})}{|1 + w|^2 + d^2 |1 + \bar{w}|^2} \quad \text{for } |w| \leq 1.
\]
Hence using b) we conclude that
\[ \text{Re } S(\eta, (I + \mathbf{E}^{-1}J)R, -(I - \mathbf{E}^{-1}J)) \geq 0 \text{ for } \eta \geq 0 \]
and this set is also continuous at \( \eta = 0 \).

**Theorem 4** Let \( \sum_{i=0}^{k} \alpha_i x(t-ih) = h \sum_{i=0}^{k} \beta_i x(t-ih) \) be an A-stable LMS method. Suppose \((A,B)\) is strictly passive, and \(L, C > 0\). Then the following system of difference equations (the difference analog of (3.1)) is stable.

\[
\begin{align*}
A x + B y &= 0 \\
L w_1 + Y_1 &= 0 \\
x_2 + C w_2 &= 0 \\
(y_3 - R x_3)(t) &= E[-1J(y_3 + Rx_3)](t-\tau) \\
\sum_{i=0}^{k} \alpha_i x_i(t-ih) &= h \sum_{i=0}^{k} \beta_i w_i(t-ih) \\
\sum_{i=0}^{k} \alpha_i y_i(t-ih) &= h \sum_{i=0}^{k} \beta_i w_2(t-ih) \\
\end{align*}
\]

**Proof:** We let \( z(sh) \equiv \frac{\rho(sh)}{\sigma(sh)} = \frac{\sum \alpha_i e^{-ish}}{\sum \beta_i e^{-ish}} \). Lemma 2 and A-stability establish a). With \( W(s) = e^{-sT} \), then \( W(s) \) clearly satisfies b). If \( \beta(sh) \neq 0 \), the roots of the characteristic equation of (3.4) are given by (3.2). Since \( \beta(sh) \) has no roots in \( \text{Re}(sh) \geq -\delta_2 \) for some \( \delta_2 > 0 \) and theorem 3 establishes that (3.2) has no roots in \( \text{Re}(sh) \geq -\delta_1 \) some \( \delta_1 > 0 \), then (3.4) is stable.

The next theorem deals with what might be termed passive interpolation. Here we use a result of Strang (1962). If we want to solve the above equations on a computer, we will be required to interpolate to get the values of the delayed terms in (3.4). To evaluate a delayed term, say \( x(t-\tau) \), let \( m_\tau \) and \( \eta \) be defined by \( \tau = m_\tau h-\eta \) where \( 0 \leq \eta < 1 \) and suppose \( t = nh \). Then the Lagrange interpolating polynomial using \( 2N+1 \), or \( 2N \) points is

\[
x(t-\tau) = \sum_{k=-N}^{N} \gamma_k(\eta) x((n - m_\tau - k)h) \tag{3.5}
\]

where

\[
\gamma_k(\eta) = \sum_{j=-N}^{N} \binom{\eta-j}{k-j}
\]

and \( \hat{N} = N \) (odd case) or \( \hat{N} = N-1 \) (even case). Strang's theorem (1962)
NUMERICAL A-STABILITY FOR DIFFERENCE-DIFFERENTIAL SYSTEMS

is that for $\omega$ real

$$
\left| \sum_{k=-N}^{N} \gamma_k(\eta) e^{ik\omega} \right| \leq 1 \quad \text{for} \quad \begin{cases} -1 \leq \eta \leq 1 & \hat{N} = N \\ 0 \leq \eta \leq 1 & \hat{N} = N-1 \end{cases}
$$

If one uses (3.5) in (3.4), to evaluate the term

$$
\{ E^{-1} J (y_3 + R x_3) \}(t-\tau),
$$

then the characteristic equation will be as in (3.2) where $W = \text{diag} (w_1', \ldots, w_n')$,

$$
w_i(s) = e^{\tau_i} \sum_{k=-N}^{N} \gamma_k(\eta_i) e^{-ksh}, \quad \text{and} \quad \tau_i = \frac{m_i}{h} - \eta_i \quad 0 \leq \eta_i < 1.
$$

Lemma 3

$$
|w_i(s)| \leq 1 \quad \text{for} \quad \Re s \geq 0.
$$

Proof:

$$
|w_i(s)| \leq e^{\tau_i} \left| \sum_{k=-N}^{N} \gamma_k(\eta_i) e^{-(N+k)sh} \right|.
$$

Define

$$
f(z) = \sum_{k=-N}^{N} \gamma_k(\eta_i) z^{k+N}.\text{This maps the unit disk into itself since for } |\zeta| > 1
$$

$$
\Delta \arg (f(z) - \zeta) \bigg|_{z=e^{i\theta}} = 0
$$

using Strang's result. Since $f(z)$ has no poles in $|z| < 1$, then $f(z) - \zeta$ has no zeros there i.e. $f(z) \neq \zeta$, $|\zeta| > 1$, $|z| \leq 1$.

Theorem 5

Let $\alpha_i, \beta_i$ be an A-stable LMS method. Suppose $(A, B)$ is strictly passive and $L, C > 0$. Then the following system of difference equations (the discrete analog of (3.1) with passive interpolation) is stable,

$$
\begin{align*}
Ax(t) + By(t) &= 0 \\
Lw_1(t) + y_1(t) &= 0 \\
x_2(t) + Cw_2(t) &= 0 \\
y_3(t) - Rx_3(t) &= E \tilde{w}_3(t) \\
\sum \alpha_i x_1(t-ih) &= h \sum \beta_i w_1(t-ih) \\
\sum \alpha_i y_2(t-ih) &= h \sum \beta_i w_2(t-ih) \\
w_3(t) &= E^{-1} J (y_3(t) + R x_3(t)) \\
(\tilde{w}_3)_i(t) &= \sum_{k=-N}^{N} \gamma_k(\eta_i) (w_3)_i(t - m_i h - kh) \quad i = 1, \ldots, n_3
\end{align*}
$$

(3.6)
Proof: \( w_i(s) \) is clearly continuous in \( \text{Re}(s) \) for \( \text{Re}(s) \geq 0 \). Application of Lemma 3 and Theorem 4 gives the required result.

In fact it is always possible to use the odd case, \( \hat{N} = N \), since if \( x(jh) \) is known for \( j \leq n-1 \) and \( x(nh) \) is the new unknown, then interpolation can be done for \( x(nh-\tau_i) \) using 3 points (quadratic interpolation) if \( \tau_i \geq h \). If \( \tau_i \geq 2h \), then five points (quartic interpolation) can be used etc.

However if \( \tau_i < h \), then implicit interpolation using 3 points would have to be used i.e. the equation
\[
(\hat{\omega}_3)_i(nh) = \sum_{k=-1}^{1} \gamma_k(n_i)(\omega_3)_i((n-1)h-kh)
\]
would be regarded as an equation involving the unknown \( (\omega_3)_i(nh) \) which would have to be solved simultaneously* with the other equations of (3.6) for unknowns at \( t = nh \).

It would seem that to preserve the freedom of choosing \( h \) as large as we please, we would want to allow for the possibility of implicit interpolation and let \( h > \tau \) if required.

* C. W. Ho suggested that this should be implemented by predicting the forward point and then iterating to satisfy the interpolation equations, rather than a simultaneous direct solution with the other equations.
Let the problem of optimal control be defined as

\[ \dot{\eta} = \phi(t, \eta, u), \quad t_o \leq t \leq t_e \]

with boundary conditions for the state variable \( \eta(t) \) at \( t = t_o \) and \( t = t_e \), \( t_e \) fixed or free, \( u(t) \) is the control; the quantity \( \eta \) to be optimized is denoted by \( \Phi(t_e, \eta(t_e)) \).

It is well-known that the solution of this problem may be obtained as a solution of the two-point boundary value problem

\[ y' = f(x, y) \]
\[ r(y(0), y(1)) = 0 \]

where

\[
y = \begin{pmatrix} \eta \\ \lambda \\ t_e - t_o \end{pmatrix}, \quad f = \begin{pmatrix} \phi \\ -\frac{\partial H}{\partial \eta} \\ 0 \end{pmatrix}, \quad H = \lambda^T \phi
\]

\[ t = t_o + (t_e - t_o)x, \quad 0 \leq x \leq 1. \]

The control \( u(t) \) is expressible in terms of the Lagrangian multipliers \( \lambda \) and thus may be eliminated. If, in addition, constraints are imposed upon the control vector or the state variables, switching

\[ \text{Full manuscript to be published later in a periodical.} \]
The differential equations are mildly stiff.

Suppose now a numerical solution of (2) has been found and, to fix ideas, let $y(x)$ be the optimal flight path for an aerospace vehicle.

Let the vehicle fly along this path. During the flight a deviation from the computed optimal path may occur (due to erroneous measurements of initial data etc.) say at $x = x^*, 0 < x^* < 1$.

The following problem now arrives: find the new control $u$ and the new optimal path in $x^* < x < 1$ according to (1) but with boundary conditions at $t = 0$ replaced by the new data of the state variables at $x = x^*$.

This leads again to a two-point-boundary value problem as already described above. The problem, however, is now to use algorithms which compute the new optimal path from the old one as fast as possible in order to be competitive with the flight-time of the vehicle. This problem has been partially successfully attacked by a multiple shooting procedure: The interval $0 \leq x \leq 1$ is divided into subintervals $0 = x_1 < \ldots < x_j < \ldots < x_m = 1$

and the Jacobian matrices

$$G_j := \frac{\partial y(x_{j+1})}{\partial y(x_j)}$$

are computed and stored before the flight actually takes place.

During the flight, i.e., for $x > x^*$, a Newton-Raphson procedure is then applied to find the "new" optimal path from the "old" one. At the end of each iteration step the matrices $G_j$ are modified and adapted to the new iterates by a rank-one-method due to Broyden et al.

Numerical experiments have shown, that in case of an optimal deceleration maneuver of an Apollo-type vehicle in the Earth-atmosphere the computing time for deviations up to 1% is about 1 sec (on an IBM 370 - 165).

In less critical cases, for example, the ascent of a vehicle from the moon surface, deviations up to 10% are allowed with a necessary computing time of about 0.5 sec. The method may still be improved by using initial-value-solver which are more adapted to the above problem and, as we hope, by using rank-two-methods for the modification of the iteration matrices $G_j$.

Incidentally, the method may be of some use for the real-time-control of chemical reaction problems.
SIMULATION OF CHEMICAL KINETICS TRANSPORT IN THE STRATOSPHERE*

J. S. Chang, A. C. Hindmarsh and N. K. Madsen
Lawrence Livermore Laboratory, University of California
Livermore, California 94550

INTRODUCTION

It is well known that the chemical rate equations are often very stiff and their numerical solutions must be treated quite carefully. At the present this does not present any major difficulty due to the class of special methods for stiff ordinary differential equations. However, in the simulation of chemical kinetics transport in the stratosphere and related fields we are faced with a large system of stiff partial differential equations. Abstractly we may write the equations as

$$\frac{\partial y}{\partial t} = T(y) + C(y)$$

where $y$ is a vector representing the concentrations of the chemical species and $t$ is simply time. $T$ is a linear partial differential operator describing the mass transfer of the chemical species [Colgrove (1965A), Gudiksen (1968A)]. $C$ is a highly nonlinear chemical kinetics operator involving not only the concentrations of these species but also some complicated integrals of these concentrations [Crutzen (1971A), Johnston (1971A)]. The physical problem is usually formulated either as an initial-boundary value problem or simply as a boundary value problem (i.e. $\frac{\partial y}{\partial t} = 0$).

In the numerical solutions of Eq. (1) we must deal with a large set

*Work performed in part under the auspices of the U. S. Atomic Energy Commission and in part supported by the Climatic Impact Assessment Program, Office of the Secretary, Dept. of Transportation
†See references at the end of the paper.
of strongly coupled partial differential equations (usually greater than ten) describing some physical processes in a nonhomogeneous medium (e.g. the atmosphere). Furthermore these processes are characterized by widely separated time constants (e.g. $10^{-6}$ sec to $10^{+8}$ sec). Following the basic concept of the method of lines [Liskovets (1965A), Walter (1970A)] by first reducing this system to a much larger system of ordinary differential equations and then utilizing a solution technique based on the method of Gear, we have successfully constructed a program for carrying out such simulations. In this report we would like to discuss the motivation, effectiveness and limitations of this solution technique. We will also present some simulation results on some physical problems of current interest.

**CHEMICAL KINETICS TRANSPORT IN THE STRATOSPHERE**

The conservation equation for the concentration $c^i$ of any chemical species may be written as

$$ \frac{dc^i}{dt} = \nabla \cdot \left( \nabla c^i \right) + P(c) - L(c) + S_i $$

where

- $t =$ time;
- $\nabla c^i =$ gradient of $c^i$
- $P(c) =$ photochemical production of species $c^i$ from other species;
- $L(c) =$ photochemical loss of species $c^i$ in reactions with other species;
- $S_i =$ any other possible sources.

The dependent variables are of course functions of the spatial coordinates and $t$.

A typical system may be

$$ \frac{3y_1}{3t} = \nabla \cdot \left( \nabla y_1, y_1 \right) + S_1 + J_2 y_2 + J_3 y_4 - J_4 y_1 - k_1 y_1 y_2 - k_2 y_1 y_4 $$

$$ \frac{3y_2}{3t} = \nabla \cdot \left( \nabla y_2, y_2 \right) + J_4 y_1 - J_2 y_2 - k_4 y_2 y_3 $$

$$ \frac{3y_3}{3t} = \nabla \cdot \left( \nabla y_3, y_3 \right) + J_4 y_4 + k_2 y_2 y_4 - k_3 y_2 y_3 $$

$$ \frac{3y_4}{3t} = \nabla \cdot \left( \nabla y_4, y_4 \right) + k_3 y_2 y_3 - k_4 y_4 - k_2 y_1 y_4 $$
The transport flux \( \dot{\psi}(V_c^i, c^i, t) \) may be obtained either from the direct solution of the equations of motion of the earth's atmosphere or through parameterization of observed motions. Since we are interested in the distributions of the so called minor chemical species of the stratosphere we shall adopt the latter approach. A chemical species is considered to be a minor species if the ratio of its concentration to the ambient air density is less than \( 10^{-5} \). Due to their relatively low concentrations we would like to assume that they do not affect the atmospheric motions significantly. Strictly speaking this is not quite correct, e.g. the atmospheric ozone is a major heat source in the stratosphere. From studies on atmospheric nuclear fallout \( \text{[Gudiksen (1968A)]} \) we know that the above assumption is valid at least as a first order approximation.

The transport flux \( \dot{\psi} \) consists of a term representing the mean motion and a term representing the large scale turbulent mixing commonly called eddy diffusion \( \text{[Reed (1965A)]} \),

\[
\dot{\psi} = K \cdot V_c^i + V_c^i \cdot \nabla.
\]

\( K \) is a matrix whose elements are the eddy diffusion coefficients and \( V \) is a vector describing the observed mean atmospheric motion. There are two types of terms for the photo chemical processes: the production or loss due to photolysis,

\[
A_i + \text{radiation} \rightarrow A_j + A_k,
\]

and that due to chemical interactions

\[
A + B + C + D.
\]

The former process is described by the coefficients \( J_{\ell} \) in Eq. (3), where \( \text{[Gelinas (1973A)]} \)

\[
J_{\ell} = \int d\lambda \sigma^i_{\ell}(T)Q_{\ell}(i+j)I_o \exp[-\tau(x,t)]
\]

\( J_{\ell} \) is commonly called the photodissocation coefficient; \( I_o \) is the radiation intensity for each wavelength \( \lambda \) at the top of the atmosphere; \( \tau \) is an attenuation factor at point \( x \) and time \( t \), \( \sigma^i_{\ell} \) is the total absorption cross section of molecules \( A_i \); \( Q_{\ell}(i+j) \) is the quantum yield for the process in consideration; \( T \) is the temperature and \( x \) represents the spatial coordinates. The attenuation factor (optical depth) is in turn an integral over all absorptions and scattering of a given wavelength due to all chemical species present. Clearly many simplifying assumptions must be utilized in order to evaluate all the complicated integrals, however this is of no essential interest in the present context and interested readers may refer to \( \text{[Gelinas (1973A)]} \). By comparison the chemical
interactions are simply represented by temperature dependent coefficients $k_i$ for each interaction.

Physically it is known that the photochemical reaction rates span a range of time constants from about $10^{-6}$ sec to $10^{+5}$ sec. On the other hand it is also known that the atmospheric transport processes have characteristic time constants of about $10^8$ or $10^9$ seconds. When these processes are coupled together as in Eq. (2), the numerical solution of Eq. (2) becomes difficult. Due to the vast difference in time scales, operator splitting techniques or the method of fractional steps [Yanenko (1971A)] may seem to be quite appropriate. However, it quickly becomes evident that every time the transport operator is applied the chemical system is rudely perturbed out of equilibrium and the computing effort required to smooth the total system is again costly. Also there is the ever present question of how accurate is the time dependent solution under such artificial perturbations. On the other hand, in recognizing the diffusion operator as a smoothing operator while the chemical kinetics operators are the dominant driving force of the system, we then discretize the transport operators and solve the resulting system of ordinary differential equations with a stiff ODE solver. This approach carries with it the technical difficulties of storage and the repeated inversion of very large matrices. From the viewpoint of the numerical solution of parabolic partial differential equations this is really equivalent to using a variable order difference formula for the time operator and this by itself is an interesting numerical experiment.

**NUMERICAL METHOD**

We shall illustrate the details of the numerical technique by an example incorporating all the essential features of Eq. (2). Let us consider the one-dimensional equation

$$\frac{\partial c_i}{\partial t} = \frac{3}{\partial z} K(z) \frac{\partial c_i}{\partial z} + P^i(c) - L^i(c)$$

for $i = 1, \ldots, I$, with the appropriate initial and boundary conditions at $z = 0$ and $z = 1$. With $J$ equally spaced mesh points, the semi-discrete approximating system of ordinary differential equations is

$$\frac{\partial c_i}{\partial t} = \frac{1}{(\Delta z)^2} \left[ K^i_{j+\frac{1}{2}} c_{j+1}^i - \left( K^i_{j+\frac{1}{2}} + K^i_{j-\frac{1}{2}} \right) c_j^i + K^i_{j-\frac{1}{2}} c_{j-1}^i \right]$$

$$+ P^i(c_j) - L^i(c_j) ,$$

and appropriate equations at the boundary are assumed.
Since there are $I$ chemical species and $J$ spatial zones the size of the problem is $N = IJ$. The $I$ species in any given zone interact with each other according to the system of kinetics rate equations adopted. Each given species also interacts with the same species in the neighboring zones, and the transport coefficients governing that interaction are the same for all species. Suppose that we order the variables by grouping all species in zone 1, then all species in zone 2, etc. Then the system of equations may be written

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

with $y = (c_1^1, \ldots, c_I^1, c_1^2, \ldots, c_2^1, \ldots, c_J^1, \ldots, c_J^I)$.

It follows from Eq. (5) that $A$ will have the following block structure:

$$A = \begin{bmatrix}
A_1 & t_{12} & t_{13} \\
t_{21} & A_2 & \\
t_{31} & t_{32} & A_3
\end{bmatrix}$$

with $I$ blocks on the left and $J$ blocks on the right.
Here the basic blocks are $I \times I$ matrices, and there are $J^2$ such blocks comprising $A$. The diagonal blocks $A_1, \ldots, A_J$ represent the chemistry within the zones plus the diagonal element of the transport operator. The off-diagonal blocks represent the remaining portion of the transport operator, and each such block is a multiple of the $I \times I$ identity matrix. Hence $A$ is fully described by the $J$ matrices $A_j$ (each $I \times I$) and the $J \times J$ matrix $T = (t_{ij})$ of transport coefficients.

Given any initial conditions Eq. (6) is solved by a modified version of Gear's method [Gear (1968A)]. We will not discuss the details of the method here as it is well documented elsewhere [Hindmarsh (1972A), (1972B), (1972C)]. The basic nonlinear multi-step formula

$$y_n = \sum_{i=1}^{q} \alpha_i y_{n-i} + \beta_0 h f(y_{n}, t_n)$$

is solved by a modified Newton method. This is necessitated by the complex photodissociation coefficients. The matrix to be inverted has the form $P = I - \beta_0 h \frac{\partial f}{\partial y}$ which in turn has the same structure as matrix $A$ except that the diagonal blocks $A_j$ are to be replaced by another set $D_j$. During the computation the $P$ matrix is kept as long as possible and is reevaluated only after the failure of corrector convergence. For one-dimensional problems the storage of $P$ is of no major concern but as we go to higher dimensional problems we will exceed the memory size of most computers in existence. For the present example storing the full $P$ matrix require $I^2 J^2$ locations while storing the central band requires $I^2 (2J-1) - I (J-1)$ locations and similar estimates can be given for multi-dimensional problems. We have adopted a block iterative method for the inversion of the $P$ matrix which reduces the storage requirements to essentially $I^2$ times the total number of spatial zones.

Heuristically, we know that the diagonal matrices $D_j$ contain not only all the nonlinear chemistry terms but also the diagonal elements of the transport operator. Therefore, when considered in its block form, the matrix $P$ should be diagonally dominant. Consequently we may be fairly confident that successive overrelaxation (SOR) method will converge. Let $x_j = (c_{j1}, c_{j2}, \ldots, c_{jI})$ be the $j$th block of $y$, then the system

$$Py = b$$

is iteratively solved using the SOR formula.
SIMULATION OF CHEMICAL KINETICS TRANSPORT IN THE STRATOSPHERE

\[
D_j x_j^{(u+1)} + w \sum_{i<j} t_{ji} x_i^{(u+1)} = (1-w) D_j x_i^{(u)}
- w \sum_{i>j} t_{ji} x_j^{(u)} + w b_j ,
\]

(7)

\(b_j\) being the \(j^{th}\) block of \(b\). The over-relaxation parameter \(w\) is determined by an algorithm similar to the one first suggested by Carré (Carré (1961)). There are two advantages in using SOR Eq. (7). The first is the relatively small storage required for \(P\). Now we need only to store \(J\) matrices of dimension \(I \times I\) or \(I^2 J\) total locations, or \(I^2 \) times the total number of spatial zones in general (plus a small matrix \(T = (t_{ij})\)). The second advantage is that by retaining the LU decompositions of each \(D_j\) then the overrelaxation iterations (Eq. (7)) become computationally inexpensive. Furthermore the same LU decomposition is used as long as the modified Newton iterations converge. It also should be noted that each time a new LU decomposition is required the total computational cost is much less than for the full \(P\) matrix. We shall call the SOR iterations Eq. (7) the inner iteration as compared to the basic modified Newton's iteration in the original ODE solver. The convergence criteria for the relative error of the inner iteration is set to be \(\varepsilon\). By varying the parameter \(\varepsilon\) over a range of 0.1 to 0.00001, no significant differences were found in the entire solutions of various problems. By this we mean that the relative error of the solutions has always remained less than the preset relative error constant of the ODE solver. This indicates that only a very modest accuracy of the SOR solution is required for the ODE solver to perform adequately. We have found that the higher order implicit formula for the time operator indeed limits the truncation error to be purely due to the spatial difference formula. The available numerical data suggests that this solution technique may prove to be applicable to broad classes of parabolic problems. At the present we are studying this in more detail but we already have some interesting evidence. For example, if we use the familiar factor \(\frac{a \Delta t}{(\Delta x)^2}\) as a heuristic guide, the present method at times can use time steps so large that \(\frac{a \Delta t}{(\Delta x)^2} \approx 10^5\) while preserving the prescribe accuracy.

While applying this method to a wide class of physical problems we noticed an instability which is somewhat puzzling. This is illustrated in Fig. 1. This is a graph of the time step size history. The crosses indicate when the program actually decided on an

Figure 1. An example of the time history of step size $h$ exhibiting certain instabilities in the error and step size control algorithms.
increase in step size and then failed to converge in the corrector stage. In fact there exist multiple failures even though at the later failures the time step size has already gone below the original time step size prior to the attempted increase. By modifying the step size changing algorithm to include some memory of previous failures and limiting the allowable increases accordingly, we are able to eliminate multiple failures but still observe over estimation. We should note that during all the multiple failures the solutions have hardly changed at all. Prior to and after this interval of instability the step size behaves very well and increases monotonically. This kind of instability may warrant further study in the near future.

The same solution technique has been applied to several two-dimensional problems with completely satisfactory results. For example the largest set of equations that we have solved involves 9 chemical species, 44 vertical zones, and 20 horizontal zones. This means that we have applied this solution technique to a system of 8316 ordinary differential equations. The solution was obtained over a time interval from 0.0 to $10^{10}$ sec. and it took 50 minutes on the CDC 7600.

SST AND THE POLLUTION OF THE STRATOSPHERE

The stratosphere is by definition a relatively calm region of the atmosphere. Consequently, the residence time of any material brought into the stratosphere is quite large (order of a year or more). With the advent of supersonic transports, which will be flying at the stratospheric altitudes, the environmental impact of the engine exhaust has become a major concern. A principal concern has been the possible catalytic destruction of ozone ($O_3$) by nitric oxide (NO). The hot engine exhaust contains a significant amount of NO which will be deposited in the stratosphere. Given a fleet of SST's the injected NO concentration could exceed many times the ambient [SCEP (1970A)]. Furthermore, the enhanced NO concentrations will greatly increase the destruction of $O_3$ through the so called $NO_x$ catalytic cycle [Crutzen (1971A), Johnston (1971A)]

$$NO + O_3 \rightarrow NO_2 + O_2$$

$$NO_2 + O \rightarrow NO + O_2$$

The net effect of these two reactions is to convert $O_3$ and $O$ to two $O_2$'s. It has been estimated that based on certain estimates of NO production rate from SST there could be a world wide reduction in ozone of from 3% to 50% [Johnston (1971A)]. Unfortunately ozone
Figure 2. Examples of model computed ambient concentrations and some comparable measurements.

Figure 3. The model predicted decreases in average ozone density due to fleets of US SST flying between 19.5 and 20.5 km and Concorde flying between 16.5 and 17.5 km.
not only is a major source of heat in the stratosphere but it is also the shield of ultraviolet (UV) radiation. Due to the large optical depth of UV radiation even a few percent change in ozone will cause up to many folds of changes in UV radiation depending on the frequency. Of course, UV radiation is directly related to many forms of skin cancer in man and the delicate balance of the biosphere in general.

In view of this and in recognition of the limitations of the early studies, a large effort is currently being devoted to studying this question of possible SST pollution in the stratosphere. As a part of this effort we have developed and are developing several theoretical models to study the governing physical processes in the stratosphere.

We have used equations of the form Eq. (2) to simulate the globally averaged distribution of minor chemical species in the stratosphere. Some examples of the computed distributions are shown in Fig. 2 along with some measurements. Given estimates on the size of the proposed SST fleet, the flight frequency, the fuel consumption rates and the engine emission index, we can estimate the NO source rate, $S_{NO}$, and use Eq. (2) to obtain the range of probable perturbations in the global ozone content. Fig. 3 gives some of the theoretical estimates based on a one dimensional model. A distinction is made between the American SST and the Concorde because they not only have different engines with different emission rates but also different flight altitudes. The latter is especially important since the lower the flight altitude: the shorter the residence time for the pollutant. This will significantly affect the extent of the perturbations as is evident in Fig. 3. It is clear that the predicted perturbations are much smaller than the previously published estimates. At the present it is not clear how we may use these results and other similar predictions to be obtained from more elaborate models in projecting the environmental consequences. Hopefully this will be possible in the near future.

**ATMOSPERIC NUCLEAR TESTS AND OZONE**

In the study of chemical kinetics transport in the stratosphere, validation of the simulation results through comparison with observations is most essential. Unfortunately it is impossible to carry out global scale experiments involving chemical releases. We are forced to search the past for possible means of validation. It has been pointed out [Foley (1973A)] that the atmospheric nuclear tests in the 1950's and early 1960's injected substantial amounts of NO into the stratosphere. It would follow that an examination of the ozone measurements of that period could possibly be used
Figure 4. The model predicted decreases in yearly averaged ozone density due to atmospheric nuclear tests in the 1950's and early 1960's and a trend analysis of ozone measurements.

Figure 5. A comparison of the computed and measured deviations from normal of some representative traces from 1952 to 1970.
to estimate the significance of NO\textsubscript{x} catalytic destruction of ozone in the stratosphere. This past inadvertent NO injection experiment is of great interest provided that we can properly estimate the expected perturbation and the available data. In order to carry out such a study one must use a time dependent model and compute the time history of perturbations from 1956 to 1970 starting with the ambient atmosphere as determined by the model. Based on our solution technique we are able to carry out such a simulation [Chang (1973A)].

By parameterizing the processes of production and injection of NO into the stratosphere during each atmospheric nuclear test, we have constructed a highly discontinuous source function,

\[ S_{NO}(t,z) = \sum_{m=1}^{M} S_m \delta(t-t_m) H(z-z_m^B) H(z_m^T-z) \]

where \( \delta(\cdot) \) is the delta function and \( H(\cdot) \) is the unit step function. \( S_m \) is the total NO produced per unit volume. \( t_m \) is the time of the \( m \)th test. \( z_m^B \) and \( z_m^T \) are the bottom and top of the nuclear cloud for the \( m \)th test. In the one-dimensional simulation, \( S_m \) has been averaged over the surface of the earth. This will introduce significant error locally but negligible error in the global averaged model [Chang (1973A)].

Figure 5 gives the predicted changes in the yearly averaged ozone measurement and a trend analysis of the ozone data in the literature. The data is normalized to the simulation result at 1970. The apparent agreement is still being discussed in the literature since there exists other data analyses which may or may not show the same trend. However, we have another source of unexpected support in the time history of the predicted total excess NO + NO\textsubscript{2} in the stratosphere from the nuclear test. From chemical kinetics theory NO + NO\textsubscript{2} can serve as a tracer in the stratosphere. Figure 6 is a graph of the excess NO + NO\textsubscript{2} and the concentrations of two observed tracers, namely strontium 90 and excess carbon \( 14 \) [Johnston (1973A)]. All data are normalized at 1969 but all the curves are plotted on the same semi-log scale. We feel that the agreement is a validation not only of this study but also of the success of the numerical method.
BIBLIOGRAPHY

1973


Hahn, J., "Nitrous Oxide in Air and Sea Water Over the Atlantic Ocean," presented at the CACGP Symposium on Trace Gases, Mainz, Germany.


1972


Hindmarsh, A. C., "Construction of Mathematical Software; Part III: The Control of Error in the GEAR Package for Ordinary Differential Equations," Lawrence Livermore Laboratory Rept. UCID-30050, Pt. 3.


1971

Crutzen, P. J., "SST's — A Threat to the Earth's Ozone Shield," *Ambio* 1, p. 41.


1970


1968


1965


THE SETS OF SMOOTH SOLUTIONS OF DIFFERENTIAL AND DIFFERENCE EQUATIONS

Germund Dahlquist, Department of Information Processing

The Royal Institute of Technology, Stockholm, Sweden

INTRODUCTION

The following is a report on some recent results of a team-work in progress at the Royal Institute of Technology, sponsored by the Swedish Institute for Applied Mathematics. Some of the results are proved in a report by G. Dahlquist and B. Lindberg (TRITA-NA-7302, Department of Information Processing, Royal Institute of Technology, Stockholm). Other results will be proved and extended in reports by I. Karasalo and/or J. Oppelstrup.

1. SMOOTH SOLUTIONS OF LINEAR ANALYTIC SYSTEMS WITH LARGE LIPSCHITZ CONSTANTS

The smoothness of functions will here be discussed in terms of norms for analytic functions on the unit circle. (The preference of the unit circle is just a matter of scaling the independent variable.) If

\[ y(t) = \sum_{i=0}^{\infty} y_i t^i \]  

(1.1)

then we define the \( p \)-norm

\[ \|y\|_p = \left( \sum_{i=0}^{\infty} |y_i|^p \right)^{1/p}, \quad (1 \leq p < \infty). \]  

(1.2)
When we write \( \| \cdot \| \), we mean one of these norms with \( p=1, p=2 \) or \( p=\infty \). Norms of \( s \)-dimensional (complex) vectors and matrices will be denoted by single bars, \( | \cdot | \), like the moduli of complex numbers.

A function \( y \) is said to be "smooth" if \( \| y \| \) is "not large". In the above-mentioned report of Dahlquist and Lindberg, see also Lindberg (these proceedings), a function \( x \) was said to be smooth at the time \( t_1 \) with respect to the time constant \( \tau \), if for some prescribed constants \( \alpha, \beta \) the following inequalities are satisfied,

\[
| x^{(n)}(t_1) | \leq \alpha \tau^{-n}, \quad n = 0, 1, 2, \ldots, \beta.
\]

For many considerations these notions are almost equivalent, but we now find this terminology formally inconvenient.

Bounds for \( |y(t)|, |y'(t)|, \ldots \) for \( |t| < 1 \), can easily be expressed in terms of \( \| y \| \). A useful relation, which is not quite obvious, is

\[
\| xy \| \leq \| x \| \cdot \| y \| \tag{1.3}
\]

which also holds for vector- or matrixvalued function \( x, y \). Norms of vector- and matrixvalued functions are also defined by Eq.(1.2).

**Lemma 1.1** Let \( \| \cdot \| \) be defined as in the comment to Eq.(1.2), let \( \lambda \) be a complex constant, and

\[
f = \sum_{i=0}^{\infty} f_i t^i.
\]

Then the differential equation

\[
y' - \lambda y = f \tag{1.4}
\]

has a solution such that

\[
\| y \| \leq k_0 |\lambda|^{-1/2} \| f \| \cdot \tag{1.5}
\]

(\( k_0 \) is an absolute constant.)

We omit the proof. This simple equation is, of course, of very little interest in itself, but since it is basic for an iterative construction of smooth solutions for more general systems, we shall dwell on it for a while.

First note that we make no assumption concerning the real part of \( \lambda \). In this section we consider smooth solutions of linear systems with large Lipschitz constants in the unit circle without bothering about their stability properties on the real axis.
Definition: Denote by \( L_\lambda \) a linear operator, such that \( y = L_\lambda f \) is a solution of Eq. (1.4) satisfying the inequality (1.5). \( L_\lambda \) is not uniquely determined by these requirements, but once it has been chosen we call \( L_\lambda f(0) \) the \( L \)-point of Eq. (1.4).

For example the solution of Eq. (1.4) with minimum \( \ell_2 \)-norm can be written, \( y = L_\lambda f \). It can be shown that in this case,

\[
\sup_f \| L_\lambda f \|_2 / \| f \|_2 \sim |2\lambda|^{-1/2}, \quad |\lambda| \to \infty.
\]

(1.6)

This shows that the inequality (1.5) is of the 'best possible' type for large \( |\lambda| \). In fact, for

\[
f(t) = \frac{\exp(\lambda t)}{t}
\]

the value of \( \| L_\lambda f \|_2 / \| f \|_2 \) is very close to the supremum. We know that the inequality (1.5) is generally true with \( k_0 = 1 \).

Another example of an operator \( L_\lambda \) - useful also in the \( \ell_1 \) and \( \ell_\infty \) cases - is defined by the recurrence relation for the Taylor coefficients,

\[
(i+1)y_{i+1} - \lambda y_i = f_i
\]

(1.7)

with the condition \( y_{N+1} = 0 \), where \( N = \text{entier}(|\lambda|) \). This means that the recurrence relation is to be used backwards from \( i = N \) to \( i = 0 \), and forwards for \( i > N \).

Let \( y \) be an arbitrary solution of Eq. (1.4). We then have

\[
y(t) - L_\lambda f(t) = ce^{\lambda t}
\]

\[
\| y \|_1 + \| L_\lambda f \|_1 \geq |c| e^{\lambda t}
\]

\[
|y(0) - L_\lambda f(0)| = |c| \leq (\| y \|_1 + \| L_\lambda f \|) e^{-|\lambda| t}.
\]

(1.8)

This shows that the interval of starting points for solutions of (1.4) with an \( \ell_1 \)-norm of reasonable size is very small when \( |\lambda| \) is large. This gives an excuse for talking about "the" \( L \)-point for Eq. (1.4). Similar inequalities hold for the other norms.)

If \( \| f(k) \| < \infty \), for some \( k \geq 1 \), an expression for \( L_\lambda f(0) \) can be obtained. Put

\[
z = L_\lambda f + \lambda^{-1}f + \lambda^{-2}f' + \ldots + \lambda^{-k}f(k-1).
\]

Then

\[
z' - \lambda z = \lambda^{-k}f(k).
\]
By (1.8)

$$|z(0) - \lambda^{-k} L_\lambda f(k)(0)| \leq (||z||_1 + ||\lambda^{-k} L_\lambda f(k)||_1) e^{-|\lambda|},$$  (1.9)

Hence we obtain an expression for the L-point,

$$L_\lambda f(0) = -\lambda^{-1} f(0) - \lambda^{-2} f'(0) - \ldots - \lambda^{-k} f(k-1)(0) + z(0)$$  (1.10)

where, by (1.9) and (1.5),

$$|z(0)| \leq k_0 |\lambda|^{-k-1/2} (1 + e^{-|\lambda|}) ||f(k)||_1 + e^{-|\lambda|} \left( k_0 |\lambda|^{-1/2} ||f||_1 + \sum_{i=1}^{k} |\lambda|^{-n} ||f(n-1)||_1 \right)$$  (1.11)

The expansion on the right hand side of (1.10) diverges when k→∞, unless f is an entire function of order one or less, but this remainder formula is of general validity.

Finally, we may note that the factor $|\lambda|^{-1/2}$ in (1.5) is not appropriate when $|\lambda|$ is small. It can be replaced by the inequality (see Lemma 3)

$$||y||_1 \leq \min \left( \frac{e|\lambda|^{-1}}{|\lambda|}, k_0 |\lambda|^{-1/2} \right) ||f||_1,$$  (1.12)

but this result can probably be improved.

**Lemma 1.2** Consider the scalar differential equation

$$y'(t) - (\sum_{i=0}^{\infty} q_i t^n) y(t) = f(t).$$

Put

$$n = k_0 |q_0|^{-1/2} \sum_{i=1}^{\infty} |q_i|$$

If $n < 1$, there exists a solution y such that

$$||y||_p \leq \frac{1}{1-n} ||L_{q_0} f||$$

**Proof** Rewrite the equation in the form,

$$y'(t) - q_0 y(t) = f(t) + \left( \sum_{i=1}^{\infty} q_i t^n \right) y(t)$$

Consider the sequence of functions $y^0, y^1, y^2, \ldots$ defined by the relations

$$y^0 = 0, \quad y^{i+1} = L_{q_0} \left[ f + \left( \sum_{i=1}^{\infty} q_i t^n \right) y^i \right].$$
Hence
\[ y^{i+1} - y^i = L_{q_0} \left( \sum_{n=1}^{\infty} n^i n^n \cdot (y^i - y^{i-1}) \right) \]
By the inequalities (1.5) and (1.3),
\[ ||y^{i+1} - y^i|| \leq k_0 |q_0|^{-1/2} \cdot \sum_{n=1}^{\infty} |q_n| \cdot ||y^i - y^{i-1}|| = n ||y^i - y^{i-1}||. \]
The result now follows, since
\[ y^1 - y^0 = y^1 = L_{q_0} f. \]

For non-stiff systems the following result can be proved by ordinary Picard iteration.

Lemma 1.3 Let \( A(t) = \sum_{v=0}^{\infty} A_v t^v \), and put
\[ \alpha = \sum_{v=0}^{\infty} |A_v|, \quad k_1 = \exp(\alpha), \quad k_2 = (\exp(\alpha)-1)/\alpha. \]
Then the solutions of the system
\[ y'(t) - A(t)y(t) = f(t), \]
satisfy the inequality,
\[ ||y|| \leq k_1 |y(0)| + k_2 ||f||. \]

We are now in a position to treat the linear system with analytic coefficients,
\[ d'(t) - J(t)d(t) = r(t) \quad (1.13) \]
Suppose that \( J(0) \) has \( k \) eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_k \), of "large" modulus, \( |\lambda_1| = \min|\lambda_i| \). Assume that \( J(0) \) is brought into block-diagonal form by the matrix \( P \), i.e.
\[ J(0)P = \begin{bmatrix} A(0) \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{bmatrix}, \quad P = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_k \end{bmatrix} \]
Here $P_0$ is a $(s-k) \times s$ matrix, and the $P_i$, $i = 1, 2, \ldots, k$, are row vectors. Put

\[
P_0d = u, \quad P_0r = f
\]

\[
P_i d = v_i, \quad P_i r = g_i, \quad i = 1, 2, \ldots, k
\]

(1.14)

We then write the system (1.13) in the form,

\[
u'(t) - A(t)u(t) = t \sum_{j=1}^{k} b_j(t)v_j(t) + f(t)
\]

(1.15)

\[
v_i'(t) - \lambda_i v_i(t) = t c_i(t)u(t) + t \sum_{j=1}^{k} d_{ij}(t)v_j(t) + g_i(t)
\]

\[(i = 1, 2, \ldots, k)\]

i.e. the system is transformed into

\[
\begin{pmatrix}
u' \\
v'
\end{pmatrix} = \text{PJF}^{-1} \begin{pmatrix} u \\
v
\end{pmatrix} + \begin{pmatrix} f \\
g
\end{pmatrix}, \quad \text{where}
\]

\[
\begin{align*}
\text{PJF}\begin{pmatrix} A(t) \\
\end{pmatrix} &= \begin{pmatrix} \text{column vectors } t \cdot b_j(t) \\
\end{pmatrix} \\
\text{k \times k matrix } &= \text{diag}\{\lambda_i\} + t\{d_{ij}\} \\
\text{row vectors } t \cdot c_i(t)
\end{align*}
\]

An \textit{L-solution} of this system is defined as the limit functions,

\[
u = \lim u_n, \quad v_i = \lim v_{n,i}, \quad \text{of the following iterative process, if it converges. An initial condition is prescribed for } u \text{ only.}
\]

\[
u_0(t) - A(t)u_0(t) = f(t), \quad u_0(0) = u(0),
\]

and

\[
v_{0,i}(t) = L_{\lambda_i} \left(tc_i(t)u_0(t) + g_i(t)\right), \quad i = 1, 2, \ldots, k.
\]

Then, for \(n = 1, 2, 3, \ldots, u_n\) is the solution of the initial value problem,

\[
u_n'(t) - A(t)u_n(t) = f(t) + t \sum_{j=1}^{k} b_j(t)v_{n-1,j}(t), \quad u_0(0) = u(0)
\]

and

\[
v_{n,i}(t) = L_{\lambda_i} \left(tc_i(t)u_{n-1}(t) + t \sum_{j=1}^{k} d_{ij}(t)v_{n-1,j}(t) + g_i(t)\right).
\]

(1.17)
The \( L \)-set is defined as the set of possible starting points \((\bar{u}(0), \bar{v}(0))\) of \( L \)-solutions. The \( L \)-point of the system is the starting point for which \( \bar{u}(0) = 0 \). In general \( \bar{v}(0) \neq 0 \) for the \( L \)-point of an inhomogeneous system, (see Eq.(1.20)). The \( L \)-space is the \( L \)-set of the homogeneous system corresponding to (1.15), i.e. a system where \( f = 0, g_i = 0, i = 1,2,..., k \). The dimension of the \( L \)-space is \( s-k \); the vector \( u(0) \) is arbitrary. The \( L \)-set is always the sum of the \( L \)-point and the \( L \)-space.

Introduce the row vector \( B \), the column vector \( C \) and the square matrix \( D \),

\[
B = \{ k_2 \| t_{b,j} \| \}, \quad C = \{ k_0 \| \lambda_i^{-1/2} t_{c,i} \| \}, \quad D = \{ k_0 \| \lambda_i^{-1/2} t_{d_{ij}} \| \}
\]  

By application of Lemma 1, Lemma 3 and inequality (1.3) one obtains after some computation,

\[
\begin{pmatrix}
\| u_{n+1} - u_n \| \\
\| v_{n+1,i} - v_{n,i} \|
\end{pmatrix} \leq
\begin{pmatrix}
0 & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
\| u_n - u_{n-1} \| \\
\| v_{n,i} - v_{n-1,i} \|
\end{pmatrix}
\]

The following result is then obtained after a lengthy computation.

**THEOREM** Suppose that the linear analytic system (1.13) is brought to the form (1.15) by the transformations in Eq.(1.14). Assume that the spectral radius of \( D \) is less than one (see Eq.(1.18)) and that

\[
\eta := B (I-D)^{-1} C < 1.
\]

Then the iterative process defined by Eqs.(1.16) and (1.17) converges, for arbitrary values of the \((s-k)\)-dimensional vector \( u(0) \).

Put

\[
k_3 = \frac{B(I-D)^{-1} \{ \| v_{0,i} \| \}}{1-\eta}.
\]

The \( L \)-solution \( \bar{u}, \bar{v} \) satisfies the inequalities

\[
\| \bar{u} \| \leq \| u_0 \| + k_3 \leq k_1 \| u(0) \| + k_2 \| f \| + k_3
\]

\[
\{ \| \bar{v}_{-i} - v_{0,i} \| \} \leq D(I-D)^{-1} \{ \| v_{0,i} \| \} + k_3(I-D)^{-1} C.
\]

(The parameters \( k_1, k_2 \) were defined in Lemma 3.)
In the system (1.15), consider the $\lambda_i$ as large quantities, while the other coefficient functions are not large. Let $u, v$ be a smooth solution of this system. (By the theorem the L-solution is one example.) Then

$$v_j^{(n)}(0) \approx \lambda_j^{-1} \left( g_j^{(n)}(0) + a_j^{(n)} u(0) + b_j^{(n)} \right)$$

(1.19)

where the $a_j^{(n)}, b_j^{(n)}$ are uniquely determined by the first $n$ Taylor coefficients of the functions $A, f$ and $c_j$. (The $a_j^{(n)}$ are independent of $f$.) In particular, we obtain the following approximate formulas for the L-set ($u(0)$ arbitrary);

$$v_j^{(1)}(0) \approx -\lambda_j^{-1} \left( g_j^{(1)}(0) + c_j(0) u(0) \right)$$

$$v_j(0) = -\lambda_j^{-1} g_j(0) + \lambda_j^{-1} v_j^{(1)}(0)$$

(1.20)

Note that

$$v_j(0) \approx -\lambda_j^{-2} c_j(0) u(0)$$

for a homogeneous problem. This shows that the L-space is generally not identical to the subspace spanned by the eigenvectors belonging to the small eigenvalues of $J(0)$, though the angle between the subspaces tends to zero, as $|\lambda_1| \to \infty$.

More accurate approximations (similar to Eq.(1.10) can be obtained by using the recurrence relations for the Taylor coefficients backwards for the $v_i$ and forwards for $u$.

The inequalities

$$|v_i^{(N)}(0)/N!| \leq ||v_i||$$

are then needed for some value of $N$, $N > n$. For the L-set bounds for the $||v_i||$ are obtained by the theorem and Lemma 1.

For a given decision about the grouping of the eigenvalue moduli into "large" ones and "small" ones. The two assumptions in the theorem can always be fulfilled by an appropriate decision about the scaling of the independent variable. (If the new unit is equal to $R$ old units, then $\eta$ and the spectral radius of $D$ are to be multiplied by $R^{7/2}$ and $R^{3/2}$, respectively.) These decisions are also restricted by the requirements that the parameters $k_1$ and $k_2$ (approximately equal to $\exp(R \|A\|_1)$ after scaling) should be of reasonable size and that the value at $t=0$ of any solution with a norm of reasonable size should not be far from the L-set. (Suppose some kind of generalization of inequality (1.8) to systems.) The combination of these requirements seems to imply a lower bound for the difference (after scaling) between the smallest among the "large" moduli of eigenvalues and the largest
among the "small" moduli of eigenvalues. In rare cases, this may make the theory hard to apply. In investigations of numerical methods these decisions are also related to the choice of step-size, because those components which are not reproduced with a good relative accuracy in a numerical solution with a certain step-size should be associated with "large" eigenvalues.

2. APPLICATION TO THE IMPLICIT MIDPOINT METHOD WITH RICHARDSON EXTRAPOLATION

Our attempts to extend the theory to analytic non-linear systems are not yet ready for reporting. With a few additional assumptions, we can, however, apply our results for the linear problem to the study of a numerical method for solving stiff, analytic non-linear systems.

Consider an analytic, non-linear system,

\[ y' = f(y), \]  

and assume that \( \bar{y}(t) \) is a particular solution, for which \( \| y \|, \| y' \|, \| y'' \|, \ldots \) are not large. Put

\[ J(t) = f'(\bar{y}(t)) \]

\[ |J(t)| \text{ is large, but assume that } |f''(\bar{y}(t))| \text{ is not large.} \]

Further assume that the variational equation,

\[ p' = J(t)p \]

satisfies the conditions of the theorem, and that \( |J'| \) is not large. In particular, \( |J'(t)|, |J''(t)|, \ldots \) are not large. By differentiation of (2.1), we obtain, \( y'' = f'(y)y', \) i.e. \( y'(t) \) is a smooth solution of the variational equation. By (1.19) and (1.20) we then obtain (using the notation of Eq.(1.14)),

\[ P_j\bar{y}'(0) \approx -\lambda_j^{-2} c_j(0) P_0\bar{y}'(0) \]

\[ (n>0), \quad P_j\bar{y}(n+1)(0) \approx -\lambda_j^{-1} \alpha_{jn} P_0\bar{y}'(0) \]

Until now, we have considered smooth solutions in the complex plane, without bothering about their stability properties for real, increasing \( t \). Now look upon \( t \) as a real variable. Suppose that an inner product \( \langle \cdot, \cdot \rangle \) is defined in \( \mathbb{R}^8 \), and put \( |z|^2 = \langle z, z \rangle \). Assume that this norm is a Liapunov function for the solution \( y(t) \) of (2.1). Consider the equation,
\[ z' = f(\tilde{y}(t) + z) - f(\tilde{y}(t)) \]

Since
\[ \frac{d|z|^2/\mathrm{d}t}{2} = 2 \langle z, dZ/dt \rangle = 2 \int_0^1 \langle z, f'(\tilde{y}(t) + \theta z) \rangle \, d\theta, \]
we see that a sufficient condition for this is that, for some \( \rho > 0 \),
\[ \langle z, f'(y)z \rangle \leq 0, \quad y \in S_\rho, \quad \forall z \in R^s, \] (2.5)
where \( S_\rho = \{ y: |y - \tilde{y}(t)| \leq \rho \} \). We assume that this holds.

Consider the application of the implicit midpoint method to (2.1),
\[ y_{n+1} - y_n = h f\left( \frac{y_n + y_{n+1}}{2} \right) \] (2.6)
Let \( t_0 = 0 \) be the starting time, or a time where a new step-size \( h > 0 \) is chosen. Put
\[ t_v = vh, \quad J(t) = f'(\tilde{y}(t)). \]
then the numerical solution can (in several ways) be decomposed as follows:
\[ y_n = \tilde{y}(t_n) + h^2 \tilde{d}(t_n) + w_n \] (2.7)
where \( \tilde{d}(t) \) is an \( L \)-solution of the differential equation,
\[ d'(t) - J(t)d(t) = \frac{1}{12} \tilde{y}'''(t) - \frac{1}{8} J'(t)\tilde{y}(t) \] (2.8)
and \( \{w_n\} \) satisfies the difference equation,
\[ w_{n+1} - \frac{w_n}{h} = J_{n+\frac{1}{2}} \left( \frac{y_n + y_{n+1}}{2} \right) + p_n, \] (2.9)
where
\[ J_{n+\frac{1}{2}} = \int_0^1 f\left( \frac{y_n + y_{n+1}}{2} - \theta \frac{w_n + w_{n+1}}{2} \right) \, d\theta, \]
\[ p_n = h^5 \left[ \frac{\tilde{d}'''}{12} - \frac{\tilde{y}'}{120} - \frac{1}{3 \delta t^4} (\tilde{y}' - J\tilde{y}') - \frac{1}{8} (\tilde{d}'' + J\tilde{d}'') - \frac{\tilde{y}'}{12} \right] \]
\[ + \frac{1}{2} f''(\tilde{y}) (\tilde{d} + \frac{\tilde{y}'}{8})^2 + O(h^7) \] (2.10)
The expressions in Eq. (2.8) and (2.10) are obtained by coefficient matching in an expansion into powers of \( h \). A similar decomposition is possible for many other methods — one can also have more terms.
in (2.7). The following lemma is not so easily proved for more complicated methods, see however also Stetter (1973), Theorem 3.5.8.

**Lemma 2.1** If (2.2) is satisfied, then
\[ |w_m| \leq |w_0| + \sum_{v=0}^{m-1} |p_v| \]
as long as
\[ |w_0| + \sum_{v=0}^{m-1} |p_v| + h^2|\dd(t_m)| < \rho, \quad h = 0, 1, 2, \ldots \]

**Proof** For m=0 the result is trivially true, since
\[ \sum_{0}^{-1} |p_v| \]
is to be interpreted as a vanishing sum. Suppose that the result is true for m=n. We shall show that it is true for m = n+1.

Concerning the existence of a solution of (2.6) in \( S_\rho \), the reader is referred to an analogous discussion in Dahlquist (1963), p. 38. It follows that
\[ \frac{\gamma_{n+1} + \gamma_n}{2} - \theta \frac{w_{n+1} + w_n}{2} \in S_\rho \]
for \( 0 \leq \theta \leq 1 \) and so, by (2.5) and the definition of \( J_{n+\frac{1}{2}} \),
\[ \langle z, J_{n+\frac{1}{2}} \rangle \leq 0, \quad \forall z. \]

Multiply (2.9) by \( w_{n+1} + w_n \),
\[ \langle w_{n+1} + w_n, w_{n+1} - w_n \rangle = \frac{h}{2} \langle w_{n+1} + w_n, J_{n+\frac{1}{2}}(w_{n+1} + w_n) \rangle + \langle w_{n+1} + w_n, p_n \rangle \leq 0 + |w_{n+1} + w_n| \cdot |p_n|. \]

Hence,
\[ |w_{n+1}|^2 - |w_n|^2 \leq (|w_{n+1}| + |w_n|) \cdot |p_n| \]
\[ |w_{n+1}| - |w_n| \leq |p_n|, \]
which proves the Lemma.

Now, consider the right hand side of Eq.(2.10). It follows from the assumptions made that the derivatives of \( \dd \) and \( \dd \) are not large for \( |t| < 1 \). It is less obvious that \( \dd \dd \dd \) and \( \dd \dd \dd \dd \) are not large but, by Leibniz' rule,
\[ J\ddot{d} = \frac{d^2}{dt^2} (J\dot{d}) - 2J'\ddot{d}' - J''\ddot{d}. \]

By Eq.(2.8) the right hand side can be expressed in terms which are not large. A similar calculation can be carried out for \( J\dot{y}'' \).

The terms hidden under the \( O(h^7) \) cover in Eq.(2.10) are also not large.

It now follows that
\[
\sum_{0}^{n-1} |p_v| = O(h^4) \quad (\text{if } n \cdot h = O(1)),
\]
but in order to make a conclusion about the order of magnitude of \( (w_n) \) we have, by Lemma 2.1, to consider \( w_0 \) also, and this is more problematic! We shall discuss this in geometrical terms.

Denote the L-point of the system (2.8) by \( \hat{d}(0) \).
The set \( L(h) \) for the difference equation (2.6) at \( t=0 \) is defined as the sum of the point
\[ \gamma(0) + h^2\hat{d}(0) \]
and the L-space of the system (2.8).

By (2.7)
\[ w_0 = y_0 - \gamma(0) - h^2(\hat{d}(0) + p), \quad p \in L\text{-space}. \]

It follows that it is possible to have \( w_0 = O(h^4) \) if and only if the distance of \( y_0 \) from \( L(h) \) is \( O(h^4) \). \( L(h) \) can therefore be regarded as an \( O(h^4) \)-accurate approximation to the starting set of smooth solutions of the difference equation (2.6), close to \( y(0) \).

Consider the natural choice \( y_0 = \gamma(0) \). If \( P_0w_0 = 0 \), then \( P_0P = 0 \) and hence \( p=0 \). After a straight-forward application of (2.8), (1.20) and (2.4) we obtain the result,
\[ P_jw_0 = -h^2P_j\hat{d}(0) \approx -\frac{1}{8}h^2\lambda_j^{-1} c_j(0)P_0\gamma'(0) \]
\[ j = 1, 2, \ldots, k \]
(2.11)

This is the initial amplitude of a transient which has an oscillatory behaviour, roughly like
\[ \left( 1 + \frac{1}{2}h\lambda_j \right)/(1 - \frac{1}{2}h\lambda_j) \]

Such oscillating error components are not well suited to Richardson extrapolation, though for several reasons the stiffest components are not the worst. Note for instance the factor \( \lambda_j^{-1} \) in Eq.(2.11). Furthermore the amplitude is practically constant (for even \( n \)) as long as \( |4n/(\lambda h)| \ll 1 \). Hence this error component is approximately proportional to \( h^2 \), and it can therefore be
reduced by extrapolation! On the other hand, if $|\lambda_j h|$ is not large, $|\lambda_j h| < \frac{4}{3}$ say, the oscillations decrease quickly. The transients corresponding to values of $|\lambda_j h|$ between these extremes are, however, not reduced by extrapolation, and because of this, smoothing is needed, if a good approximation to $\dot{y}(0)$, obtained by Richardson extrapolation, is used as the starting point after a change of step-size.

There is also another possibility for choosing starting point. Suppose that step-sizes $h$ and $h/2$ have been used up to the time $t=0$, after which step-sizes $H$ and $H/2$ are to be used. Denote the results obtained by $y_{0-}(h)$ and $y_{0-}(h/2)$ respectively, and suppose that $y_{0-}(h) \in L(h)$, $y_{0-}(h/2) \in L(h/2)$.

We want to find starting points $y_{0+}(H) \in L(H)$, $y_{0+}(H/2) \in L(H/2)$.

Since $\dot{y}(t)$ is a smooth solution of the homogeneous variational equation, then

$$y_{0-}(h) - a\dot{y}(0)$$

is close to $L(h)$ for any scalar $a$.

We want also to eliminate the component of the error in the direction of $\dot{y}(0)$, which sometimes is an important part of the error. Put

$$r = \frac{h}{3} \left[ y_{0-}(h) - y_{0-}(h/2) \right], \quad p = \dot{y}(0)$$

$$\alpha = \langle p, r \rangle / \langle p, p \rangle,$$

with some suitable definition of inner-product. Then the formula required reads

$$y_{0+}(H) = y_{0-}(h) - \alpha p + ((H/h)^2-1)(r-\alpha p)$$

(2.12)

and similarly for $y_{0+}(H/2)$. The dangerous thing is the amplification (if $H \gg h$) of noise components in $r - \alpha p$, which must be compensated by the use of smoothing, for example by the formula

$$y_n = y_n - \frac{1}{16} \delta^4 y_n.$$

This device worked well on a test problem which had caused trouble for earlier versions of the extrapolation method. It must be emphasized, however, that we have not yet had a chance to test it systematically. It is only mentioned as an example of a possible application of the introduced concepts to the design of algorithms.

The set $L(0)$ can be interpreted as a tangential $L$-set for the
equation } y' = f(y) \text{ at } y = \tilde{y}(0).

The reason for the creation of transients at step-size changes is that the L-point of (2.8) is not zero and hence \( L(h) \neq L(0) \). It is interesting that for linear multistep methods, the right-hand side of the equation corresponding to (2.8) is of the form \( cy_{(p+1)}(t) \), and by (2.4) \( P_j^T \tilde{y}(j+1)(0) \approx -\lambda_j^{-1} a_j P_0 \tilde{y}'(0) \).

Hence, by (1.20),

\[
P_j w(0) = -h P_j d(0) \sim -ch \lambda_j^{-1} P_0 y_{(p+1)}(0) + \mathcal{O}(\lambda_j^{-2}) \sim k \lambda_j^{-2} h P_0 y_{(p+1)}(0),
\]

whence the oscillations are \( \mathcal{O}(\lambda_j^{-2}) \) initially instead of \( \mathcal{O}(\lambda_j^{-1}) \) as for the implicit midpoint method.

For the trapezoidal method this advantage counteracted by the possibility of the growth of oscillations, see Gourlay (1969), Stetter (1973). The analog of Lemma 2.1 here only guarantees that

\[
| (I - \frac{hJ_n}{2}) w_n | \leq | (I + \frac{hJ_0}{2}) w_0 | + \sum |p_v|,
\]

and it may happen that

\[
| (I - \frac{1}{2}hJ_n) w_n |^{-1} \ll | I + \frac{1}{2}hJ_0 |.
\]

We hope that the concepts introduced here can be of some use for the understanding of differential equations with large Lipschitz' constants and their numerical treatment. We have only considered the L-space at a specified point \( t=0 \). Of course it can be defined for any \( t \). We hope that a \( t \)-dependent linear transformation of a linear system, based on the local partitioning into the L-space and some appropriate complement can become a more powerful device than the local diagonalization of the coefficient matrix.
1. INTRODUCTION

One important application area of digital simulation of dynamical systems is the field of chemical kinetics. The problem to be treated here concerns the simulation of chemical reaction kinetics in homogeneous solutions according to the mass action law. The subject is to study the concentrations versus time of the components involved in the system by solving the ordinary differential equations (ODE's) describing the kinetics with some numerical method. Numerical problems in these kinds of systems especially concern stiffness and size, i.e. the number of ODE's involved. The aim is to utilize the special structure in order to gain efficiency. The efficiency aspect is important also for systems of moderate size when you want the solution for several different sets of parameters involved in the system, e.g. rate constants or initial concentrations. This situation occurs e.g. in the estimation of parameters from empirical curves and in the problem of verification of a hypothetic mechanism for a system of reactions, i.e. the identification problem.

The literature on the subject is vast, surveys can be found in e.g. Crossley and Slifkin (1970 ), Grover (1967 ), Mentor (1969 ) for chemical applications and Garfinkel, Garfinkel, Pring, Green and Chance (1970 ) for biochemical applications. The problem of stiffness in the numerical solution of the ODE's describing the kinetics was perhaps first pointed out by Curtiss and Hirschfelder (1952 ), who treated a reaction with radicals involved. The stiffness in these reactions as well as in those with quasistationary species is caused by the reactions taking place in several elementary
steps the rate constants of which are widely separated in magnitude.

One special example which has become very famous because it is often used as a test example for stiff methods is a system of three non-linear ODE's describing the kinetics of an autocatalytic reaction given by Robertson (1966). The structure of the reactions are

\[ k_1 \]
\[ A \rightarrow B \]
\[ k_2 \]
\[ B + C \rightarrow A + C \]
\[ k_3 \]
\[ 2B \rightarrow C + B \]

and the corresponding system of ODE's is

\[ \dot{x}_1 = -k_1 x_1 + k_2 x_2 x_3 \quad x_1(0) = 1 \]
\[ \dot{x}_2 = k_1 x_1 - k_2 x_2 x_3 - k_3 x_2^2 \quad x_2(0) = 0 \quad (1) \]
\[ \dot{x}_3 = k_3 x_2^2 \quad x_3(0) = 0 \]

where \( x_1 = C_A, x_2 = C_B, x_3 = C_C \) denoting the concentration of a component, \( k_i \) is a rate constant and \( \cdot \) denotes differentiation with respect to time. The numerical values of the rate constants are

\[ k_1 = 0.04, \quad k_2 = 10^4 \quad \text{and} \quad k_3 = 3 \cdot 10^7. \]

The system is very stiff as can be seen from the variation of the Jacobian eigenvalues along the solution trajectory:

<table>
<thead>
<tr>
<th>t</th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.04</td>
</tr>
<tr>
<td>( 10^{-2} )</td>
<td>0</td>
<td>-0.36</td>
<td>-2180</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-0.0048</td>
<td>-4240</td>
</tr>
<tr>
<td>( \infty )</td>
<td>0</td>
<td>0</td>
<td>-10^4</td>
</tr>
</tbody>
</table>

As is typical for problems arising in chemical kinetics this special system has a small very quick initial transient. This phase is followed by a very smooth variation of the components where a long stepsize would be appropriate for a numerical method. One of the usual explicit methods, e.g. Runge-Kutta, however, requires a stepsize of order \( 10^{-4} \) for this special problem, which is unacceptable for efficiency reasons as the solution is of interest for a time interval of order \( 10^3 \).
Further examples and references can be found in Bjurel, Dahlquist, Lindberg, Linde and Odén: Survey of stiff ordinary differential equations, report NA 70.11, Department of Information Processing Computer Science, The Royal Institute of Technology, Stockholm.

Before implicit methods suited for stiff equations were implemented by e.g. Gear (1971B) and Lindberg (1973)* stiff problems were attacked either by explicit methods such as Euler or Runge-Kutta or an explicit method combined with "steady state" approximations, see e.g. De Los (1967A), (1967B), Garfinkel (1968), Laurendeau and Sawyer: General reaction rate problems: combined integration and steady state analysis, rep. no TS-70-14, College of Engineering, Univ. of California, Berkeley and Snow (1966). In recent years, however, program packages for simulation, similar to the one to be described here, have been developed by e.g. Curtis and Chance (1972). Efforts have also been made to construct programs for parameter estimation. see e.g. Curtis and Chance (1972) and Hemker (1972B).

Steady state approximations have been widely used by chemists to overcome the stiffness problem. In a general formulation the method works as follows: Let the system of ODE's be

\[
\dot{x} = f(x) \; ; \; x(0) \text{ given} \tag{2}
\]

Some of the components \(x_i\) quickly reach a stage when the derivatives \(\dot{x_i}\) are very small compared to the individual terms on the right hand side, so the system (2) can be approximated by

\[
\dot{y} = f_1(y,z) \; ; \; y(0) \text{ given} \tag{3}
\]

\[
o = f_2(y,z)
\]

where \(y, z\) is a partition of \(x\) so that \(z\) corresponds to the components which have reached steady state while \(y\) corresponds to the slow components. If the partition is chosen properly the system (3) can be solved with an explicit method combined with an iterative procedure for solving the non-linear equations in (3). If this method is used on Robertson's example (1) we get

\[
\begin{align*}
\dot{x}_1 &= -k_1x_1 + k_2x_2x_3 \\
\dot{x}_3 &= k_3x_2^2 \\
o &= k_1x_1 - k_2x_2x_3 - k_3x_2^2
\end{align*}
\tag{4}
\]

if \(x_2\) is supposed to be in a steady state from \(t = 0\). The eigenvalues of the Jacobian then vary with time as follows:

*These Proceedings.
\begin{align*}
  & \lambda_1 & \lambda_2 \\
  t = 0 & 0 & 0.41 \\
  t = 10 & 0 & -0.06 \\
  t = \infty & 0 & 0
\end{align*}

so an explicit method would do excellently on this system. However, there are some disadvantages to the steady state method:

1) If the steady state approximation is used from $t = 0$ the initial value $\bar{x}(0)$ in (2) may not be consistent with $\bar{f}_2(\bar{y}(0), \bar{z}(0)) = 0$, so the initial value for $z$ must be modified in some way. If the whole system (2) is integrated from $t = 0$ and the steady state approximations are applied successively according to some numerical criterion for steady state, however, this problem can be overcome.

2) In the ODE's describing the kinetics there are linear relations between the $x_i$-variables such as $Lx = \text{const.}$ for all values of $t$ due to conservation laws, e.g. conservation of mass. These linear relations in the original system (2) will not be preserved when the approximation (3) is used.

3) The partitioning (3) is not always possible to state a priori, i.e. the criterion for steady state must be examined by some numerical test which must be performed during the integration in order to see which variables $x_i$ can be supposed to belong to $z$ in (3).

4) The errors of the approximation can be disastrous especially in the transient, see e.g. Gelinas (1972). These observations are confirmed by Edelson (1973).

If a numerical method is used on the whole system (2) the above disadvantages are avoided. The problems when using an explicit method are well-known so an implicit method should be used and then we have the bulk of computational work when solving the system of non-linear equations by a quasi-Newton method in each step. However, as is discussed by Dahlquist (1973), the amount of work can in fact be comparable to solving (3) with an explicit method if the iterative scheme is designed appropriately. As the system of ODE's for these problems has a special structure it is also worthwhile to look for "short-cuts" in the computational implementation. These possibilities will be discussed in ch. 4 of this paper.
2. STATEMENT OF THE PROBLEM

When the elementary steps of some composite reaction taking place in a homogeneous solution according to the mass action law are known, the system of ODE's describing the kinetic behaviour of the system can be set up from the coefficients and the structure of the reactions, see e.g. Mentzen (1969) and Snow (1966). Suppose a reaction takes place in \( N \) elementary steps where the \( j \)-th step is written:

\[
\begin{align*}
\text{\( r \)}_j \text{G}_1 + \text{\( r \)}_j \text{G}_2 + \ldots + \text{\( r \)}_j \text{G}_M & \rightarrow \text{\( q \)}_j \text{G}_1 + \text{\( q \)}_j \text{G}_2 + \ldots + \text{\( q \)}_j \text{G}_M,} \\
\end{align*}
\]

where \( \text{\( r \)}_j, \text{\( q \)}_j \) are coefficients (integers), \( \text{G}_i (i = 1, 2, \ldots, M) \) are symbols for the components of the system taking part in the reaction and \( k_j \) is the rate constant which is constant if the reaction is held at a fixed temperature. The system of ODE's is then:

\[
\begin{align*}
\dot{x} &= Ap ; \quad x(0) \text{ given} \quad (5), \text{ where} \\
- x &= \text{an \( M \times 1 \) state vector whose elements are the concentrations of the chemical components in the system, expressed in appropriate units,} \\
A &= \text{an \( M \times N \) matrix with integer entries: \( A_{ij} = q_{ji} - r_{ji} \),} \\
p &= \text{an \( N \times 1 \) vectorvalued function of the concentrations \( x_i \) with elements} \\
& \quad \text{\( p_j = k_j \prod_{i=1}^{M} x_i \) (rate functions) and} \\
-x(0) &= \text{an \( M \times 1 \) vector of initial concentrations.} \\
\end{align*}
\]

Ex 1. See Robertson's example (1).

\[
\begin{align*}
\begin{array}{c}
k_1 \\
k_2
\end{array} \\
\end{align*}
\]

Ex 2. The system \( A \rightarrow B \rightarrow C \) is governed by

\[
\begin{align*}
\dot{x}_1 &= -k_1 x_1 \quad & x_1 &= C_A \\
\dot{x}_2 &= k_1 x_1 - k_2 x_2 \quad & x_2 &= C_B \\
\dot{x}_3 &= k_2 x_2 \quad & x_3 &= C_C \\
\end{align*}
\]

This is an example of a first order reaction which is described by a linear differential equation. A thorough treatment of the mathematical properties of these systems can be found in Wei and Prater (1962).
As was stated earlier there are linear relations between the dependent variables. In ex.1 we have $x_1 + x_2 + x_3 = \text{const}$, a relation which holds for ex.2 as well. In fact, referring to the general problem (5), if the matrix $A$ has rank $R$ we have the following relations:

$$PA = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} ; \quad A_2 = LA_1$$

(6)

where $A_1$ is an $R \times N$ matrix, $A_2$ is an $(M-R) \times N$, $L$ is an $(M-R) \times R$ and $P$ is a permutation matrix. So if $A$ in (5) is permuted according to (6) we have:

$$\begin{align*}
\frac{\dot{x}_1}{\dot{x}_2} &= A_1P \\
\frac{\dot{Lx}_1}{\dot{x}_2} &= x_2(0) + L(x_1 - \bar{x}_1(0)),
\end{align*}$$

where $\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$ is a partition according to (6).

These linear relations thus lower the order from $M$ to $R$, whence we solve the system

$$\begin{align*}
\frac{\dot{x}_1}{\dot{x}_2} &= A_1P ; \quad \bar{x}_1(0) \text{ given} \\
\bar{x}_2 &= \bar{x}_2(0) + L(\bar{x}_1 - \bar{x}_1(0))
\end{align*}$$

(7)

and this reduction can be of importance for systems of high order as for the efficiency aspect. This technique of eliminating the linear relations between the variables also has an important consequence concerning the conditioning of the nonlinear system of algebraic equations to be solved in each step. When we use the implicit midpoint rule on the system (2) we obtain the difference equation

$$\bar{x}_{n+1} - \bar{x}_n = h\bar{f}\left(\bar{x}_{n+1} + \bar{x}_n\right)/2$$

(8)

to be solved for $\bar{x}_{n+1}$. For stiff problems this is done by a quasi-Newton method, so we need an approximation of the inverse of

$$I - \frac{h}{2} \frac{\partial \bar{f}}{\partial \bar{x}},$$

see Lindberg’s paper. For the special problems treated in this paper it is not unusual that the diagonal elements of $\partial \bar{f}/\partial \bar{x}$ become so large that $I$ is shifted out in computational calculations. Applying this fact to (5) we see that
\[ I - \frac{h}{2} \frac{\partial^2 p}{\partial x} \approx - \frac{h}{2} A \frac{\partial p}{\partial x} \]

which, however, is singular due to the rank deficiency of \( A \), so the stepsize \( h \) must be reduced in order to avoid the singularity of

\[
\left( I - \frac{h}{2} A \frac{\partial p}{\partial x} \right).
\]

By working with (7) we avoid this problem since \( A \) has full rank.

3. THE PROGRAM PACKAGE

The program package, for the time being implemented only on a Hewlett-Packard 2000A time-sharing BASIC system connected to a Tectronix 4010-1 display, makes it possible to perform interactive simulation of the kinetics of a chemically reacting system described in ch.2. The package is divided into three parts:

1) A computer that translates the reactions into the system of ODE's describing the kinetics. The nonzero elements of \( q_{ji} - r_{ji} \) and \( r_{ji} \) are extracted from the reactions and the matrix \( A \) in (5) is factorized according to (6). The information about the structure of the reactions, sufficient to set up the system of ODE's, is stored in a matrix \( L \) and the elements \( q_{ji} - r_{ji} \) and \( r_{ji} \).

2) The information from part 1 is used to build up subroutines that generate the right hand sides \( A_1 P \) of (7) and the corresponding Jacobian

\[
A_1 \left( \frac{\partial p}{\partial x_1} + \frac{\partial p}{\partial x_2} \cdot L \right).
\]

These are needed for the integration method which is the implicit mp rule with smoothing and extrapolation, as developed by Lindberg herein. Additional information needed to complete the problem such as time interval of integration, local error bound, rate constants and the initial concentrations are given whereupon the integration starts.

3) A graphical routine is used to plot the concentrations of the components versus time on the display.

Ex. The following example is taken from Villadsen, Lassen: Integration of the rate equation in chemical kinetics by an orthogonal collocation method. Inst. for Kemiteknik Dth (1968).

The reactions are:

\[
\begin{align*}
A + B & \overset{10^2}{\rightarrow} C \\
2B & \overset{10^4}{\rightarrow} D
\end{align*}
\]
with \( C_A = C_B = 1, C_C = C_D = 0 \) at \( t = 0 \).

The interaction with the computer is as follows: The dialogue between the computer (C) and the user (U) is written in capital letters.

**Comments**

A reversible reaction is counted as 2

* denotes the end of the string.

Arbitrary characters may be used in the names of the components except for \(+, \, *, \) and \( = \). Nor may the names begin with an integer, because it is interpreted as a stoichiometric coefficient. For a reversible reaction the symbol = may be used.

The computer lists the names of the components given by the user in the order they entered by the reactions. The matrices A and L are printed as a check. The column to the left of A is the permutation vector.

End of first part.

Further explanations of how the stepsize strategy depends on the error bound given is found in Lindberg's paper.

The order of the components may be changed by the permutation in part 1.

By this information the stepsize strategy is arranged such that the numerical values of the components get about the same relative error. If nothing is known of the order of the components default values should be 1.
The values from the integration are stored on a file for graphical output whether a table is wanted or not.

The Lipschitz constant gives an idea of the stiffness of the problem. The stepsize printed is the initial stepsize used by the integration. This stepsize is successively increased during the integration.

By choosing different time intervals you can study different parts of the trajectory.

If scales are given appropriately the y-axes is utilized optimally for each component.

See next page for plots.

The part of the program that automatically translates the chemical reactions in the matrix $A$ and the powers $r_{ji}$ which uniquely define the problem operates in the following way:

Every reaction is treated as a string which is examined with respect to the coefficients and the components involved. When reaction number $i$ is given the $i$:th column of $A$ obtains its nonzero entries $q_{ji} - r_{ji}$ in row number $j$. So each chemical component encountered must be compared with the earlier ones, which are stored in a reference string where they are represented by an internal number according to their position in the string. The coefficients $r_{ji}$ determine the rate functions except for the factor $k_j$, and they are stored in the following way:
If the time interval is 0, 0.001 we get the following graph:
INTEGRATION PACKAGE FOR CHEMICAL KINETICS

\[ C_j = \sum_{i=1}^{M} r_{ji} \]

\[ \begin{array}{ccccccc}
C & C_1 & C_2 & \cdots & C_j & \cdots & C_N \\
\end{array} \]

\[ \begin{array}{ccccccc}
R & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
C_1 & C_2 & \cdots & C_j & \cdots & \cdots & \cdots \\
\end{array} \]

R stores the indices of the components in the rate functions each index repeated \( r_{ji} \) times. From this compact way of storing the powers, it is easy to calculate the rate functions \( p_j \). Let \( K[j] \) denote the rate constants. In order to show the technique, in ALGOL-notation, we then compute the \( p_j \)'s with the following statements.

\[
\begin{align*}
s &:= 0; \\
\text{for } j &:= 1 \text{ step 1 until } N \text{ do} \begin{align*}
P[j] &:= K[j]; \\
\text{for } l &:= s+1 \text{ step 1 until } s+C[j] \text{ do} \begin{align*}
P[j] &:= P[j] \times X[R[l]]; \\
s &:= s+C[j]
\end{align*}
\end{align*}
\end{align*}
\]

The calculation of \( \partial p/\partial x \) needed in the Jacobian does not take much work either.

As \( p_j = k_j \prod_{i=1}^{M} x_i^{r_{ji}} \) we have \( \frac{\partial p_j}{\partial x_i} = r_{ji} x_i^{r_{ji}-1} \). So we need \( C = \sum_{i=1}^{N} C_i \) multiplications to calculate \( \tilde{p} \) and approximately another \( C \) divisions in order to calculate \( \tilde{p}/\partial x \). Thus, the Jacobian is calculated exactly and not by numerical differentiation (ND) as is often done in general purpose programs. The reason for this is that, apart from the fact that it is more efficient for iteration convergence to have an accurate Jacobian, it is in fact in most cases more expensive to calculate it by ND rather than exactly for problems with this special structure. If the sparsity of \( \partial p/\partial x \) is not considered, ND requires \( (M+1) \cdot C \) multiplications when a first order difference formula is used. If sparsity is considered this number can be reduced by approximately a factor 3/4, as is reported by Curtis and Chance (1972).

The matrix \( A \) is factorized according to (6) by a generalized LU-decomposition technique before it is stored in a compact way. The aim is to locate a set of rows that span the range of \( A \). If the calculations are done without rounding error the procedure runs as follows:
Gaussian elimination of the columns is used with the elimination procedure going from left to right in the matrix A. If the pivot element in row k becomes zero in the elimination the corresponding column is searched until a nonzero element is found in row l, say. The rows k and l are interchanged by updating a permutation vector and the process can continue. If all elements in a column equal zero, however, it is linearly dependent on the linearly independent columns to the left of it. This column is then excluded from the elimination process, but the linear dependence is marked by a "flag" in the position of the corresponding column in an indication vector. In the upper right half of A we store the factors in a matrix L₁ corresponding to the matrix L in usual LU-decomposition. The above described procedure continues to the last column of A. By this stage the first R entries of the permutation vector are indices of a set of "base-rows" spanning the range of A. Now the remaining M-R rows can be expressed as linear combinations of the "base-rows" by completing the Gaussian elimination (the back-substitution part). Finally, from the factorized form, the first R rows are recreated.

As a final result of the whole procedure we obtain the desired representation (6) in the following positions of the original matrix A:

\[
\begin{bmatrix}
  i_1 \\
  i_2 \\
  \vdots \\
  i_R \\
  \vdots \\
  i_M \\
\end{bmatrix}
\begin{bmatrix}
  A_1 \\
  \vdots \\
  L \\
\end{bmatrix}
\]\n
\[\begin{bmatrix}
R \\
M-R
\end{bmatrix}
\]

Permutation vector P₁

The symbol | represents the lower part (i.e. the last M-R entries) of the columns that are linearly dependent of the linearly independent columns to the left. The matrix L in (6) consists of the last M-R rows in (9) apart from the columns marked |.

It should be pointed out that the representation (9) can be achieved on the same space of memory as the original matrix A, except for one permutation vector of dimension M×1 and one indication vector of dimension 1×N.

Ex. The coefficient matrix of the reaction system

\[A + B \xrightarrow{\neq} C\]

\[2C \rightarrow D + E\] is
so the concentrations of A and C completely describe the system.

The matrix $A_1$ is then stored in a compact way:

$$D = \begin{bmatrix} D_1 & D_2 & \cdots & D_i & \cdots & D_R \end{bmatrix}, \quad D_i = \sum_{j \neq i} r_{ji} q_{ji}$$

$A_1[1,k]$ stores the column numbers of the nonzero elements of $A_1$

$A_1[2,k]$ stores the nonzero elements of $A_1$

From this representation it is easy to compute the right-hand sides of (7), i.e. $A_1 p$ and its Jacobian

$$A_1 \left( \frac{\partial p}{\partial x_1} + \frac{\partial p}{\partial x_2} \cdot L \right).$$

The elements of $A_1 p$ are calculated by the following ALGOL statements:

```algol
s := 0;
for i := 1 step 1 until R do
begin F[i] := 0; for j := 1 step 1 until D[i] do
begin s := s + 1; k := A1[1,s];
end
end;
```

Ex. Robertson’s example ... (1) is in this way of storing information represented as:

$$C = \begin{bmatrix} 1 & 2 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 3 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 2 & 3 & 2 & 2 \end{bmatrix}, \quad A1 = \begin{bmatrix} 1 & 2 & 1 & 2 & 3 \\
-1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

$$L = (-1, -1) \quad P_1 = (1, 2, 3)$$
5. CONCLUSIONS

This paper has given a description of a special purpose program for simulation of chemical kinetics. Efforts have been made to use the special structure of the problem class considered in order to gain efficiency. Examples of "short-cuts" are the reduction of the number of equations by making a preprocessing of the right-hand sides, and the possibility of generating the Jacobian exactly and even more cheaply than numerical differentiation. The work is still under development. There are plans to implement the program on other systems, but the aim of the research is to work out efficient methods for parameter estimation, especially studying the problem of examining how much information about some parameters can be obtained from empirical curves. This aspect is important since different rate expressions may give very close numerical solutions. Some investigations in this field have been done recently by Curtis and Edsberg: Some investigations into data requirements for rate constant estimation. Rep. Theoretical Physics Division, A.E.R.E. Harwell, Didcot, England (1973).

6. ACKNOWLEDGEMENTS

This work has been proposed by professor Germund Dahlquist and is supported by the Swedish Institute for Applied Mathematics.
OPTIMAL SECOND DERIVATIVE METHODS FOR STIFF SYSTEMS

W. H. ENRIGHT
University of Toronto, Scarborough College
Toronto, Ontario

ABSTRACT

In implementing numerical methods for stiff ordinary differential equations several difficulties arise which are not usually encountered with methods designed for non-stiff equations. These difficulties include the choice of the basic formula, the choice of the iteration scheme to solve the implicit set of equations associated with each step, and the choice of a valid error estimate and step control strategy. We will introduce a class of second derivative multistep formulas suitable for stiff equations and show how they have been implemented in an efficient variable-order method.

We will also discuss the special difficulty associated with large systems and discuss how, in this case, one must take advantage of a sparse Jacobian. We will consider two modified versions of our method which are suitable for the solution of large systems. One modification involves the introduction of complex arithmetic and the second involves the introduction of a different class of basic second derivative formulas. Numerical results will be given for the original method and each of the modified versions.

This work was supported in part by the National Research Council of Canada and by USAEC while the author held a visiting appointment at Argonne National Laboratory.
1. INTRODUCTION

Recently several new formulas have been suggested for the numerical solution of stiff equations. Although there seems to have been substantial analysis of the stability and asymptotic accuracy of these formulas, little work has been done in implementing these formulas in complete numerical methods.

In section 2 we will introduce a class of second derivative multistep formulas which have been derived for the solution of stiff equations. We will show, in section 3, how this class of formulas has been implemented in an efficient and reliable numerical method. Several decisions that arise in the implementation of methods for stiff equations will be discussed. These decisions, which include the choice of error estimate and the form of the iteration scheme, can be very critical to both the reliability and efficiency of a method.

In section 4 we will investigate the special difficulty associated with large stiff systems and it will be shown that some modifications are necessary to enable the method to take full advantage of a sparse Jacobian. Numerical results on two typical stiff systems of equations will be presented in section 5.

2. BASIC SECOND DERIVATIVE FORMULAS

In a previous paper (Enright (1973)) second derivative multistep formulas were defined and a class of formulas suitable for stiff equations were developed.

Basically our idea was to use the Jacobian matrix in the multistep formula as well as in the associated iteration scheme. By requiring desirable stability characteristics in a neighbourhood of the origin and in a neighbourhood of \(-\infty\) we were lead to a class of formulas of the form:

\[ y_{i+1} = y_i + h \sum_{r=o}^{k} \beta_r y_{i+1-r} + h^2 \gamma_0 y_{i+1}^{(2)}. \]  

The remaining coefficients, \(\beta_0, \beta_1, \ldots, \beta_k, \gamma_0\) were then chosen to obtain formulas of order \(k+2\). The coefficients for this class of formulas appear in table 1 and the corresponding stability regions appear in figure 1. It can be seen that the formulas are stiffly-stable for \(k\) less than or equal to 7.
## Table 1

Coefficients of the basic second-derivative $k$-step formulas of order $k + 2$. 

<table>
<thead>
<tr>
<th>$k$</th>
<th>order</th>
<th>$\gamma_0$</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\beta_5$</th>
<th>$\beta_6$</th>
<th>$\beta_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>$-\frac{1}{6}$</td>
<td>$\frac{2}{3}$</td>
<td>$\frac{1}{3}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$-\frac{1}{8}$</td>
<td>$\frac{29}{48}$</td>
<td>$\frac{5}{12}$</td>
<td>$-\frac{1}{48}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>$\frac{-19}{180}$</td>
<td>$\frac{307}{540}$</td>
<td>$\frac{19}{40}$</td>
<td>$-\frac{1}{20}$</td>
<td>$\frac{7}{1080}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>$\frac{-3}{32}$</td>
<td>$\frac{3133}{5760}$</td>
<td>$\frac{47}{90}$</td>
<td>$-\frac{41}{480}$</td>
<td>$\frac{1}{45}$</td>
<td>$\frac{-17}{5760}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>$\frac{-863}{10080}$</td>
<td>$\frac{317731}{604800}$</td>
<td>$\frac{2837}{5040}$</td>
<td>$\frac{-1271}{10080}$</td>
<td>$\frac{373}{7560}$</td>
<td>$\frac{-529}{40320}$</td>
<td>$\frac{41}{25200}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>$\frac{-275}{3456}$</td>
<td>$\frac{247021}{483840}$</td>
<td>$\frac{12079}{20160}$</td>
<td>$\frac{-13823}{80640}$</td>
<td>$\frac{8131}{90720}$</td>
<td>$\frac{-5771}{161280}$</td>
<td>$\frac{179}{20160}$</td>
<td>$\frac{-731}{725760}$</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>$\frac{-33953}{453600}$</td>
<td>$\frac{1758023}{3528000}$</td>
<td>$\frac{1147051}{1814400}$</td>
<td>$\frac{-133643}{604800}$</td>
<td>$\frac{157513}{1088640}$</td>
<td>$\frac{-2797}{36288}$</td>
<td>$\frac{86791}{302400}$</td>
<td>$\frac{-35453}{5443200}$</td>
<td>$\frac{8563}{12700800}$</td>
</tr>
</tbody>
</table>
Figure 1. Boundaries of the stability regions for the basic k-step second derivative multi-step formulas. Regions are symmetric about the real axis. For \( k = 8 \) the region is disconnected.

In section 3 an implementation of these formulas in a variable-order, variable-step method will be discussed. A detailed specification of the implementation considered and a FORTRAN listing is available from the author.

It will be shown, in section 4, that when one is solving large systems of equations, a different choice of the coefficients \( \beta_0, \beta_1 \ldots \beta_k, \gamma_0 \) might be more appropriate.

3. IMPLEMENTATION CONSIDERATIONS

Numerical comparisons of methods for non-stiff equations (Hull et al (1972A)) have shown that, to be efficient over a range of error tolerances, a method must be able to vary its order as well as its stepsize. We have implemented the formulas described in the previous section in a variable-order method that starts with the one-step third order formula and uses formulas up to order nine.
Our implementation has been structured in a disciplined manner to facilitate readability and ease of modification. We follow the ideas of Hull et al (1972A) and view a method as consisting of four basic stages:

i) prepare for a step (choose stepsize and order)
ii) calculate an approximation to $y(x_i+h)$
iii) estimate the local error
iv) make decisions, exiting if appropriate.

For a more detailed discussion and justification of this organisation of methods for ordinary differential equations see Hull and Enright*.

In implementing methods for stiff equations there are several decisions and strategies which require special consideration.

The strategy for estimating the local error is a critical component since it, in effect, determines the reliability of the method. Because $h \|\partial f/\partial y\|$ can be quite large, it is not obvious that the usual predictor-corrector type of error estimate can be justified. We have adopted a one-step, two half steps error estimate similar to that used in many Runge-Kutta methods. This requires that we compute two approximations to $y(x_i+h)$, one with stepsize $h$ and another with stepsize $h/2$. This form of local error estimate is justified by the fact that for a large class of linear stiff equations, we can guarantee the reliability of the estimate, even when $h \|\partial f/\partial y\|$ is large (see Enright (1972) Chapter 3)†. It has the disadvantage that we must solve the implicit set of equations 2.1 three times for each attempted step.

The decision whether to accept a step, $(x_{i+1}, y_{i+1})$, is based on a desire to keep the local error per unit step less than the specified error tolerance, $\tau$. That is we accept the step $(x_{i+1}, y_{i+1})$ if the magnitude of the maximum component of the local error estimate is less than $\tau (x_{i+1} - x_i)$.

The stepsize choosing strategy is critical to the efficiency of a method since too small a step can lead to

†Reference is cited at end of paper.
many extra function evaluations, while too large a step will lead to a rejected step which in turn leads to extra matrix inversions (or L-U decompositions) in the iteration scheme. We must also realize that a change in step-size is expensive since it results in a matrix inversion (or L-U decomposition) and therefore an increase in the stepsize should only be attempted when a significant gain can be achieved. Keeping this in mind and considering the error estimate that is used, we have found it convenient to restrict stepsize changes to halving and doubling, and to only double when our estimate of the error per unit step on a doubled step is less than \( r/2 \). Because of this conservative strategy we do not expect rejections, but when one occurs we halve the step and restart at third order. An increase in stepsize is considered only if the step has been constant for \( k \) steps (where \( k \) is the number of steps in the formula being used).

The order-choosing strategy we have adopted is based on a combination of heuristics and numerical experience. Initially a maximum order is chosen based on the specified error tolerance. The method then begins at third order and increases order only when the stepsize has been constant for more than \( k+1 \) steps. This continues until the order is equal to the maximum order.

The remaining component of our method which must be discussed is our choice of iteration scheme to solve the implicit set of equations 2.1. We have ignored \( \partial f/\partial y^2 \) and used a modified Newton-Raphson iteration similar to that suggested by Liniger and Willoughby (1970).

\[
y^{0}_{i+1} = y_i
\]
\[
y^{\ell+1}_{i+1} = y^{\ell}_{i+1} + \Delta^{\ell+1}_{i+1} \quad \text{for } \ell = 1, 2, \ldots, 10;
\]

where

\[
W \Delta^{\ell+1}_{i+1} = -y^{\ell}_{i+1} + h\beta_{\ell} f(y^{\ell}_{i+1}) + h^2\gamma_{\ell} \frac{\partial f}{\partial y}(y^{\ell}_{i+1})
+ y_i + h \sum_{r=1}^{k} \beta_r y^r_{i+1-r}, \quad (3.1)
\]

and

\[
W \approx [I - h\beta_0 \left( \frac{\partial f}{\partial y} \right) - h^2\gamma_0 \left( \frac{\partial f}{\partial y} \right)^2].
\]
W is recalculated when order or stepsize changes or when the iteration scheme is not converging fast enough. A maximum of ten iterations are performed and if there is still no convergence the stepsize is halved and a new step attempted. The iteration scheme is considered to have converged if
\[ \Delta_{i+1}^k < \tau(x_{i+1} - x_i). \]

In the solution of 3.1 we can either retain \( W^{-1} \) or the L-U decomposition of \( W \). For convenience we retain \( W^{-1} \) but for a production code the L-U decomposition should probably be used. Note that because of our error estimate we must retain two matrices (or L-U decompositions).

A comprehensive comparison of methods for stiff equations is being carried out (Enright, Hull and Sedgwick*) and we will not discuss the issues involved or the detailed results here. It should be noted, though, that the method described in this section has proved to be both reliable and efficient for a wide class of stiff equations. Typical numerical results on two stiff systems will be presented in section 5.

4. MODIFICATIONS FOR LARGE SYSTEMS

One result that was apparent from our comparisons is that, as the dimension \( n \) of the system increases, our method spends most of its time computing \( (\partial f/\partial y)^2 \) and \( W^{-1} \), which require \( O(n^3) \) operations, and in calculating \( (\partial f/\partial y) f(y) \) and solving 3.1 which require \( O(n^2) \) operations on each iteration. Note that similar difficulties are present in all methods for stiff equations and remain even if L-U decompositions and back substitutions are used.

Since most stiff equations of large dimension that arise in practice have a sparse Jacobian it is obvious that, to be efficient for large systems, a method must take advantage of sparseness. This is the approach taken by Hachtel et al (1971A) and by Gear (1971C) who have developed preprocessors which will generate efficient code for both the L-U decomposition and the back substitution given a sparse system of equations.

If we consider the implicit set of equations 3.1, it is not clear that $W$ will be sparse, even if $\partial f/\partial y$ is sparse. We will consider two different modifications to our method which will allow us to overcome this difficulty and take advantage of a sparse Jacobian.

The first modification introduces complex arithmetic and was suggested by R. Willoughby (private communication).

If we let $b$ equal the right hand side of 3.1 and factor the left hand side we obtain

$$-\gamma_0 \left( h \frac{\partial f}{\partial y} - ri \right) \left( h \frac{\partial f}{\partial y} - r i \right) \Delta_{i+1}^{\ell+1} = b, \quad (4.1)$$

where

$$r = -\frac{\beta_0}{2\gamma_0} + i \sqrt{\left( \frac{\beta_0}{2\gamma_0} \right)^2 - \frac{1}{\gamma_0}}. \quad (4.2)$$

For the formulas discussed in section 2

$$-\left( \frac{\beta_0}{2\gamma_0} \right)^2 - \frac{1}{\gamma_0} > 0,$$

so $r$ has a non-zero imaginary part.

Now setting

$$u_{i+1}^{\ell+1} = \left( h \frac{\partial f}{\partial y} - ri \right) \Delta_{i+1}^{\ell+1}, \quad (4.3)$$

we can determine $u_{i+1}^{\ell+1}$ by solving the complex system

$$-\gamma_0 \left( h \frac{\partial f}{\partial y} - ri \right) u_{i+1}^{\ell+1} = b, \quad (4.4)$$

From 4.3, since $\frac{\partial f}{\partial y}$ and $\Delta_{i+1}^{\ell+1}$ are both real, we can equate imaginary parts and obtain

$$\Delta_{i+1}^{\ell+1} = \text{im} (u_{i+1}^{\ell+1}) / \text{im}(r) = \text{im}(u_{i+1}^{\ell+1}) / \sqrt{\left( \frac{\beta_0}{2\gamma_0} \right)^2 - \frac{1}{\gamma_0}} \quad (4.5)$$

We see that with this approach we can avoid the calculation of $(\partial f/\partial y)^2$ and also take advantage of a sparse Jacobian in the solution of the complex system of equations 4.4.

The second approach we consider involves the choice of a different set of formulas as a basis for our method.
Consider the multistep formula 2.1. In section 2 we chose the coefficients to obtain a formula of order \( k+2 \). If instead, we set \( \gamma_0 = - (\beta_0/2)^2 \) and choose the remaining coefficients to obtain a formula of order \( k+1 \), the associated iteration scheme 3.1 would reduce to:

\[
(I-h^2 \frac{\partial^2 f}{\partial y^2})^2 \Delta Y_{i+1} = -Y_{i+1}^{2} + h\beta_0 f(Y_{i+1}^{2}) - h^2 \left( \frac{\beta_0}{2} \right)^2 \frac{\partial f}{\partial y}(Y_{i+1}^{2}) + y_{i+1} + h \sum_{r=1}^{k} \beta_r y_{i+1-r}.
\]  

This approach is similar to that taken by Nörsett (1973) who has recently investigated higher derivative, one-step methods for a special class of stiff equations. Note that when we use these formulas we no longer have to square a matrix and we can take advantage of a sparse Jacobian, but we must perform two back substitutions on each iteration.

In determining the coefficients of these formulas we have made the multistep formulas exact for polynomials of degree \( \leq k+1 \) and obtained the following system of equations:

\[
A \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \\ \beta_0 \\ \left( \frac{\beta_0}{2} \right)^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix},
\]

where \( A \) is a \((k+1) \times (k+2)\) matrix.

\[
A_{rj} = \begin{cases} 
1 & \text{for } r = 1, j = 1, 2 \ldots k+1, \\
r(1-j)^{r-1} & \text{for } r = 2, 3, k+1, j = 1, 2 \ldots k, \\
r & \text{for } r = 2, 3 - k+1, j = k+1, \\
-r(r-1) & \text{for } r = 1, 2, -k+1, j = k+2.
\end{cases}
\]

We solved this system of equations by applying Gaussian elimination to reduce the \((k+1)\)st equation to a standard quadratic equation; solving the quadratic equa-
tion to determine \( \beta_0 \); and using back substitution to solve for the remaining coefficients. We have solved these systems for \( k \leq 8 \) and found that in all cases the coefficients are real and, for each value of \( k \), the two solutions to the quadratic result in one formula with \( 0.4 \leq \beta_0 \leq 0.6 \) and another with \( 1.0 \leq \beta_0 \leq 3.4 \). The formulas with the smaller value of \( \beta_0 \) have smaller truncation error coefficients but less desirable stability regions. In fact for \( k \geq 5 \) these formulas are no longer stiffly stable since the stability region is disconnected, similar to the tenth order formula in figure 1. On the other hand, the formulas with the larger values of \( \beta_0 \) have been found to be stiffly stable for \( k \leq 7 \). Considering these factors and various numerical experiments we have chosen the formulas with the smaller value of \( \beta_0 \) for \( k \leq 3 \) and those with the larger value of \( \beta_0 \) for \( 4 \leq k \leq 7 \). These formulas have been implemented in a numerical method similar to the one described in section 3. The coefficients of the formulas used appear in table 2 and the corresponding stability regions appear in figure 2.

We have discussed two modifications to our method which would enable it to take advantage of a sparse Jacobian. It is important to note that these modifications would only be necessary for systems of dimensions larger than about 20. For these systems the Jacobian should never be computed explicitly, but instead, efficient subroutines should be provided for the L-U decomposition and back substitutions associated with the iteration scheme and the product of the Jacobian matrix and an arbitrary vector. Ideally these subroutines would be generated automatically, using symbol manipulation, from the Fortran code for \( f(y) \).

5. NUMERICAL RESULTS AND DISCUSSION

In the previous sections we have discussed an implementation of a method for stiff equations and two modifications to the basic method which make it suitable for large systems. The basic method, described in section 2, and available from the author in FORTRAN, will be referred to as SDBASIC. The two modified versions will be referred to as SDCMPLX for the case which involves complex arithmetic, and SDRROOT for the case which uses a different class of multistep formulas. Although these modified versions of the basic method are capable of exploiting sparseness, we have not included this technique in our program. The assessment of how much this technique
<table>
<thead>
<tr>
<th>k</th>
<th>order</th>
<th>$Z$</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\beta_5$</th>
<th>$\beta_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>$\sqrt{2}$</td>
<td>2</td>
<td>-1</td>
<td>$-Z$</td>
<td>+ $Z$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>$\sqrt{6}$</td>
<td>$\frac{4}{3}$</td>
<td>$\frac{2}{9}$</td>
<td>$-\frac{5}{9}$</td>
<td>+ $\frac{4}{9}Z$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>$\sqrt{5}$</td>
<td>$\frac{12}{11}$</td>
<td>$\frac{-257}{2904}$</td>
<td>$\frac{137}{363}$</td>
<td>$\frac{-1103}{2904}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>$\sqrt{\frac{1419}{4}}$</td>
<td>$\frac{24}{25}$</td>
<td>$\frac{1057}{22500}$</td>
<td>$\frac{-3661}{15000}$</td>
<td>$\frac{3853}{7500}$</td>
<td>$\frac{-12449}{45000}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>$\sqrt{\frac{5118}{2}}$</td>
<td>$\frac{120}{137}$</td>
<td>$\frac{-261979}{9009120}$</td>
<td>$\frac{2416169}{13513680}$</td>
<td>$\frac{-2083057}{4504560}$</td>
<td>$\frac{2889973}{27027360}$</td>
<td>$\frac{-5534137}{768320}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>$\sqrt{\frac{117573}{4}}$</td>
<td>$\frac{40}{49}$</td>
<td>$\frac{1231883}{62233920}$</td>
<td>$\frac{-20297}{14406}$</td>
<td>$\frac{124541}{288120}$</td>
<td>$\frac{-2887799}{3889620}$</td>
<td>$\frac{587501}{768320}$</td>
<td>$\frac{-10783}{72030}$</td>
</tr>
</tbody>
</table>

Table 2. Coefficients of the modified second derivative formulas of order $k+1$. 
can help and where the trade-offs lie will be investigated in the future. Because of this, we only include numerical results on two typical small stiff systems. These results will indicate what, if anything, we have to sacrifice in the solution of small systems to make the method efficient for large systems.

We also include the results of Gear's variable-order multistep method (Gear (1971A)), which is probably the most widely used method for stiff equations. We have had to modify Gear's program to conform to an error per unit step error control. (See Hull et al (1972A) for a discussion of the necessity and implications of this change). The results appear in table 3.
Problem 1: Linear problem from circuit theory (Klopfenstein and Davis (1970A)).

\[
\begin{bmatrix}
-1800 & 900 & 0 & . & . & 0 \\
1 & -2 & 1 & . & . & 0 \\
0 & 1 & -2 & 1 & . & . \\
\end{bmatrix}
\begin{bmatrix}
y' \\
y \\
0 \\
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}
\]

initial conditions

\[
y(0) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{range} = [0, 120], \ n=9 
\]

error tolerance = $10^{-6}$.

Problem 2. Non-linear problem from control systems (Davison (private communication)).

\[
y'_1 = y_2 \\
y'_2 = y_3 \\
y'_3 = y_4 \\
y'_4 = (y_1^2 - \sin(y_1) - \Gamma^4)y_1 + (y_2y_3(y_2 + 1) - 4\Gamma^3) \\
+ (1-6\Gamma^2)y_3 + (10e^{-y_2^2} - 4\Gamma)y_4 + 1 
\]

where $\Gamma = 100$ (gain of system)

range = $[0, 1]$

\[
y(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad h_0 = 6.8 \times 10^{-3} 
\]

error tolerance $\tau = 10^{-6}$.

We have tested these methods on several other stiff systems and the results are consistent with those reported here. In general the three second derivative methods require substantially fewer function evaluations than Gear's method but the number of Jacobian evaluations is, of course, much greater. The method SDBASIC and GEAR are
<table>
<thead>
<tr>
<th>Problem Method</th>
<th>Function calls</th>
<th>Jacobian calls</th>
<th>Matrix inversions</th>
<th>Total time (sec.)</th>
<th>Maximum error/step</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD BASIC 131</td>
<td>409</td>
<td>405</td>
<td>34</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>SD COMPLEX 131</td>
<td>176</td>
<td>594</td>
<td>41</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>SD Root 123</td>
<td>123</td>
<td>26</td>
<td>3.6</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>GEAR 15</td>
<td>15</td>
<td>1.1 x 10^-6</td>
<td>1.1 x 10^-6</td>
<td>1.1 x 10^-6</td>
<td>1.1 x 10^-6</td>
</tr>
</tbody>
</table>

Table 3: Numerical results on two stiff systems. (The time is in seconds on the IBM 370/165.)
comparable in terms of total time required to solve the
problem, whereas the two modified versions, SDCMPLX and
SDRROOT, require approximately twice as much time to
solve the problem.

As would be expected the statistics for SDCMPLX are
the same as for SDBASIC except for the total time which
reflects the cost of the complex arithmetic. The method
SDRROOT requires about 50% more function evaluations and
Jacobian evaluations in the examples included in table 3.
In general, over the much wider class of problems tested,
this figure varied from 10% - 50%.

These results indicate that SDBASIC is an efficient
and reliable method that compares quite well with GEAR.
In addition the two modified versions designed for large
systems are quite reliable and reasonably efficient,
even for small systems where no attempt is made to take
advantage of sparseness. Of course these methods would
become more efficient (in terms of total time) than
SDBASIC in the solution of large systems.

Acknowledgment

The author is grateful to T.E. Hull and A.E.
Sedgwick for the many discussions that have influenced
this work, especially the structuring and organization
of SDBASIC.

Reference

1972

Enright, W. H., "Studies in the numerical solution of stiff
of Toronto, Computer Science Report 42.
THE STABILITY OF AUTOMATIC PROGRAMS FOR NUMERICAL PROBLEMS

C. W. Gear*, K. W. Tu†, D. S. Watanabe*

*Dept. of Computer Science, University of Illinois
†Dept. of Computer Science, Rutgers University

ABSTRACT

Many classes of methods for numerical problems can be characterized by parameters. For example, Adams methods for integrating ordinary differential equations can be characterized by the step size and order. In many cases it is known that a method is stable for fixed values of those parameters, whereas many programs vary those parameters in order to reduce computation time. This paper examines the effects on stability of changes to two kinds of parameters—"discrete" and "continuous"—corresponding to the parameters of order and step size in the numerical solution of ordinary differential equations. Conditions are given that are sufficient to ensure that these changes do not affect stability. These results are applied to two classes of automatic methods for ordinary differential equations—one based on unequal interval quadrature formulas, the other based on interpolation and equal interval quadrature formulas.

INTRODUCTION

Many numerical methods consist of a calculation of a sequence of values \( \mathbf{x}_n \) in the following form:

\[
\begin{align*}
\mathbf{x}_{n+1} &= P(\mathbf{x}_n, \mathbf{y}_n), \\
\mathbf{y}_{n+1} &= V(\mathbf{x}_n, \mathbf{y}_n),
\end{align*}
\]

where \( \mathbf{x}_0 \) and \( \mathbf{y}_0 \) are given. The variables in the vector \( \mathbf{x} \) are assumed to be approximations to the desired data values, while the
variables in the vector \( v \) are assumed to be parameters that determine the method. The computation in \( P \) is usually called the numerical technique, while the computation in \( V \) is the scheme for selecting parameters. The variables are split between these two vectors because we are interested in examining the stability of the computation of \( x \), but typically we do not care about the computed values of \( v \). An example might clarify this. Suppose we wish to solve \( y' = f(y, t) \) with \( y(0) \) given. The collection of variables in \( x \) will be all the saved information concerning the solution at \( t=v \), while the variables \( v \) will include the step size \( h \), the current time \( t \), and parameters which determine the method to be used for the next step.

We assume that the solution of the underlying problem is \( x(v) \). That is, the numerical method is approximating \( x(v) \) by \( x_n \). The local error \( \epsilon_n \) is defined by

\[
\epsilon_n = P(x(v_n), v_n) - x(v_{n+1}).
\]

This is "the amount by which the solution fails to satisfy the method." \( \epsilon_n \) can include round-off errors in the evaluation of \( P \). For example, if the problem is to solve a system of equations and (1) represents some consistent iterative scheme, then the solution \( x(v) \) is independent of \( v \), and \( \epsilon_n \) will be solely the round-off error in the computation of \( P \).

The global error \( \epsilon_n \) is defined by

\[
\epsilon_n = x_n - x(v_n).
\]

Eqs. (1), (2), and (3) immediately give

\[
\epsilon_{n+1} = T_n \epsilon_n + \epsilon_n - \epsilon_{n+1},
\]

where \( T = \partial P/\partial x \) with elements evaluated at \( v = v_n \) and \( x \) somewhere between \( x_n \) and \( x(v) \). (We assume that \( P \) is continuously differentiable with respect to \( x \) in a suitable region.) We use the notation

\[
T_n^m = \begin{cases} \begin{bmatrix} I \\ T_{m-1} & T_{m-2} & \cdots & T_n \end{bmatrix}, & m = n; \\ T_n, & m > n. \end{cases}
\]

Then, eq. (4) implies

\[
\epsilon_N = \sum_{n=0}^{N-1} T_n^N \epsilon_n + T_N^N \epsilon_0.
\]

By stability we mean that \( ||T_n^m|| \) is uniformly bounded by some constant \( K \) for all \( m \leq n \). This definition is most appropriate for
methods for differential equations on a finite interval. (It may be desirable to change it for other problems. For example, we could define a method as stable for solving sets of nonlinear equations if there exists \( K > 0 \) and \( \lambda > 0 \) such that \( ||T_m|| \leq K \lambda^m \) for all \( m \geq n \).) If \( x_n \) is the numerical approximation to a solution of a set of ordinary differential equations at \( t_n \), and if \( h_n = t_{n+1} - t_n \), we can see from eq. (5) that if \( ||a_n|| = o(h_n) \) and the method is stable, then \( ||e_n|| \to 0 \) for all \( t_n \in [0, T] \) as \( \max(h_n) \to 0 \).

We wish to determine what happens to a given set of methods \( P = P(\cdot, v) \), which are stable for fixed \( v \), when \( v \) is varied during the computation. However, we do not want to specify the behavior of \( v \) too precisely because programs vary \( v \) on the basis of computationally estimated factors. Moreover, these factors may be applied to the algorithm on a heuristic basis. Consequently, we wish to discover what restrictions on an otherwise free choice are sufficient to maintain stability.

ORDER AND STEP VARIATION IN ORDINARY DIFFERENTIAL EQUATIONS

The stability of step variation techniques was analyzed in Tu.* A summary can be found in Gear and Tu (1973A). These results were extended to order variation techniques in Gear and Watanabe.† The form of the computation \( P \) in eq. (1) is analyzed in detail in those reports. We will summarize the results here.

For each component \( y \) to be integrated, a subvector of information \( y_n = [y_n, y_{n-1}, \ldots, y_{n-k+1}, h_n, y_n', \ldots, h_n y_{n-k}' \] is carried at \( t = t_n \) is the collection of all \( y_n \). (Linear transformations may be applied to \( y_n \) to get and store \( Qy_n \), but these are unimportant to a discussion of stability.) One step of integration takes the form

\[
y_{n+1} = S_n C_n y_n + h_n \phi_n (C_n y_n),
\]

where \( Q_n \) represents those operations necessary to change order, \( C_n \) represents those operations necessary to change step size (see below), \( S_n \) represents that part of the computation which is

---


independent of the differential equation, and \( \phi \) represents the contribution from the differential equation. (The "one-step like" notation is due to Skeel*, who has extended Tu's result on the stability and convergence of one form of step variation.) Useful integration methods have the property that \( \phi_n \) is uniformly Lipschitz in its argument.

The precise form of \( S \) depends on the method used. We will consider multistep methods of the form

\[
\sum_{i=0}^{k} \left[ a_{i,n} y_{n-i} + \beta_{i,n} h^{n-l-i} y'_{n-l} \right] = 0. \tag{7}
\]

The form of \( S \) is independent of how this is solved (by predictor-corrector or Newton iteration, for example) and is, assuming \( a_{0,n} = -1 \),

\[
S_n = \begin{bmatrix}
    a_{1,n} & \ldots & a_{k-1,n} & a_{k,n} & \beta_{1,n} & \ldots & \beta_{k-1,n} & \beta_{k,n} \\
    1 & \ldots & 1 & 0 & 0 & \ldots & 0 & 0 \\
    0 & \ldots & 0 & 0 & 1 & \ldots & 1 & 0
\end{bmatrix} . \tag{8}
\]

Two different techniques for changing step size and order are in use. We call these the variable step technique and the interpolation technique. They are described below.

**Variable Step Technique**

The coefficients in eq. (7) are functions of \( h, \ldots, h^{n-k} \) chosen so that the equation is exact when \( y \) is any polynomial of degree \( p \). Since there are \( 2k+1 \) coefficients in eq. (7) and

p \leq k+1 for a strongly stable method, at least k-1 additional constraints must be imposed to determine the coefficients. We will assume that these constraints are such that the coefficients are analytic functions of the \( h \), in a suitable region. (See Tu, op. cit, for details.) Typically, these constraints require that many of the coefficients be zero. For example, if \( \alpha_{q,n} = 0 \), \( q \geq 2 \) and \( \beta_{q,n} = 0 \), \( q \geq p \), we have the \( p \)-th order Adams-Moulton method. If \( \beta_{q,n} = 0 \), \( q \geq 1 \) and \( \alpha_{q,n} = 0 \), \( q \geq p+1 \), we have the \( p \)-th order backward differentiation formula (BDF).

The operators \( C_n \) and \( O_n \) are the identity operators in this technique.

**Interpolation Technique**

The coefficients in eq. (7) are values, independent of \( h \), chosen so that the equation is exact when \( y \) is any polynomial of degree \( p \) and the \( h \)'s are equal. (The coefficients may still depend on \( n \) since the order may change from step to step, or we may switch from, say, Adams formula to the BDF.) Changing step size is accomplished by interpolating through known values \( \{y_{n-1}\} \) and \( \{h y'_{n-1}\} \) to get values based on a different step size. The matrix \( C_n \) accomplishes this. The matrix \( O_n \) could be the identity if a sufficient number of past values were saved to allow the order to be increased. Usually, however, only enough variables are saved to work at the current order, and the matrix \( O_n \) represents an interpolation to obtain additional values when the order is increased.

**Comparison of the Two Techniques**

The variable step technique requires that the coefficients \( \alpha \) and \( \beta \) be recomputed frequently. This is an expensive process, but it is independent of the number of equations. Consequently, this technique is attractive for large systems of equations. The interpolation technique is usually used in a Nordsieck type method, where the saved information represents approximations to \( y_n \), \( h y' \), \( h^2 y''/2 \), ..., \( h^k y^{(k)}/k! \). Since these scaled derivatives are "easy" to "interpolate" when the step size changes, this method is attractive for small systems of equations.
TWO SIMPLE LEMMAS

Lemma 1

If \( T = T_1 + T_2 \) and there exist real functions \( v(t) > 0 \) and \( u(t) \), and a set of real numbers \( 1 \leq k \) and \( t_0 < t_1 < t_2 \ldots \) such that

(i) \( ||T_n^m|| = ||T_{n-1}^m + T_{n-2}^m \ldots + T_1^m|| \leq K \exp\left[ \int_{t_n}^{t_{n+1}} v(t) dt \right] \),

(ii) \( ||T_n^m|| \leq \frac{t_{n+1} - t_n}{t_{n+1}} v(t) dt \exp\left[ \int_{t_n}^{t_{n+1}} u(t) dt \right] \),

then \( ||T_n^m|| \leq K \exp\left[ \int_{t_n}^{t_{n+1}} (u(t) + K v(t)) dt \right] \).

Proof. It is true for \( m=n \). Assume it is true for \( T_{n-1}^m \) with \( m \geq j > n \). Then

\[
||T_n^m|| \leq ||T_{n-1}^m|| + \sum_{j=n+1}^{m} ||T_j^m|| \leq ||T_{n-1}^m|| \leq K \exp\left[ KL(t_{m} - t_{n}) \right].
\]

Note that

\[
K||T_{j-1}^m|| \leq \exp\left[ \int_{t_j}^{t_{j-1}} K v(t) dt \right] \exp\left[ \int_{t_j}^{t_{j-1}} u(t) dt \right].
\]

On substituting this and the bounds for \( ||T_j^m|| \) and \( ||T_{j-1}^m|| \) into eq. (10) we get the desired result.

Applications of Lemma 1 to Ordinary Differential Equations

Stability was defined earlier as a bound on \( ||T_n^m|| \). This depends on the differential equation. If \( ||\hat{T}_n^m|| \) satisfies a Lipschitz condition in \( y \) with constant \( L \), we can apply Lemma 1 with \( \hat{T}_n^m = S_n C_n \), \( v(t) = L \), and \( u(t) = 0 \) to see that

\[
||\hat{T}_n^m|| \leq K \exp\left[ \int_{t_n}^{t_{n+1}} K v(t) dt \right] \leq K \exp\left( KL(t_{m} - t_{n}) \right).
\]

Hence, we call the condition \( ||\hat{T}_n^m|| \leq K \) a stability criterion. This criterion is independent of the differential equation. Unfortunately, this leaves a lot unsaid. Verifying whether \( ||\hat{T}_n^m|| \leq K \) is a difficult task, and the validity of this assumption may depend on the variations in step sizes and order. For this reason, we must define the stability of variable order and step methods with respect to a step selection scheme and a formula selection scheme. A step selection scheme is a function \( \Theta(t, h) \) in \([\Delta, 1]\) where \( \Delta > 0 \) such that the step \( h \) is given by \( h \Theta(t_n, h) \). A formula selection scheme is an integer function \( I(t_n, h) \) such that the \( I \)-th member
of a set of formulas like eq. (7) is used from \( t \) to \( t + 1 \). A
method, which is defined as a set of formulas and a technique for
changing step size and order, is said to be stable with respect
to the step and order selection schemes 0 and 1 if there exist
\( K \) and \( h_0 > 0 \) such that \( \left| \hat{\Delta} \right| < K \) for all \( 0 < \Delta \leq T \) whenever
\( 0 < \Delta < h_0 \). Results will be summarized in the next section.

On the infinite interval, we are interested in stable
differential equations, particularly stiff ones. In this case,
we hope to be able to use Lemma 1 with negative \( \mu(t) \) to bound the
global error \( \epsilon \). This will require that some stable part of the
differential equation be included in \( \hat{T} \). The approach looks
promising, but we have not yet been successful.

Lemma 2
Let \( \{S_i\} \) and \( \{O_i\} \) be sets of \( q \times q \) matrices with the following
properties:

(i) If \( \{\lambda_{i,j}\}, j = 1, \ldots, q \), are the eigenvalues of \( S_i \), then there
exist constants \( \lambda, \nu, \) and \( n \) such that
\[
\lambda = |\lambda_{i,1}| = |\lambda_{i,2}| = \ldots = |\lambda_{i,v}|
\]
\[
\lambda > n > |\lambda_{i,v+1}|, \ldots, |\lambda_{i,q}| \text{ for all } i.
\]

(ii) There exists a constant linearly independent set of vectors
\( \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_v \) of norm \( 1 \) such that
\[
S_i \tilde{x}_j = \lambda_{i,j} \tilde{x}_j, \quad j = 1, \ldots, v \quad \text{and for all } i.
\]

(iii) If \( Q_i \) is such that its columns have a norm of one and
\( Q_i^{-1} S_i Q_i = J_i \) is the Jordan canonical form of \( S_i \) (without
loss of generality, we can assume that the first \( v \) columns
of \( Q_i \) are \( \tilde{x}_1, \ldots, \tilde{x}_v \)), then there exist constants \( C_1, C_2 \)
such that
\[
\|Q_i^{-1}\| \leq C_1, \quad \|Q_i\| \leq C_2 \quad \text{for all } i.
\]

(iv) \( O_i \tilde{x}_j = \tilde{x}_j, \quad j = 1, \ldots, v \) and there exists a constant \( C_0 \)
such that \( \|O_i\| \leq C_0 \) for all \( i \).

Then for any constant \( K > C_0 C_1 C_2 \) there exists a set of integers
\( \{p_i\} \) such that
\[
\| \prod_{j=1}^N (S_i)_{j,j} \prod_{j=1}^K p_{i,j} \| \leq 100\|
\]
whenever \( p_{i,j} \geq p_{i,j} \).
The proof is a trivial modification of the proof of Lemma 2.1 in Gear and Watanabe, *op. cit.* The lemma can be applied to situations in which a fixed process (e.g., constant order integration scheme) is applied for a number of steps, then a change is made to another fixed process, and so on. If the different processes have error amplifications of $S_i$ (called $T_i$ earlier) and a special operation $O_i$ is needed to change to the method based on $S_i$, the hypotheses of the lemma are sufficient to guarantee the existence of an integer for each process such that stability is not lost as long as the process based on $S_i$ is applied at least $p_i$ times.

Condition (i) simply says that the error amplification is bounded by $\lambda$ in the spectral norm. For ordinary differential equations on the finite interval, $\lambda$ is one. In that case, condition (ii) requires that the undamped components of the numerical solution be the same for all methods. One of these components must correspond to the solution of the differential equation. This is the eigenvector for the eigenvalue $\lambda_{i1} = 1$ and is independent of the method. Other eigenvalues on the unit circle arise only for weakly stable methods. Therefore, sets of strongly stable methods automatically satisfy (ii). In general, weakly stable methods will not satisfy the hypotheses of the lemma. (See Gear and Watanabe, *op. cit.*, for examples.) If the set of processes is finite, condition (iii) is trivially satisfied. The first part of condition (iv) is also automatically satisfied for strongly stable methods in ordinary differential equations since it requires the method changing operator should not destroy the solution of the differential equation. The second part of (iv) is trivially satisfied for a finite set of methods.

The lemma can also be applied to a set of linearly convergent iterative processes. If the rate of convergence of each process is at least $\eta$ (that is, $\rho(S_i) \leq \eta < 1$ for all $i$), then we can choose any $1 > \lambda > \eta$ and $\nu=0$, and the lemma guarantees that for a set of processes there exist integers $p_i$ such that the convergence rate of a combined process will be at least $\lambda$. If the set of processes is infinite, we must also satisfy conditions (iii) and (iv). Although this lemma gives a slower rate of convergence for the composite algorithm, it is useful when a heuristic method believed to yield a greatly improved rate of convergence is used to switch between iterative processes. This lemma gives conditions which, if satisfied, guarantee that the heuristic does not destroy convergence.

**APPLICATION TO ORDINARY DIFFERENTIAL EQUATIONS**

In this section we will summarize the results that were obtained for variable step variable order methods in the previously
cited references by the authors. These results indicate that the
variable step technique is more likely to lead to a stable process
than the interpolation technique, at least in current implementa-
tions. (The variable step technique is completely specified, but
the interpolation technique depends on the number of past values
used in the interpolation. Current implementations use only those
values required if the step size is not changed, but a higher
order of interpolation accuracy would be obtained if more values
were used. This may well affect stability.)

The strongest results are for Adams methods. The first result
is a generalization of that of Piowtrowski (1969A). A constant \( p \)
method is one in which the \( a \) coefficients are independent of \( n \).
(Adams method is a particular case of this.) If the variable step
technique is used, these methods are about as stable as we could
want; specifically:

**Theorem 1**

If the coefficients \( a_i \) in (8) are independent of \( n \) and if
the \( \beta_{i,n} \) are uniformly bounded, then

\[
\| S_{m,n}^{m-1} \| = \prod_{j=m}^{m-1} S_{j}^{n}
\]

is uniformly bounded iff the polynomial

\[
\rho(\xi) = -\xi^k + a_1 \xi^{k-1} + \ldots + a_k \xi^0
\]

has roots which are inside the unit circle or simple on the
unit circle.

The proof is simple because the only potentially unstable
submatrix of \( S_{m,n} \) depends on powers of the submatrix occupying
the top left hand quadrant of \( S_{n} \). However, the characteristic poly-
nomial of this submatrix is \( \rho(\xi) \). The uniform upper bound on
\( \beta_{i,n} \) is met if the \( \beta_{i,n} \) are chosen so that the order is maximized
subject to \( \beta_{i,n} = 0, s_{n} > a_{i,n} \), and the step selection scheme \( e \) is
such that \( e \) is uniformly bounded away from zero. There are no
restrictions on the order selection scheme.

If the interpolation scheme is used, it can be shown that
Adams method is stable if, after a change to a \( k \)-th order method,
the step size and order are maintained for at least \( k \) steps when
the \( O_N \) are the identity matrix, and at least \( k+1 \) steps when the
\( O_N \) represent interpolations. Skeel, op. cit, has sharpened this
result with respect to step changes, and shown that only \( k-l \)
constant steps are needed after a change of step size. This improvement also carries over to order changes. Tu points out that the fourth order Adams method with interpolation is unstable if \( h_{2n} = 10h_{n+1} = h_{2n+2} \) so that for this case, the variable step technique does not cause instability but the interpolation technique does.

Lemma 1 can be applied to the general method in either the interpolation or the variable step form to prove the following result:

**Theorem 2**

If \( \theta(t,h) \) is such that

\[
\frac{h_{n+1}}{h_n} = 1 + O(h)
\]

then a multistep method is stable with respect to \( \theta \) and a constant \( I \).

The proof is based on the fact that \( \hat{T} \) can be expressed as \( \hat{T} + h^n \hat{T} \) where \( \hat{T} \) is the error amplification matrix for a constant step method. Skeel in his thesis, op. cit, has improved this result for the interpolation technique to show that if \( \theta(t,h) \) is a function of bounded variation in \( t \) over \( [0,T] \), then a multistep method is stable. This was done by deriving an error bound that depended on \( V_n \) where

\[
V_n = \frac{1}{h} \sum_{n=0}^{N-1} |h_n - h_{n+1}|
\]

This can be shown for the variable step technique also provided that \( \theta \) is such that the \( a_{i,n} \) and \( b_{i,n} \) in eq. (7) are uniformly bounded for all \( h \) and \( n \).

Tu gives the following example in his thesis, op. cit. \( y' = -y, y(0) = 1 \) is integrated using step sizes \( h_{2n} = .05 \) and \( h_{2n+1} = .005 \) in a three step BDF method. The method is apparently stable if the variable step technique is used, and apparently unstable if the interpolation technique is used. Hence, consideration of both BDF and Adams methods suggests the superiority of the variable step approach. We have attempted to find examples for which the variable step form of BDF is clearly unstable but have not been successful. However, it seems likely that we can get instability, so we do need to consider what restrictions should be placed on step variation.
(Parenthetically, let us note that we could choose a constant \( p \) form of (7) in which the \( a \)'s correspond to the \( a \)'s for a constant step BDF method and the \( b \)'s are chosen to maintain the order when the step changes. Then Theorem 1 guarantees stability with respect to any bounded step changes. However, this technique will require saving more information from step to step.)

Mr. M. Ostrar of the University of Illinois has computed the \( p \) of Lemma 2 for the BDF methods using the constructive proof given in Gear and Watanabe, op. cit. If the order is only increased or decreased by one at a time, \( p \) steps without order change are sufficient after a change to \( p \)-th order to guarantee stability.
AN IMPLEMENTATION OF GEAR'S ALGORITHM FOR CSMP III

A.R. Gourlay* and H.D.D. Watson+

*Department of Mathematics, Loughborough University of Technology, Leicestershire
+Advanced Applications Group, IBM UK Scientific Centre, Peterlee, Co. Durham

INTRODUCTION

In this paper we discuss an implementation of Gear's integration algorithm for stiff systems for use with the IBM 'Continuous Systems Modeling Program', referred to hereafter as CSMP III. We will have little to say from the algorithmic point of view - rather we are concerned with the problems of implementation which arise when one takes a relatively sophisticated algorithm such as that of Gear and attempts to interface it with a complex, powerful and general black-box system such as CSMP III. In particular our comments will be concerned with an efficient implementation of the corrector step using sparse matrix techniques, and our attempts to ensure that the algorithm is as robust and fail-safe as possible. We will also report on our experience of using our implementation for solving CSMP III models of stiff systems.

CSMP III

A definition of simulation could be "the act of representing some aspects of the real world by numbers or symbols which can be easily manipulated to facilitate their study" (McLeod, J.; 'Simulation: The dynamic modelling of ideas and systems with computers'; McGraw - Hill).

Simulation, as a topic, can if desired be broken down into two not necessarily independent, sub-topics usually known as discrete simulation and continuous simulation. Here we essentially consider only the latter subject area where the model to be simulated is represented as a complicated system of ordinary differential equations. We may consider then the role of the digital computer as solving this set of differential equations over some time interval by a discrete approximation in time.
The uses of simulation have varied widely from relatively simple applications to the modelling of complex systems involving perhaps electronic, thermodynamic, chemical and mechanical interactions. These situations were traditionally simulated on analogue computing devices, but with the development of the digital machine towards its present advanced stage the advantages which it offers (flexibility, accuracy, ease of development), coupled with an ever improving price/performance factor, have become increasingly obvious.

As the importance of continuous simulation became more widely recognised, special software in the form of high-level continuous system simulation languages (CSSL) began to appear and achieve wide acceptance as an important aid to the engineer in his studies. Perhaps the most significant CSSL was DSL/90 (1965) from which some of the more modern CSSL's have evolved. IBM's most recent offering in this field is CSMP III and it is with particular reference to this language that we will discuss the implementation of a stiff-system integrator. However a few comments on the philosophy of these languages are necessary for a better appreciation of the particular constraints which arise in attempting to interface a sophisticated integration routine with a black-box type of program package such as CSMP III.

CSMP III is a compiler oriented language in the sense that source statements are processed by a translator or pre-compiler into FORTRAN, which is then compiled and executed in conjunction with pre-compiled CSMP III routines. In principle all the facilities of FORTRAN are available in CSMP III- arithmetic statements provide a powerful arithmetic capability- standard FORTRAN functions (SIN, COS, EXP, AMAX1 etc) and functions commonly used in simulation (DELAY, RAMP, LIMIT, for example) are available in a library to which the user may add his own special purpose routines; the logical and input/output capabilities of FORTRAN are at his disposal. The translator can add to the flexibility and power of FORTRAN; for example free format coding is implemented, and a macro facility is provided. More significant is the fact that the translator will do the statement sorting; in execution, centralized integration is employed, with the user being given the choice of a number of methods. In CSMP III all the integration methods are explicit, and standard (almost classical) with the exception of the new method STIFF designed for the solution of stiff systems. This routine employs the explicit Fowler-Warten method. Full details of this method may be obtained in [Fowler and Warten (1967A)]. Values of the dependent variables at points required for output are determined by imposing the requirement that all output points coincide with integration points. This restriction is not acceptable in the development of a sophisticated stiff system routine. Indeed, insufficient attention is often paid to the question of the determination of values at output points when designing a general purpose differential equation solver.
One of the major improvements in the available facilities in CSMP III was the provision of interactive graphic simulation. However in the area of stiff system modelling it was evident that to allow interactive simulation, a considerable improvement would be required in the speed of execution of the program and that this was tantamount to providing a very fast stiff system integrator. It was therefore decided to implement a version of the well-known variable order-variable step Adams method, usually known as Gear's method, for users of CSMP III. This paper reports on the practical difficulties encountered in this project, and on some of the experience gained in the course of the work.

IMPLEMENTATION OF GEAR'S ALGORITHM FOR CSMP III

In choosing a stiff routine for testing in conjunction with CSMP III we must meet several conditions:

(i) The routine should have been documented, tested and accepted by numerical analysts.

(ii) It should be applicable to both small and large systems - the number of (equivalent) first order equations, \( n \), may be several hundred.

(iii) Its linkage to a black-box package such as CSMP III should require as little modification as possible to existing CSMP III routines.

One of the more satisfactory approaches for tackling stiff problems is that originally due to [Curtis and Hirschfelder (1952A)], which has become particularly associated with the name of [Gear (1968A and 1971A)], who implemented a variable order - variable step version. Whilst the particular details of the scheme do not concern us it is relevant to briefly describe the various stages of the algorithm in order to indicate the areas where interface problems exist. We suppose the system to be defined by

\[
y_i' = f_i(y_1, \ldots, y_n, t) \quad i = 1, \ldots, n
\]

with \( n \) the order of the system. In maintaining a history of the solution it is convenient for the purpose of interpolating values at output points to work with the Nordsieck vector representation. In other words at each point we compute for each \( y_i \) a vector with components \( Z_{ik} \) where for a method of order \( q \)

\[
Z_{ik} = h^{k-1} y_i^{(k-1)} / (k-1)! \quad (k = 1, \ldots, (q + 1))
\]

and \( h \) is the current stepsize. The Gear algorithm is a predictor corrector process with the corrector being solved by a Newton-type method. The predicted value for time \( t, Z_{ik}(0) \) is obtained from the Nordsieck vector at time \( t' \) (\( Z'_{ik} \)) where \( t = t' + h \), by

\[
Z_{ik}(0) = \sum_{p=1}^{q+1} Q_{kp} Z'_{ip} \quad (i = 1, \ldots, n)
\]
where Q is the Pascal triangle matrix given by

\[ Q_{i,j} = \begin{cases} \binom{j-1}{i-1} & i \leq j \\ 0 & i > j \end{cases} \]

The solution of the corrector equation, which is accomplished by a Newton-type method, involves the solution of a set of linear algebraic equations to give

\[ Z_{ik}(p + 1) = Z_{ik}(p) + \lambda_i d_k \]

where d is determined by the present order of the method and the \( \lambda \)'s are the solution of the system:

\[ \sum_{j=1}^{n} \left( 1 - h d \frac{\partial f_i}{\partial y_j} \right) \lambda_j = h f_i (Z_{i1}(p), Z_{i2}(p), \ldots, t) - Z_{i2}(p) \quad (i = 1, \ldots, n) \]

The corrector equation is iterated until satisfactory convergence is achieved. In the CSMP III implementation it was in fact more economical and convenient to divide through the above equation by \( h d_1 \), and noting that \( \frac{\partial f_i}{\partial y} \) is the Jacobian, \( J \), of the system we have obtained a set of linear equations of the form

\[ \sum_{j=1}^{n} W_{ij} \lambda_j = b_i \quad (i = 1, \ldots, n) \]  

(1)

where \( W = [I/hd_1 - J] \). This simple change achieves substantial economy in making a change of step size or order, in that only the diagonal elements of \( W \) need to be updated.

The size of realistic problems tackled by a package like CSMP III may vary from a modest few differential equations \( n \sim 10 \) to complex systems involving several hundreds of differential equations \( n > 300 \), with typical problems in the range \( 40 < n < 200 \). It follows therefore that of the total time spent on solving the differential systems by far the greatest part will be taken up in solving the systems of linear equations of order \( n \) which arise at each corrector iteration, as indicated above. Also, and closely linked to this part of the work will be the overhead to the integration process coming from the need to re-evaluate the Jacobian matrix of the differential system. An algorithm such as Gear's would be impractical if such a re-evaluation were required at each integration step. However, even the commonly accepted strategy of Jacobian re-evaluation by numerical differencing would not be feasible unless it was capable of being undertaken with an acceptable computational cost.

As we show later, the implementation of Gear's algorithm for large systems gives rise to computational problems of such a nature that the problem faced is not simply one of scale, but is
qualitatively; different from implementations such as that of Gear (1971A).

In general we iterate the corrector system until satisfactory convergence has been achieved. Error control and step size/order prediction follow closely the strategies proposed originally by Gear, though we have found it necessary to modify certain empirical coefficients in order to ensure appropriate build up in the order of the method.

In order to select this particular integration method the user of CSMP III (with the additional package) simply includes a particular METHOD card in his CSMP III program, indicates the accuracy he desires, and further includes a storage allocation subroutine of standard form by means of which his request for main storage may be tailored to the needs of his model. This is in line with the philosophy of the CSSL — the integration is quite transparent to the user, and he expects a successful solution to any problem, however badly conditioned, which he may pose. Therefore the implementation of a routine must be as self-contained, automated, and as fail-safe as possible.

MATRIX HANDLING

Earlier we made reference to the importance of efficiently manipulating the large matrices which occur in the corrector step of the Gear algorithm. As we must design a system capable of handling both small (n ~ 10) and large (n ~ 300) systems it is evident that both to economise on storage and also to minimise computing time we must take account of the sparsity of the Jacobian matrix of the system. We can in fact break the problem down into three components:

(i) We must find efficient processes for establishing the structure (i.e. the positions of the non-zero elements) of the Jacobian matrix of the system and for (independently) finding the values of these non-zero elements. From this information the matrix W can readily be obtained. Further, the particular form chosen above for W leads to efficient updating of its elements if either the order of the method or the current step size are changed.

(ii) We must make available a highly efficient and numerically stable set of routines for solving large sparse linear systems.

(iii) We must link (i) and (ii) above in a flexible manner so that the maximum amount of useful information can be carried over from one corrector iteration to the next, thus minimising the amount of computation required.

We are considering numerical procedures which involve \(O(n^2)\), \(O(n^3)\) or \(O(n^4)\) operations. The reader will fully appreciate the need for extreme care if he keeps in mind a system of order 100, where \(n^2\) and \(n^3\) take the values 10,000 and 1,000,000.
One approach to establishing the structure of the Jacobian matrix and the values of its non-zero elements which is very attractive involves techniques of symbolic differentiation. However, symbolic methods in turn give rise to a whole series of problems, any discussion of which lies outside the scope of the present paper. In the implementation for CSMP III a much more rough-and-ready technique was adopted. The position and value of the non-zero elements in the Jacobian are calculated by a simple one-sided difference approximation with a fairly small perturbation of the dependent variable. If a situation arises in which it is decided that the Jacobian is no longer numerically acceptable, a re-evaluation of the non-zero elements is made. Only under conditions of extreme difficulty is it necessary to re-evaluate the structure of the Jacobian. This strategy gave rise to a considerable saving in computer time, involving $O(n)$ operations rather than $O(n^2)$, and seems effective in all situations. The actual information on the sparse Jacobian is stored in one dimensional arrays to enable easy interface with the matrix routines required in the second problem area (ii).

The particular choice of routines for (ii) is aimed at giving a flexible yet robust means of tackling the solution of the corrector equation. After a symbolic re-evaluation of the Jacobian (that is, one to discover the positions of the non-zero elements), the first task undertaken is the determination for this particular matrix ($W$) of a near optimum pivot choice to minimise build-up of non-zero elements in LU factorization, bearing in mind at the same time the need to avoid numerically unacceptable pivots. Following this a symbolic factorization is carried out to determine the structure of the decomposed matrix before undertaking an actual numeric factorization. The map of the structure of $W$, and of its LU decomposed form is retained for as long as possible, while numeric values may change. The numeric factorization and the solution of a particular system are carried out as two separate tasks, and followed by iterative refinement of the solution.

Apart from the obvious reasons of efficiency this modular approach to the handling of the Jacobian matrix/$W$ matrix allows for greater freedom in recovery procedures following failure of the corrector process. For example three of the options open to us are as follows:

(a) A change of step size and/or order, i.e. a change in the diagonal elements of $W$.

(b) Numeric re-evaluation of $J$ and hence $W$, retaining the structural maps of $W$ and its decomposition, revising the pivot choice if necessary.

(c) Complete symbolic and numeric re-evaluation of $J$, choice of new pivots and determination of new structure of LU decomposed form.

This modular approach addresses the problem (iii) above as the only quantities likely to change from step to step are $W$. 
(subsequent to a change in step size or order) and its numeric factorization. It also allows us to make the system self-responsive to a rare occurrence such as the vanishing of a given pivot element.

All the sparse matrix routines used come from one IBM program product, 'Subroutine Library Mathematics' (SL-MATH).

In the following section we discuss the interlinking of the different modules to create a robust and flexible system.

**PROGRAM STRUCTURE**

The interface between CSMP III and the integration routine, which we have called ZZSTFX, is made by substituting a modified version of the CSMP III routine ZZSIMA. This routine is the controlling routine for the execution phase, and performs such functions as run initialization prior to initiating a cycle of calling a (one step) integration routine and output and status testing routines.

The modification to ZZSIMA is required for several reasons:

(i) We give the user the opportunity to allocate a main storage work-space, which is then automatically sectioned off into the 30 or so work arrays which are required. Thus we achieve a degree of dynamic storage allocation in FORTRAN, which is clearly important when the requirement for main storage is so very highly problem dependent. To give some idea of our storage requirements, we have run problems of up to \( n = 300 \) in a main storage of 300 k bytes. (It is relevant to state here that almost all calculations are done in single precision, and that CSMP III itself is basically a single precision system.)

(ii) We wish to get away from the requirement that output points should be integration points, which is inherent in the design of CSMP III. As a consequence, the integration methods can be completely output-bound. In contrast, our implementation uses the Nordsieck solution vector for interpolation to output points, and we have seen cases where fine output was requested and a Gear integration step encompassed 20 to 30 output points.

(iii) In modifying ZZSIMA, we take the opportunity to provide the user with the ability to specify the positions, in time, of known discontinuities. What is, in effect, a stop and restart procedure is automatically invoked at such 'time-cusps'.

Once control is passed to the integration routine, either a successful integration step is taken, or an appropriate diagnostic error message is generated — e.g. 'step size less than minimum specified'. As we have emphasized in Sections 3 and 4 above, the main problems of implementation arise in the corrector loop, and in re-calculating or modifying \( W \), which in our code is
Figure 1: THE CORRECTOR ITERATION

Indicator = 1

Branch on indicator

J: Numeric Evaluation

J: Structural Evaluation

Change d or h

Choose Optimal Pivots

Obtain Symbolic L/U

Obtain Numeric L/U

Solve System

Iterative Refinement

Test LTE etc

Reset indicator

Yes

Corrector Convergence

No

Following strategy set indicator

130

A. R. GOURLAY AND H. D. D. WATSON
also done in the corrector loop. The inter-relationship of the
different ways they may be linked together is best illustrated
diagramatically (Figure 1). On each entry to the corrector loop,
whether following corrector convergence or failure, or following
a step which has been accepted or rejected after an examination of
the local truncation error an indicator will have been set to
specify one of the following options:-

1. W unchanged
2. Change in step size (h) or order (d.)
3. A numeric re-evaluation of J is needed
4. A symbolic and numeric re-evaluation of J is needed.

The corresponding 4 paths through the code are shown in
Figure 1.

It will be noticed that iterative refinement of the solution
is included amongst the modules. This has been found to be
necessary because of the very badly ill-conditioned systems which
the user of CSMP III does create. In one instance, for example,
we found elements of W and the right hand sides varying over 26
orders of magnitude. When the reader recalls that we are working
in single precision, the need for iterative refinement becomes
clear. Even then a given system may fail to solve. If this
happens we attempt a better choice of pivot elements, as we also
do if a pivot element vanishes (to the machine resolution) after
a numeric re-evaluation of the Jacobian.

The strategy adopted following corrector failure (i.e. the
sequence of indicator settings) may be adjusted by the user, but
an acceptable strategy seems to be the following sequence:-

- Re-evaluate J numerically
- Half the step size (twice)
- Re-evaluate J numerically and symbolically
- Half the step size (until convergence or $h < h_{\min}$)

An interesting question arises relating to the difference, $\delta$,
used in calculating Jacobian matrix elements. We have found it
advantageous to relate this to $\epsilon$, where $\epsilon$ is a measure (in some
appropriate norm) of the accuracy of solution required, but there
would seem to be room for further study of this point, and indeed
most of the heuristics involved in the algorithm.

NUMERICAL EXPERIENCE

The experience which we have had to date with the ZZSTFX code
is of two kinds:

(i) In a preliminary phase of the work we used a diagonal
pivoting scheme with a series of tri-diagonal models
where both the range

$$S = \frac{|\text{Re}(\lambda_i)|_{\text{max}}}{|\text{Re}(\lambda_i)|_{\text{min}}}$$

(the $\lambda_i$ being the eigenvalues
of the system, $\text{Re}(\lambda_i) < 0$), which gives a measure of the
'stiffness' of the problem, and n, were variable in a region
\[ 1 \leq S \leq 10^5 \]
\[ 10 \leq n \leq 200 \]
From these and similar experiments we were able to verify the accuracy of the ZZSTFX code results, and also to establish the important facts, in comparison with STIFF that:-
(a) The advantage of the Gear scheme over STIFF could be maintained for large n using the sparse matrix routines.
(b) The run time for the Gear scheme was practically independent of S, while that for STIFF increased very rapidly with S. In this and other experiments we commonly recorded performance improvements in excess of a factor of 50 for the larger values of S, \( \sim 10^5 \).
(c) The run time for the Gear method was fairly insensitive to the accuracy requested, an important and un-anticipated side benefit.

(ii) We have now run the code against a series of 'real life' models made available to us by users of CSMP III. We quote below the performance enhancements obtained on these models with the ZZSTFX code.

<table>
<thead>
<tr>
<th>Nature of Model</th>
<th>Size of System n</th>
<th>Perf. improvement factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical Process</td>
<td>41</td>
<td>4.3</td>
</tr>
<tr>
<td>Electronic Circuit</td>
<td>49</td>
<td>9.2</td>
</tr>
<tr>
<td>Electronic Circuit</td>
<td>84</td>
<td>15.1</td>
</tr>
<tr>
<td>Public Utility</td>
<td>139</td>
<td>19.0</td>
</tr>
</tbody>
</table>

It should be noted that the performance factors quoted here will vary marginally with the particular configuration used.

One would very much like to find an objective measure of the efficiency of the code, such as number of corrector iterations, or derivative evaluations, but unfortunately no such measure is necessarily related to any indicator of efficiency, therefore we can only quote comparative timings between different routines on the same machine.

SUMMARY

We have shown how a version of Gear's algorithm which is efficient for large systems may be constructed and interfaced to a 'black-box' application package. The results we have quoted indicate substantial performance improvements, and our experience is that the code is extremely robust, and will only fail on exceptionally ill-conditioned models, i.e. badly scaled or highly
discontinuous. However it must be pointed out that alternative pivoting schemes are available, and we are unable to make comparative evaluations, or to discuss the merits of the choice of precision. Certain functions, the output of which depends on past history information (e.g. a hysteresis function), pose special difficulties for the algorithm, which we have not discussed. However, the importance of consideration of output requirements has been stressed. This has led us to adopt the Nordsieck formulation, in contra-distinction to other workers concerned with large scale systems such as [Brayton, Gustavson and Hachtel (1972A)].

Finally, we have touched on the requirement of the simulation user that integration be transparent and automatic, which is at sharp variance with a viewpoint commonly held amongst numerical analysts that the treatment of each problem must be tailored to its characteristics. We have attempted to go as far as seemed reasonable at the present time to meet the simulation user's requirements.
A PSEUDO DYNAMIC METHOD FOR SOLVING NONLINEAR ALGEBRAIC EQUATIONS

G. Hachtel and M. Mack
Dept. of Math. Sciences
IBM T. J. Watson Research Center
Yorktown Heights, N. Y., 10598, U.S.A.

ABSTRACT

A widely convergent method for solving nonlinear algebraic equations is described which combines the algorithmic features of Gear-type automated stiff ODE solvers and Davidenko-type Parameter Stepping methods. The method consists of variable-order variable-time step transient analysis, with prediction and truncation error control, in which the Newton iteration on the final time step gives the desired solution. The method is shown to be efficient in model problems taken from the applications area of semiconductor integrated circuits.
Section 1. Introduction

In some sense we are now in a computational era of high level, problem oriented languages and semi-general purpose "package" programs capable of solving broad classes of problems in given applications areas. Many such programs (for example the Electrical Network Design Program described in the paper by Hachtel, Brayton and Gustavson, 1971) already contain or will soon be augmented by automatic stiff ODE solvers and/or subroutines for solving nonlinear algebraic equations. Reported in this paper is a new method for solving nonlinear equations based on performing a pseudo-dynamic analysis of a related set of differential algebraic equations.

The new method may be thought of as generalizing, and, it is asserted, improving, the "parameter stepping" method of Davidenko, 1953, Freudenstein and Roth, 1963, Kizner, 1969, and Cermak, 1971 (A). The improvements stem from the incorporation of algorithmic elements from the already well known techniques of automated stiff ODE solving of the type described by Gear, 1971 (A), Gear, 1971 (B), or Brayton, Gustavson and Hachtel, 1972. The new idea is to transform the original algebraic problem (cf., Equation (1), below) into a related differential-algebraic problem (cf., Equation (3), below). The transformed problem is solved, or at least seems to be solved, dynamically. The mechanisms of prediction and truncation error control of the time step and polynomial fitting order are utilized in adjusting the rate at which the problem being solved at a given time step "grows" into the problem for which the solution is desired.

In the new method, the user trades off efficiency for insurance of convergence by specifying an upper bound for the truncation error and a lower bound for the time step. It is to be emphasized that at its most efficient (and its least likely to converge) the method is only as fast as (in fact, identical to) a straight Newton’s method. Thus, the new method is intended primarily for applications where convergence problems are anticipated.

The new method is quite easily implemented, especially in systems already containing a stiff systems solver.

Numerical examples arising from semiconductor integrated circuit design have been treated as model problems which demonstrate the effectiveness of the method. The exponential non-
linearities which pervade this problem class generally prevent a straight Newton-Raphson iteration from converging. The new method succeeds (without identifying the specific exponential type of nonlinearity) in these examples in a manner which clearly demonstrates the effectiveness of the improvements made to the parent "Parameter Stepping" method. The examples also show the new method to be distinct from, and superior to, another competitive method, known as the "Charge-Up" method *.

In this alternative method, the original algebraic problem is transformed into one of finding an equilibrium state of a related dynamic system (by actually solving the differential equations on the interval \([0, \infty]\)).

The present paper begins in Section 2 with a detailed qualitative and algorithmic exposition of the new method. However, for many details of implementation, the reader is referred to the earlier paper by Brayton, Gustavson, and Hachtel, 1972.

The two numerical examples treated in Section 3 are practical problems which actually arose in integrated circuit design applications and are believed to be typical of that problem class. The first example derives from analysis of the lumped network model of a semiconductor logic circuit and involved the solution of 68 algebraic equations. The second example treats the 3200 nonlinear algebraic equations which arise from differencing 2 PDE’s (the electron continuity and Poisson’s equations) on a 40x40 rectangular grid. In this larger problem, non-exponential nonlinear mechanisms are significant and sometimes dominant.

The paper concludes in Section 4 with a summary of what has (and has not, but should have) been done, and of what claims can be and cannot be (and, specifically, are not) made for the new method.

* Currently implemented in the IBM Program Product, ASTAP-II, for solving the differential algebraic equations of electrical networks.
Section 2. The DCES (Dynamically Controlled Evolution of Solutions) Method.

In this section we discuss a new method for solving the system of nonlinear algebraic equations

\[ f(u, v, s) = 0, \]  

where \(|f|^* = |u| + |v|\), \(v\) stands for the vector of nonlinear arguments of \(f\), and \(s\) for a known vector of "source" variables. It shall be assumed that a solution \(u_0, v_0\) of (1) is known for the case \(s = 0\), i.e.

\[ f(u_0, v_0, 0) = 0. \]  

The known solution \((u_0, v_0)\) will often, but not always, be the trivial solution \((0, 0)\). Our method is intended to treat the case where a straight Newton-Raphson, with the starting guess \(u_0', v_0'\), fails to converge when applied to (1). The new method attempts to overcome this difficulty by converting (1) into the pseudo-dynamic initial value problem (note \(g\) is to be understood as a scalar functional of the vector \(v\))

\[ \bar{s} = s t / T, \quad w = g(v), \]  

\[ f(u, v, \bar{s}) = 0, \quad 0 \leq t \leq T, \]  

which is to be solved at time points \(0 = t_1, t_2, \ldots, t_N = T\). The original equations (1) are seen to be solved whenever the Newton iteration converges whenever the Newton iteration converges at time \(t_N = T\). As stated so far, the "new" method is only a minor variant of the "Parameter Stepping" Method of Davidenko-Cermak (cf. Davidenko, 1953, and Cermak, 1971 (A) and his references).

The new method, which we call the DCES (Dynamically Controlled Evolution of Solutions) method attains its novelty, and, we assert, its effectiveness, by incorporating two new (in this context) algorithmic features, namely,

* When applied to a vector (e.g., \(f\)) the absolute magnitude sign \(|f|\) will stand for the dimension of the vector.
a) **Prediction** - i.e., prediction of the zero\textsuperscript{th} Newton iterate $v_{n+1}$ at time $t_{n+1}$ by fitting a $k$\textsuperscript{th} degree polynomial through the already converged time points $v_{n+1-(k+1)}$, $v_{n+1-k}$, \ldots $v_n$; and,

b) **Truncation Error Control** - i.e., controlling of the time step, $t_{n+1} - t_n$, and polynomial fitting order, $k$, by requiring that the difference, $\Delta_k = v_{n+1} - v_{n+1}^P$, between converged and predicted vectors is less than some suitable upper bound.

The predict feature functions by providing an accurate starting guess $v_{n+1}^P$ for the $n+1$\textsuperscript{st} Newton iteration. This reduces the number of Newton iterations required; more importantly, it helps prevent divergence of the iteration. Because of the existence of the "zero-source" solution, $(u_0, v_0)$, which can be used as initial conditions for (3), the accuracy of $x_{n+1}$ may normally\textsuperscript{*} be insured by taking suitably small time steps.

The truncation error control feature functions by limiting the difference, $\Delta_k$, between converged and predicted unknowns. It has been shown (by Brayton, Gustavson and Hachtel, 1972) that the truncation error, T.E., is given by

$$T. E. = \Delta_k \frac{(t_{n+1} - t_n)}{(t_{n+1} - t_{n-k})} \sim \frac{(t_{n+1} - t_n)^{k+1} v_{(k+1)}}{(k+1)}. \quad (4)$$

Thus if $\Delta_k$ exceeds a specified upper bound, then $v_{n+1}$ is recomputed for a smaller value of $t_{n+1}$. Alternatively, if $\Delta_k$ comes out to be significantly smaller than the upper bound, then $v_{n+1}$ is accepted but the next time step, $t_{n+2} - t_{n+1}$, (possibly the order $k$ will also be altered) is suitably increased. The increase will be such that the expected (on the assumption that the $(k+1)$\textsuperscript{st} derivative, $v_{(k+1)}$, will remain unchanged) truncation error will equal the specified upper bound.

\textsuperscript{*} The behavior of the method in unusual circumstances such as non-unique solutions has not yet been investigated.
It is to be emphasized that the slave scalar variable \( w = g(v) \) of (3) need never be computed, nor must there be a backward difference formula for computing polynomial approximations to \( \dot{v}^n_{n+1} \). In fact, problem (3) has no real dynamics whatsoever, and regardless of the truncation error, any converged Newton iteration yields a solution \( v_{n+1} \) of (3b) for the specified source vector \( s_{n+1} \).

The above fact may be used to compare the DCES method to (and distinguish it from) the so-called "Charge-Up" method \( \ast \). Instead of (3) the Charge-Up method solves the truly dynamic augmented system (here note that a matrix of constants, \( A \), must be suitably chosen, and that \( w \) is a vector which must be computed)

\[
\begin{align*}
\dot{s} &= s, \quad w = \dot{v} \quad , \quad (5a) \\
f(u, v, s) + A w &= 0 \quad . \quad (5b)
\end{align*}
\]

It is, therefore, actually a method for obtaining the desired solution \( u, v \) of (1) by finding an equilibrium solution of (5), i.e., solving (5) for \( 0 \leq t < \infty \). Thus transient analysis must, with its concomitant bookkeeping expense, actually be performed. Further, due to the augmentation, a larger linear system must be solved at each Newton iteration. Finally, unless the matrix \( A \) is artfully chosen, (5) may actually be unstable and unable to reach its equilibrium points. Thus we assert, and the experiments of the following section indicate, that the DCES method is intrinsically more efficient than the Charge-Up method.

However, before going on to the computational examples of the next section, we conclude this section with an algorithmic statement of the DCES method.

\[
\text{DCES - 1) Initialize: } t_0 = 0, \quad n = 0, \quad k = 1, \quad u = u_0, \quad v_1^0 = v_0, \quad h = h_{\text{MIN}}.
\]

\[
\text{DCES - 2) Set } \quad t_{n+1} = t_n + h.
\]

\( \ast \) cf. the footnote in Section 1.
A PSEUDODYNAMIC METHOD FOR NONLINEAR EQUATIONS

DCES - 3) Unless \( n = 0 \), set \( v_{n+1}^0 = v_{n+1}^P \), where

\[
v_{n+1}^P = \sum_{i=1}^{k+1} \gamma_i v_{n+1-i}.
\]  

DCES - 4) Solve (3) to obtain \( v_{n+1}^{+1} = v_{n+1}^+ \) by a Newton's or Newton-like method, using as a convergence criterion

\[
v_{n+1}^{+1} - v_{n+1}^+ = \delta_a + \delta_r \| v_{n+1} \|,
\]

where \( \delta_a \) and \( \delta_r \) are specified truncation error bounds, computed new \( h \) and \( k \) such that \( h \) is the largest value for which the \( \Delta_h \) estimated for that \( h \) and \( h \) is such that (8) is satisfied. If the inequality is not satisfied, reduce \( h \) and go to DCES-2.

DCES - 5) When \( \Delta_k = (v_{n+1}^+ - v_{n+1}^P) \) satisfies the inequality

\[
T.E. = \frac{\Delta_k h}{(t_{n+1} - t_{n-k})} = \Delta_a + \Delta_r \| v_{n+1} \|,
\]

where \( \Delta_a \) and \( \Delta_r \) are specified truncation error bounds, compute new \( h \) and \( k \) such that \( h \) is the largest value for which the \( \Delta_h \) estimated for that \( h \) and \( h \) is such that (8) is satisfied. If the inequality is not satisfied, reduce \( h \) and go to DCES-2.

DCES - 6) Set \( n = n+1 \) and go to DCES-2), unless \( t_n = T \), in which case stop.

A complete treatment of step 5 of the above algorithm is given on p.100 of the paper by Brayton, Gustavson and Hachtel, 1972. Although the procedure given in the above reference involves the unnecessary calculation of \( v_{n+1}^+ \), it completely subsumes the more efficient procedure and so need not be included here.

---

Various methods for obtaining the \( \gamma_i \) are discussed in Brayton, Gustavson, and Hachtel, 1972.
Section 3. Numerical Results on Model Problems
From Semiconductor Integrated Circuit Design

In this section we report the results of numerical experimentation with the method of Section 2. The two examples treated are practical problems which arise in semiconductor integrated circuit design. The first problem required solution of the nonlinear algebraic equations which describe the equilibrium behavior of an electrical network. The second problem required solution of the nonlinear difference equations arising from the two-dimensional spatial discretization of the semiconductor Poisson's and electron continuity equations. Both problems fit quite naturally into the formalism (3), and in both cases the results obtained compare quite favorably with prior results obtained with competitive alternative methods.

3.1 Example 1 - Equilibrium Analysis of a Nonlinear Electrical Network.

Example 1 derives from a high speed transistor logic circuit which was described in detail in another paper (Hachtel, Brayton, and Gustavson, 1971).

The network may be briefly characterized for our present purposes by noting that it consisted of 18 resistors, 6 nonlinear dependent current sources, 3 independent voltage sources and 7 ideal diodes. The diodes have nonlinear characteristics of the form

\[ I_d = I_s (e^{\theta V_d} - 1) . \]

It may be therefore shown that

\[ |v|^* = 13 , \quad |s| = 3 , \quad |u| = 55 , \quad |f| = 68 , \]

which establishes this example as a slightly smaller than average, but still substantial model problem.

The first experiment performed was to investigate the effect of the truncation error controls \( \Delta_a \) and \( \Delta_r \) (see step 5 of the

* Recall that \(|v|\) stands for "dimension of vector v".
A PSEUDODYNAMIC METHOD FOR NONLINEAR EQUATIONS

DCES algorithm) on $N$, the total number of time steps, and on $\mathbf{v}_{\text{TOT}}$, the total number of Newton iterations. The results are summarized in Table 1, and were obtained for Newton iteration error controls of $\delta_r = 0.05$, $\delta_a = 0.005$ (see discussion of step 4 of the DCES algorithm).

<table>
<thead>
<tr>
<th>a</th>
<th>r</th>
<th>Time Steps (N)</th>
<th>Iterations $\mathbf{v}_{\text{TOT}}$</th>
<th>$v_1(T)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>0.005</td>
<td>0.005</td>
<td>10(78) * *</td>
<td>43(79) * *</td>
</tr>
<tr>
<td>2)</td>
<td>0.01</td>
<td>1</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>3)</td>
<td>0.1</td>
<td>1</td>
<td>6</td>
<td>23</td>
</tr>
<tr>
<td>4)</td>
<td>1</td>
<td>10</td>
<td>5 *</td>
<td>-</td>
</tr>
</tbody>
</table>

* On fifth time step, Newton Iteration diverged with exponential overflows.

** Parenthetical numbers are for "Charge-Up" method.

Effect of Truncation Error Controls on Computational Work for Example 1.

Table 1.

Table 1 demonstrates that the computational work depends significantly on the truncation error controls. Also demonstrated is an optimum value for these controls, indicative of the following trade-off. First, there is no point in maintaining excessively small truncation error. Since there are no real dynamics present we are really interested only in Newton convergence and not time discretization convergence on any given time step. Note that the final value of $v_1$ (the first component of the nonlinear unknown vector) is essentially independent of the truncation error controls. However, if controls are too loose, then the prediction becomes inaccurate, and indeed may, as in time step 5 of case 4 of Table 1, take the first Newton Iterate right out of the region of Newton convergence.

Note from the 3rd and 4th columns of Table 1 that the average
number of Newton steps per time step is about 4 in each convergent case. This is somewhat surprising in view of the data of Table 2, which shows the time evolution of quantities of interest for case 3 of Table 1.

<table>
<thead>
<tr>
<th>Step n</th>
<th>Time $t_n$</th>
<th>Time Step $t_n - t_{n-1}$</th>
<th>Order (of fitted polynomial)</th>
<th>Newton Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.1</td>
<td>.1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>.2</td>
<td>.1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>.3</td>
<td>.1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>.53</td>
<td>.23</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>.88</td>
<td>.35</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1.</td>
<td>.12</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

$\nu_{TOT} = 23$

Time Evolution of Time Advancement Indicators.

Table 2.

Table 2 shows that more than half of the total number of iterations were expended in the first 2 time steps. It is believed that the peculiar distribution of Column 5 in Table 2 (which seemed to be a pervasive aspect of all experiments performed) is indicative of the effect of a prediction which becomes increasingly accurate as time evolves. Note that the final time step was taken for order = 2, suggesting an accurate final prediction. Note also that the small final time step ($h=.12$) would have been much larger if the final time had not been reached.

Case 1 of Table 1, which corresponds to default values of error controls in our implementation, may be compared to the results of solving the model problem with the "Charge-Up" method of (5). Using the same (default) values of $\Delta a$, $\Delta r$, $\delta a$ and $\delta r$, the charge up method required 78 time steps and 79 Newton iterations (in parenthesis in row 1 of Table 1).
For a fair comparison one must account for the bookkeeping overhead involved in taking 79 (instead of 10) time steps. Also, in the "Sparse Tableau" implementation used in these experiments the sparse matrix code for the charge up case was nearly twice as large as that for the DCES method, which, it must be concluded, is significantly faster than the charge-up method in this example.

3.2 Example 2 - A Semiconductor Device (IGFET) Described by Partial Differential Equations.

In this example, it is desired to solve, on a rectangular domain, of two space dimensions x and y, the PDE

$$\nabla \cdot J = \nabla \cdot \left\{ \mu \Phi - \rho \nabla \Phi \right\} = 0 \quad (9)$$

and

$$\nabla^2 \psi - \rho = \nabla^2 \psi - \left( e^{-\psi} N_c e^\psi \Phi \right) \mathbb{D}(x, y) = 0 \quad (10)$$

for a sequence of specified Dirichlet boundary values $\Phi_D$. On the boundary $\partial \Omega$, we have either $\Phi = \psi = 0$ or $\nabla \phi \cdot n = 0$, except on the line $x=0$, $0 \leq y \leq 1/3$, where we have $\Phi = \psi = \Phi_D - c$, $c = \text{const}$. By this means we obtain a device transfer characteristic (i.e., $I = \int_x (\Phi_D) d \mu \cdot \mathbb{D}$).

This problem is elsewhere described in detail, and can be characterized for present purposes by noting that a system of 3200 nonlinear algebraic equations is obtained by spatial discretization on a 40x40 grid of centered finite differences. Thus, in the notation of (1),

$$f = \begin{cases} \frac{\Phi}{J} & \text{at } i, j \in \Omega^{40 \times 40} \\ \Phi & \text{at } i, j, l \in \Omega^{40 \times 40} \end{cases} \quad (11)$$

$$v = \begin{cases} \Phi^2 & i, j \in \Omega^{40 \times 40} \\ \psi & i, j, l \in \Omega^{40 \times 40} \end{cases}$$

---


The system Jacobian, $\frac{\partial f}{\partial v}$, has a five diagonal structure of block 2x2 matrices, and the only deviation from the algorithm of Sect. 2, is that in step 4, instead of a straight Newton iteration

$$v = -f, \quad v = v + \Delta v,$$

(12)

a modified Stone iteration (Stone, 1968)

$$\left( \frac{\partial f}{\partial v} + S \right) \Delta v = -f, \quad v = v + \Delta v,$$

(13)

where $S$ (Stone's "Fill-in Blocking Matrix") is taken. It is easily shown that this does not alter any of the arguments of Section 2. All that is required is that the above Stone iteration converge on each time step, since again there are no real dynamics present.

In this example, the "zero source" solution $u_0, v_0$ of

$$f(u_0, v_0, 0) = 0$$

(2)

is obtained by setting $\phi(x, y) = 0$ throughout the domain $\Omega$ and by solving Poisson's equation by itself. Also, in contrast to example 1, the solution values are of interest at each time point, since we desire to sweep out a transfer characteristic (i.e. $I(\phi_d)$ vs $\phi_d \leftrightarrow I(t)$ vs. $t$).

Various numerical experiments were performed which gave the same results and indications as those reported above in Example 1, and are not repeated here (e.g. the existence of both optimum and catastrophic truncation error controls $\Delta \phi$ and $\Delta I$). However, some results which are unique to this example are given in Table 3.
A PSEUDODYNAMIC METHOD FOR NONLINEAR EQUATIONS

<table>
<thead>
<tr>
<th>Total Applied Current</th>
<th>Voltage $D(t)$</th>
<th>Degree of Convergence</th>
<th>Iterations per Time Step (No Predict)</th>
<th>(Predict)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sim 10^{-9}$</td>
<td>4</td>
<td>$\sim 10^{-6}$</td>
<td>300</td>
<td>30-100</td>
</tr>
<tr>
<td>$\sim 10^{-4}$</td>
<td>100</td>
<td>$\sim 10^{-6}$</td>
<td>150</td>
<td>25</td>
</tr>
<tr>
<td>$\sim 1$</td>
<td>1000</td>
<td>$\ll 10^{-6}$</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

* $I = \sum_j I_{i,j} = \sum_j (\nabla x)_{i,j} \Delta y_{j} \approx \int \nabla x(x,y) dy$

** $\Gamma = \left\{ \frac{i}{40^2} \sqrt{\sum_{i,j} (r_{i,j} \Delta x_i \Delta y_j)^2} \right\} / \sum_{i,j} I_{i,j}$

where

$r_{i,j} \Delta x_i \Delta y_j = \int \Delta x \int \Delta y (\nabla \cdot J)$, $1 \leq i,j \leq 40$

*** $k=1$ (i.e. linear or first order prediction used throughout)

Effect of Prediction on Stone Iteration Count

Table 3

To discuss Table 3, it is first necessary to establish some definitions. The quantity $I$ (see the asterisk in column 1 of Table 3) stands for the total x-direction cross sectional current through the device. Since, as mentioned above $\phi(x,y)=0$ for $\phi_D=0$, it follows from (9) that $I=0$ for $\phi_D=0$. Also note (double asterisk) that a quantity, $\Gamma$, has been introduced to denote the degree of convergence of the nonlinear system (11) at a given time point. $\Gamma$ is seen to be the ratio of the average (over the 40x40 grid) residual to the largest x-directed grid cell current $I_{ij}$. Note (double asterisk) that by the divergence theorem the residual itself may be seen to be the sum of the cell currents leaving a given
cell minus the sum of those entering it. Hence $\Gamma \ll 1$ indicates that overall conservation of current, which is a property of the solutions of (11), is approximately present. That is, (singly asterisk in Table 3) $r^i = 0$ implies that $I_i$ is independent of $i$, $i=1, 2, \ldots 40$, which is the desired property of convergence.

With the above definitions in mind Table 3 may be properly interpreted. The first comparison to be made is of the rightmost columns of Tables 2 and 3 (note column 2 of Table 3 stands for both the boundary value $\varphi_D$ and the time $t$). This comparison suggests that the prediction, although always for $k=1$ in the present example, becomes increasingly accurate, at least for low to medium current levels, as the voltage (time) level increases, just as it did in Example 1.

The time steps produced by the algorithm were essentially uniform (data to support this statement is omitted). Thus, since column 4 of Table 3 corresponds to an experiment in which prediction is suppressed, this column may be regarded as representative of the Davidenko-Cermak parameter stepping method (Davidenko, 1953, Cermak 1971 (A)). Comparison of columns 4 and 5 indicate a substantial pay off for the prediction feature of the algorithm.

Another interesting comparison which can be made of Tables 2 and 3 concerns the increase in the Number of Newton Steps per time step, which, for medium to high current levels, is about a factor of 10 times larger in Table 3. This ratio (the factor of 10) is surprisingly small in view of the facts that

a) Example 2 ($|v|=3200$) has more than 200 times as many nonlinear unknowns as Example 1 ($|v|=13$); and,

b) in Example 2, the iteration (10) is not a Newton iteration and is not theoretically expected (Stone, 1968) to converge quadratically.

As a final comment on Example 2, it has been noted from graphical studies of the solutions that the dramatic impact of the prediction

* Indeed, the 10hr runs on the IBM/360 Model 91 which had been required to produce the desired device transfer characteristic were reduced to about 3 hour runs.
process seems to depend on a peculiar property of semiconductor devices. The peculiarity is that contact to the device (the point at which the Dirichlet boundary value is imposed) is always made in high conductivity regions which cannot support any appreciable electric field (i.e. $-\nabla \Psi$ must be small). However, changes in $\Psi$ at a given time are imposed originally only at the Dirichlet boundary. Without prediction, these changes must work their way through the high conductivity regions before reaching, as the Stone Iteration (13) proceeds, the low conductivity regions in the interior of the device. Thus, large transient fields appear in the high conductivity regions, causing various numerical difficulties of nonlinearity (principally of the mobility $\mu$ in (9) which depends on $\parallel \nabla \Psi \parallel$ and scaling. Since all converged solutions have no field at all in these regions, prediction at any order will yield a starting guess for the Stone iteration which also has no field. Thus, the numerical difficulties are avoided.

4. Conclusions

A novel algorithm for solving nonlinear algebraic equations has been described for cases where Newton convergence difficulties are anticipated. The algorithm is easily implemented, especially in programs already containing automatic stiff differential equation solvers. The method has been tested in examples arising in semiconductor integrated circuit design. In these examples it has been shown to be more efficient than the "Parameter Stepping" and "Charge-Up" methods. The new method may require user intervention in specifying truncation error and minimum time step controls in order to get maximum efficiency. However, it is claimed that these controls are more systematic and have a more elegant theoretical interpretation than those of the competitive methods listed above.

Exponential nonlinearities played a key role in preventing straight Newton convergence in the examples treated. It should be noted that there exist in the electrical engineering literature specialized methods for treating such nonlinearities. These specialized methods solve problems similar to Example 1 in $\sim$15 iterations. Thus, the method of this paper may be regarded with

some objectivity as a general purpose, widely convergent method which is more reliable than, yet comes within a factor of 2 or so of the efficiency of, the best special purpose programs for integrated circuit analysis.

It is not claimed that such favorable efficiency can be obtained in all applications, nor is it asserted that the method will be always convergent even in the absence of anomalous effects such as non-unique solutions. This last disclaimer is based on the fact, demonstrated in Table 1, that the zealous user may, in his search for the most efficient control parameters, reach non-convergence through a catastrophic prediction.

Sorely missing from the investigations to date are theoretical studies. A theorem about guaranteed convergence is believed to exist in cases where the preconditions of the Newton-Sphere of convergence theorem * are met. Also lacking are comparisons to other known widely convergent methods for nonlinear equations, as well as treatment of whatever "classical" examples or significant model problems of other application areas which doubtless exist in the literature. Hopefully these missing items will become topics of further research.

THE VALIDATION AND COMPARISON
OF PROGRAMS FOR STIFF SYSTEMS

T.E. Hull
Department of Computer Science
University of Toronto

ABSTRACT

Software for solving stiff systems of ordinary
differential equations should be easy to understand,
reliable, efficient and convenient. Good structuring,
the use of appropriate language conventions, and proofs
of correctness contribute towards ease of understanding.
Careful comparisons are needed to assess reliability
and efficiency. Further testing and packaging help
make the system convenient to use. Each of these topics
is discussed, and comparisons between five different
methods are presented. An adaptation of Gear's backward
difference method, the second-derivative method of
Enright, and an extrapolation method based on ideas due
to Dahlquist and Lindberg are reliable and reasonably
efficient, although for each of these methods there is
a class of problems for which the method is somewhat
less efficient than the other two. On the other hand,
an implicit Runge-Kutta method and a modification of
the generalized Runge-Kutta method due to Ehle and
Lawson did not compare favourably with the first three.

1. INTRODUCTION

The development of good mathematical software
requires us to pay careful attention to several
different aspects of our programs. To begin with, it is
important that the programs be easy to understand, and
the key to understanding a program is its structure. We therefore begin in the next section by describing an overall structure that we have used in the development of programs for solving ordinary differential equations. This structure has turned out to be very helpful in the development of programs for both stiff and non-stiff systems of equations. Then in section 3 we discuss some of the language conventions that must be considered when a well-structured program is to be expressed in an existing programming language. Questions about proving correctness of the resulting program are discussed briefly in section 4.

To assess their relative merits, extensive comparisons of five different programs have been made over a representative collection of practical problems. Preliminary results and conclusions are summarized in section 5. Reliability is measured in terms of the amount by which the error committed by a method exceeds the prescribed tolerance. Efficiency is measured in terms of execution time, number of function evaluations, number of Jacobian evaluations, etc.

It is concluded that three of the programs are on the whole quite good, how good in each case depending to some extent on certain characteristics of the problem. These three programs are: (i) an adaptation of Gear's backward difference method, (ii) Enright's second derivative method, and (iii) an extrapolation method based on ideas due to Dahlquist and Lindberg. An implicit Runge-Kutta method, and a modification of the generalized Runge-Kutta method of Ehle and Lawson were generally less efficient and less reliable.

Further testing and packaging are needed before the programs can be considered convenient. These points are discussed in section 6.

An earlier version of some of the ideas in the present paper appeared elsewhere (Hull (1973A))†, but the discussion there was limited to programs for solving non-stiff systems of equations.

2. PROGRAM STRUCTURE

The key to understanding a program is its structure. If its structure is well-organized and simple, a program will be relatively easy to maintain, or to

†See end of paper for this and certain other references.
modify if necessary, and proofs of its properties will be relatively easy to carry out. Two such programs are also more easily compared.

During the last few years, several of us at the University of Toronto have been developing and testing programs for solving ordinary differential equations. We have been primarily interested in comparing methods (Hull, Enright, Fellen and Sedgwick (1972A)) and in proving correctness (Hull, Enright and Sedgwick (1972A)). In the course of these investigations, we have come to believe that there is a common overall structure we could profitably adopt for all our programs. So far, the structure that has evolved has been successfully used in the design of several programs for solving non-stiff systems. This material is to be described elsewhere ("A Structure for Programs that Solve Ordinary Differential Equations", T.E. Hull and W.H. Enright, Department of Computer Science Technical Report, University of Toronto, in preparation). It is currently being tried with stiff methods (see Enright's paper at this symposium).

To describe the structure, we must first discuss the calling sequence, which we divide into the following four parts:

(1) problem specification - N, FCN, X, Y, XEND, TOL, and possibly JACOB;

(2) helpful parameters - IND, YMAX, SCALE;

(3) "own" variables (if necessary);

(4) workspace (if necessary).

The number of equations is N and their defining functions are determined by a subprogram called FCN. The initial values of X and Y are to be replaced, respectively, by XEND and an approximation to Y at XEND. The approximation is to be within a tolerance, TOL, per unit step. (This last point must be made more precise, but the details need not concern us here.) A subprogram, JACOB, for evaluating the Jacobian matrix may also be required, as it is with methods for solving stiff systems.

We have found it helpful to include three other parameters in the calling sequence. IND is an indicator
that is set equal to 1 on initial entry. It is then changed by the program to denote the current state of the computation, such as "after a successful step", "after a failure", etc. In this capacity IND can be used to permit efficient reentry to the program with new values of XEND. IND is also used to indicate what type of error has been encountered in case an error exit is necessary.

YMAX is the user's estimate of a bound on Y, and is needed by the program to determine a suitable minimum step-size, HMIN. (We argue that there is no point in trying to make the truncation error in an individual step, which is approximately H*TOL, as small as the roundoff error, which is bounded by EPS*YMAX when H is small. This leads us to setting HMIN equal to a small multiple, say 10, of EPS*YMAX/HMIN.)

SCALE is the user's estimate of the "scale" of a problem. Some such measure is needed to enable the user to control the reliability of a method. We have usually used HMAX as the measure of SCALE. However, a good choice for HMAX depends on the method as well as the problem, and it would be better if the user did not have to know so much about the method. For non-stiff problems we have what we believe is a satisfactory alternative to HMAX, but we have so far not decided what we should require for stiff problems.

The "own" variables do not appear in the parameter list in Algol 60, but they may be needed in Fortran, since in ANSI Fortran we cannot assume that values for the preceding step-size, the error estimate, and so on, are available on reentry to the program. These values are needed for efficient reentry.

Workspace also does not need to be provided in the parameter list in Algol, but must be provided in Fortran if we are to allow variable dimensioning.

We can now describe the overall structure that we have adopted. It is shown in Figure 1. The fact that all our programs can be fitted into the same structure makes them much easier to understand and compare. The fact that each part of a program is relatively self-contained also helps. For example, having all calculations related to the preparation stage (choice of step-size, etc.) in one place makes the program much easier to understand, and is particularly helpful when trying to prove something about the program.
VALIDATION AND COMPARISON OF PROGRAMS

INITIALIZATION (calculate HMIN, HMAX from TOL, SCALE, etc., and determine maximum order if necessary)

...VALIDITY CHECK (e.g., check HMIN and HMAX)

REPEAT FOLLOWING UNTIL EXIT IN 4TH STAGE:

PREPARE (calculate step-size, also slope, Jacobian, order, if needed)

CALCULATE (make calculations for one step)

ESTIMATE (estimate error)

......DECIDE (update if acceptable and return if finished, or error exit if necessary; otherwise try again)

END OF REPEAT BLOCK

Figure 1. Overall structure of a program for solving ordinary differential equations. The dots indicate possible exit from the program.

3. LANGUAGE FACILITIES

Once the overall structure of a program is understood, we can begin to examine the details for each part, which will of course depend on the particular method being implemented. Eventually we reach a level of refinement that can be described in a particular programming language.

With this top-down approach to program development, we are led quite naturally to a programming language version that is "well-structured", in the sense in which that term is currently used in relation to programming languages. We have found that it is too much of a nuisance to try to use only a basic minimum of control constructs, such as DO-WHILE and IF-THEN-ELSE. We have found it is convenient to have a CASE construct...
as well, especially in the PREPARE stage of the program, where several cases have to be considered, such as "on initial entry", "after a successful step", etc. We have also found we needed the ability to exit from a construct, including the ability to make an error exit from the entire program.

Most of our programming so far has been in Fortran. To implement our "well-structured" programs in Fortran has led us to adopt some very restrictive conventions. We believe that the resulting disciplined use of Fortran has helped make our programs quite a bit easier to follow, at least once one knows the conventions. The details are explained elsewhere (Hull (1973A)), but the two examples shown in Figure 2 illustrate the main idea.

Statement numbers and GO TO's are used only in ways such as those shown in the examples. The program can therefore be understood without them. In fact, they can be inserted automatically. (A preprocessor to accomplish such a task has been developed by A.E. Sedgwick and C.A. Steele, and is described in a Technical Report of the Department of Computer Science at the University of Toronto, which is to appear under the title "DEFT - A Disciplined Extension of Fortran".)

4. PROVING PROPERTIES OF PROGRAMS

The ultimate in validation of a program is a proof that the program is correct. Of course there are many different senses in which we might like to prove a particular program correct, so perhaps it is best to speak about proving properties of a program.

Usually we want a program to be correct at least in the sense that it actually carries out the steps in a particular algorithm that we originally had in mind. Most of the proof of correctness in this sense is made possible, at least in practical situations, by careful structuring, and judicious choice of language facilities, as suggested in the preceding two sections.

There are many other properties that we might want to prove about a program. These will usually require a more detailed analysis of the program. For example, we have found that some careful attention to the semantics of floating point arithmetic may be needed to develop a
VALIDATION AND COMPARISON OF PROGRAMS

IF(boolean) GO TO n1
GO TO n2
C THEN
n1

n2
n3 CONTINUE

DO n1 I = 1,N

C ...EXIT DO IF CONVERGENCE
C CRITERION IS SATISFIED
IF(boolean) GO TO n2
n1
n2 CONTINUE

Figure 2. Two examples to illustrate a disciplined use of Fortran for "well-structured" programs. All statement numbers and GO TO's are used only in these special ways and the program can be understood without them.
rigorous proof that a particular program will always terminate.

I have been especially interested in proofs of correctness in a sense that guarantees something about the reliability of a method, at least over a particular class of problems. The essential idea is to prove that, for some particular class of problems, the program will always produce an approximation at XEND with the following property:

there exists a function $z(x)$ which interpolates the initial value and the computed approximation, and which satisfies

$$\|z' - f(x, z)\| < c \tau,$$

where $c$ is some known constant, and $\tau$ is the tolerance.

(The function $f$ is of course the function that defines the original differential equation.) We have called such theorems "effectiveness theorems". Further details and references are given by Hull, Enright and Sedgwick (1972A).

Unfortunately, such theorems have so far not been very carefully tied to specific programs, and roundoff is usually neglected in their proofs. They have also usually been restricted to non-stiff, linear problems. However, Enright (1972A) has considered stiff systems, and Sedgwick (1973A) has obtained results for a class of non-linear equations.

The results just referred to can no doubt be widened and improved to a considerable extent. However, it must be admitted that, by themselves, they are not likely to guarantee practical error bounds for any but rather limited classes of problems. Nevertheless, there are at least three other ways in which such results are helpful:

(1) proving effectiveness theorems helps to validate programs in the sense that it is not unreasonable to consider such results, for example for linear problems, to be necessary conditions to be satisfied by a program;
(2) the best value for the coefficient \( c \) can usually be found for linear equations, and comparing the corresponding values for two different methods provides a measure of their relative reliabilities;

(3) for non-stiff linear equations we can obtain \( c \) in terms of \( h\|A\| \), where \( h \) is the step-size and \( \|A\| \) is a norm of \( A \), and this enables us to relate the determination of a suitable \( \text{HMAX} \) to \( \text{SCALE} \), if we suppose \( \text{SCALE} \) is a bound for \( \|A\| \); our own policy has been to have \( \text{HMAX} \) computed so that we can guarantee \( c < 10 \), for linear problems with \( \|A\| < \text{SCALE} \), as long as \( h < \text{HMAX} \); (it should be pointed out that \( \text{HMAX} \) will depend on the order in variable order methods).

As mentioned earlier, we have not yet decided on a good way to handle this last point for stiff systems. However, I believe that Enright's theorem provides a reasonable basis from which we might try to derive a suitable approach. In any event, these three comments show that effectiveness theorems can have important implications quite apart from what they guarantee about the reliability of results for certain classes of problems.

5. COMPARISONS

There are at least three distinct reasons for testing programs. One is to uncover, if possible, any bugs that remain in the program; we are convinced that the program structure we have adopted, along with the language conventions, have very much reduced the possibility of any such bugs, and those that do occur are easy to find. Another reason for testing a program is to determine its reliability, especially over classes of problems not covered by effectiveness theorems. A third reason is to assess its efficiency.

Comparing methods on the basis of their reliability and efficiency has been one of our main interests at Toronto. In the paper by Hull, Enright, Fellen and Sedgwick (1972A), an attempt was made to put such comparisons on a reasonably firm basis. Problems, methods and criteria were carefully defined, and extensive experiments were performed. Conclusions were drawn
about the relative merits of Runge-Kutta, variable-order Adams, and extrapolation methods.

The testing program, called DETEST, has been modified slightly and rewritten in a well-structured form. A description of this program, and how to use it, is to appear as a Technical Report entitled "DETEST: a Program for Comparing Numerical Methods for Ordinary Differential Equations", co-authored by Hall, Enright, Hull and Sedgwick.

A new version of DETEST for testing stiff methods has also been developed and a Technical Report about it will be prepared in the near future. At present it uses 13 systems, with at most 10 equations each. (Larger systems are also available, but they are expensive.) It tests methods separately over two ranges of integration for each system, one including the transient phase and the other starting at a point where the transients have died out. Three tolerances are used with each system, $10^{-2}$, $10^{-4}$ and $10^{-6}$.

To assess the efficiency of a method, statistics are collected on computing time, overhead time, function calls, Jacobian evaluations, and matrix inversions (or triangular decompositions). To determine reliability, the maximum error in units of the tolerance is noted, and so is the fraction of steps on which the true error exceeds the tolerance. These statistics are obtained for each problem, and summaries over different classes of problems are also printed out.

The cost of testing one fairly good method is approximately $30, compared to only about $7 for a reasonably efficient method with the non-stiff version of DETEST. In either case, most of the cost is in determining reliability rather than efficiency, and this part of DETEST can be turned off if one wishes. (This would be the case if one wanted to make many runs with only slightly different values of a particular parameter in a method, in order to determine a good value for the parameter.)

Some results with the stiff version have now been obtained for the following five methods:

(1) An adaptation of Gear's backward difference method (Gear (1971A));
VALIDATION AND COMPARISON OF PROGRAMS

(2) Enright's second derivative method (Enright (1973A));

(3) An extrapolation method based on ideas of Dahlquist (1963A) and Lindberg (1971A);

(4) An implicit Runge-Kutta method based on formulas proposed by Butcher (1964A);


A preliminary analysis of these results has been made by Enright. The main conclusion is that the first three methods are quite reliable, and about equally efficient. With only one exception, the observed errors never exceed the tolerance by more than a factor of 2. (The exception occurs with Gear's method on one problem, when the error exceeds the tolerance by a factor of 27 on one or two steps.)

However, for each of these three methods there are problems for which the method is somewhat less efficient than the other two. For the first method, problems whose eigenvalues have non-zero imaginary parts cause the most difficulty. For the second one, it is the more highly non-linear problems that cause the most difficulty. For the third method, it is the problems with smaller tolerances. (This last remark is not surprising, considering the fact that the third method is a fixed order method.)

The implicit Runge-Kutta method is generally less reliable and less efficient; in fact, it was unable to complete some of the problems. The generalized Runge-Kutta method was unreliable on some of the non-linear problems, and somewhat less efficient overall.

We hope to publish the detailed results at a later date, along with a more thorough discussion of the conclusions.

6. CERTIFICATION, DISTRIBUTION, ETC.

Let us suppose that we have developed a collection of programs for solving ordinary differential equations, and let us suppose they have all of the desirable
features outlined in the preceding sections. Their structure is simple and easily understood, the programs themselves are easy to read, and major properties of the programs have been rigorously established. The programs have also been tested carefully, and they have turned out to be relatively reliable and efficient in comparison with others.

The point that I am coming to is that there is still a lot of work to be done. Such programs still need to be properly "packaged" so that they are easily accessible to a wide variety of users. This will involve further testing, especially since different machines, and different compilers on the same machine, must be provided for. Coordination of efforts, the development of control programs, careful documentation, and so on, will also be needed.

Fortunately, interest in this sort of activity is growing, as is evidenced by, for example, the NATS project (Boyle et al (1972A)) in North America and the NAG project (Ford (1973A)) in Britain. As an indication of how much effort is required for the packaging of good mathematical software, it can be pointed out that EISPACK cost approximately $500,000, in spite of the fact that a good deal of both people time and machine time was contributed at various test sites, without being charged to the project.

EISPACK is a good example to consider. It began with the very well designed and carefully tested programs for solving eigenvalue problems that appear in the handbook of Wilkinson and Reinsch (1971A). It will culminate in the publication of a user's manual (Smith et al (1973A)). This project has therefore been concerned entirely with the packaging phase, and its size ought to be carefully noted by anyone interested in the development of portable mathematical software.

One other lesson we can learn from EISPACK is about the need for a control program. Such a program has been developed at Argonne for the purpose of translating a user's description of his problem (is the matrix symmetric? is it positive definite? are the eigenvectors required? etc.) into an appropriate sequence of subroutine calls.

With programs for solving ordinary differential equations we of course have nothing comparable to the
handbook on matrix calculations by Wilkinson and Reinsch. In fact, much of the current work on programs for solving ordinary differential equations can be viewed as leading to the development of such a basic collection of programs. This development will be helped very considerably if we can find some measure of agreement on problem specification (and hence on calling sequences), and also on program structure. It may be difficult to achieve wide agreement, but even some agreement would be helpful, and it is my hope that the ideas outlined in this paper will be at least some contribution towards this objective.

Eventually, we will have to consider control programs. They will be more difficult to design than they are for matrix calculations (where, incidentally, most of the difficulty seemed to be with shortcomings of the programming language). The control program I envision would be able to make estimates of such parameters as SCALE and YMAX, if the user does not wish to exercise his option of providing them himself. The control program would be able to handle boundary value problems, as well as initial value problems, and it would normally expect the user to specify his accuracy requirements in global terms. The control program would select appropriate subroutines, depending on information provided by the user, if any, regarding stiffness, complexity of function evaluations, etc.

These objectives seem to me to be more difficult to achieve than the corresponding ones for matrix calculations. Nevertheless, I believe they show the direction in which we should try to go. In my opinion, we should place more emphasis on good software packaging than has been the case in the past.
REFERENCES

1973


1972


1. Introduction. In this paper results obtained by Leif Abrahamsson, Herb Keller and myself are discussed*. Consider a system of ordinary differential equations

\[ L(y) = \varepsilon (d^2y/dx^2) + A(x)(dy/dx) + B(x)y = F \]

in the interval \(0 \leq x \leq 1\) with boundary conditions

\[ y(0) = \alpha, \quad y(1) = \beta. \]

Here \(y = (y_1, \cdots, y_n)^T\) is a vector function, \(A, B\) are \(n \times n\) matrices which depend smoothly on \(x\) and \(\varepsilon > 0\) is a small parameter. Let \(\text{e}(A)\) denote the eigenvalues of \(A\). We assume that \(\text{Re} \quad \text{e}(A) \neq 0\). In this case the solutions of (1.1), (1.2) are smooth except in boundary layers of thickness \(\varepsilon |\log \varepsilon|\). We shall indicate how to derive asymptotic expansions

\[ y(x, \varepsilon) = \sum_j \varepsilon^j y_j(x, \varepsilon), \]

where the \(y_j\) are solutions of the reduced equation

\[ A(x)(dy/dx) + B(x)y = F, \]

and equations with constant coefficients

\[ \varepsilon (d^2y/dx^2) + A(k)(dy/dy) = G_k, \quad k = 0, 1. \]

*See citation to department report at end of this paper.
Various difference schemes are considered and it is indicated how to derive asymptotic expansions in \( h \) and \( \varepsilon \). Here \( h \gg \varepsilon \) denotes the stepsize of the difference scheme. The functions in the asymptotic expansions are solutions of difference analogs of equations (1.4), (1.5). Therefore the approximation is successful only if it works for the equations (1.4) and (1.5). It is shown that only approximations of low order accuracy guarantee uniform convergence in \( \varepsilon \). However, one can use the asymptotic expansions to increase the accuracy by a Richardson procedure.

The last part of the paper is devoted to the question of local mesh refinement. It is shown that the process is not always successful. For example using centered differences everywhere does not always work.

2. Asymptotic Expansion. For simplicity we assume now that \( A \) is a symmetric matrix of the form

\[
A = \begin{pmatrix}
-A & 0 \\
0 & A
\end{pmatrix}, \quad A^I > 0, \ A^{II} > 0.
\]

Connected with the system (1.1), (1.2) is the reduced equation

\[
A(x)\frac{dv}{dx} + B(x)v = F,
\]

\[
v^I(x) = \alpha^I, \ v^{II}(x) = \beta^{II},
\]

and the eigenvalue problem

\[
A(x)\frac{d\phi}{dx} + B(x)\phi = \lambda\phi,
\]

\[
\phi^I(0) = \phi^{II}(1) = 0.
\]

Here the partition of \( v \) into \( v^I \) and \( v^{II} \) corresponds to the partition of \( A \). The following lemma is essential:

**Lemma 2.1.** Assume that \( \lambda = 0 \) is not an eigenvalue of (2.4), (2.5). Then there are constants \( \varepsilon_0 > 0 \) and \( K > 0 \) such that for all \( \varepsilon \) with \( 0 < \varepsilon \leq \varepsilon_0 \) for the solutions of (1.1), (1.2) the estimate

\[
||y(x)||^2 \leq K \left( \int_0^1 |F(x)|^2 dx + |\alpha|^2 + |\beta|^2 \right)
\]

holds. Here \( ||y|| = \max |y(x)| \).

This lemma can be used to construct the asymptotic expansion. The idea is to subtract from \( y \) special functions such that the difference is a solution of (1.1), (1.2) with smaller right hand sides. As a first step we subtract from \( y \) the solution of the reduced
equation (2.2), (2.3). Then \( w = y - v \) is the solution of

\[
\begin{align*}
(2.8a) \quad Lw &= -\varepsilon(d^2v/dx^2) \\
(2.8b) \quad w^I(0) &= 0, \quad w^I(1) = \beta^I - v^I(1), \\
&\quad w^{II}(1) = 0.
\end{align*}
\]

The next step is to reduce the right hand sides of the boundary conditions. Let \( z(x) \) be the solution of

\[
\begin{align*}
(2.9) \quad \varepsilon(d^2z/dx^2) + \begin{pmatrix} -A^I(1) & 0 \\ 0 & A^{II}(0) \end{pmatrix}(dz/dx) &= 0,
\end{align*}
\]

which is essentially given by

\[
\begin{align*}
&\quad z^I(x) = \exp(\varepsilon^{-1}A^I(1)(x - 1))z^I(1), \\
&\quad z^{II}(x) = \exp(-\varepsilon^{-1}A^{II}(0)x)z^{II}(0).
\end{align*}
\]

It represents the shape of the boundary layer. Then \( u = w - z \) is the solution of

\[
Lu = G, \quad u(0) = u(1) = 0(\exp(-\varepsilon^{-1})),
\]

where

\[
\int_0^1 |G|^2 dx = O(\varepsilon^{1/2}).
\]

Therefore by lemma 2.1

\[
y = v + z + O(\varepsilon^{1/2}).
\]

(One can show that \( O(\varepsilon^{1/2}) \) can be replaced by \( O(\varepsilon) \). This process can be continued. We can thus derive an asymptotic expansion which is built up by solutions of the reduced equation and solutions of

\[
\varepsilon(d^2z/dx^2) + \begin{pmatrix} -A^I(1) & 0 \\ 0 & A^{II}(0) \end{pmatrix}(dz/dx) = G(x, \varepsilon)
\]

where

\[
G(x, \varepsilon) = C_1x^\gamma \exp(-\varepsilon^{-1}x) + C_2x^\nu \exp(\varepsilon^{-1}(x - 1)).
\]

We now consider difference approximations. Let \( h = 1/N, N \) a natural number, be the mesh width, and denote the gridpoints by \( x_v = vh. \) Let \( u_v = u(x_v) \) denote the solution of the difference approximation
What are the properties this approximation must have?

1) It must be uniformly stable, i.e. an estimate corresponding to (2.6) must hold for all sufficiently small $\varepsilon$ and $h$.

2) It must be consistent for the reduced equation.

3) When applied to (2.9) its solution must rapidly decay when going from the boundary into the interior.

The first property seems to be always fulfilled if the third holds. The second never poses any trouble. Let us therefore concentrate on the third property. Consider the differential equation

$$\varepsilon (d^2 y/dx^2) + a(dy/dx) = 0,$$
$$y(0) = \alpha, \quad y(1) = 0.$$  

where $a > 0$ is a constant. Its solution is essentially given by

$$y = \alpha \exp(-(a/\varepsilon)x).$$

Approximate this equation by three different difference approximations which are all well known:

$$\varepsilon D_+ D_- u_\nu + aD_0 u_\nu = 0$$
$$\varepsilon D_+ D_- u_\nu + aD_+ u_\nu = 0$$
$$\varepsilon (\varepsilon + ch)D_+ D_- u_\nu + aD_0 u_\nu = 0, \quad c > a/2$$

with boundary conditions

$$u_0 = \alpha, \quad u_N = 0.$$  

Here

$$hD_+ u_\nu = u_{\nu+1} - u_\nu, \quad hD_- u_\nu = u_\nu - u_{\nu-1}, \quad 2hD_0 u_\nu = u_{\nu+1} - u_{\nu-1},$$

are the usual difference operators. The solutions of the above equations can easily be computed. For $\varepsilon << h$ we get essentially

$$u_\nu = C_1 + (-1)^\nu C_2,$$
(2.13a) \[ u_\nu = \alpha (\tau/(1+\tau))^\nu, \quad \tau = \varepsilon/h, \]

(2.14a) \[ u_\nu = \alpha ((1-\mu)/(1+\mu))^\nu, \quad \mu = .5ah/(\varepsilon+ch). \]

The first approximation has nothing to do with the solution of the differential equation. The second is best because it decays most rapidly from the boundary. The third is all right if \(2c \geq a\) but not too large. Correspondingly the approximation (2.12) does not work for the full equation while the analogs

\[
\begin{align*}
\varepsilon D_+D_-u_\nu & - A \left( x_-\right) D_-u_\nu + .5(B(x_-)(u_\nu+u_{\nu-1})) = F(x_-), \\
\varepsilon D_+D_-u_\nu & + A \left( x_+\right) D_+u_\nu + .5(B(x_+)(u_\nu+u_{\nu+1})) = F(x_+),
\end{align*}
\]

where \(x_- = x_\nu - .5h, \ x_+ = x_\nu + .5h, \)

(2.13b) \[ (\varepsilon+Ch)D_+D_-u_\nu + AD_0u_\nu + Bu_\nu = F(x_\nu), \]

where

\[ C = C^* = .5|A| \text{ constant symmetric matrix}, \]

are convergent. In fact one can develop a theory which is a complete analog with the theory in the analytic case. For example one can prove that there is an asymptotic expansion

\[ u(x_\nu) = y(x_\nu) + \sum_j h \phi_j(x_\nu, \varepsilon) + 0(\varepsilon h/(\varepsilon h)). \]

Both methods are only first order accurate. However, in general (2.13b) is more accurate than (2.14b) and the boundary layers are sharper. The disadvantage is that the matrix \(A\) has to be in partitioned form. This is also a serious drawback if one wants to apply the method to nonlinear equations where the eigenvalues of \(A\) change sign in places not known a priori. No such restrictions are necessary for the method (2.14b). In both cases the accuracy can be improved by Richardson extrapolation.

3. Mesh Refinement. As long as one is not resolving the boundary layers, i.e., is using in the boundary layers a stepsize \(h\) which is of the same order as elsewhere, one must expect at least an error of order \(\varepsilon\) in the whole interval. The reason is that the shape of the solution in the boundary layer has an effect of order \(\varepsilon\) in the whole interval. (There is one exception: scalar equations). Consider now a mesh which is uniform in the interior but much finer near the boundaries. In this case one would expect that in the interior a centered scheme

\[ \varepsilon D_+D_-u_\nu + AD_0u_\nu + Bu_\nu = F_\nu \]
could be used. However, as shown in another paper [Kreiss (1972), cited below], this is not always true. We have not only to avoid that $\lambda = 0$ is an eigenvalue of (2.4), (2.5) but also that $\sigma = \epsilon/h^2$ is not an eigenvalue of

$$A(x) \frac{d\phi}{dx} - B(x) \phi = -4\sigma \phi,$$

$$\phi(0) = \phi(1) = 0.$$

Note that this problem has only the trivial solution for sufficiently large $\sigma$.

REFERENCES

1973


1972

INTRODUCTION

In several areas of numerical analysis there is a current feeling that what is needed is greater insight into the working of existing methods, rather than the development of yet more new methods. This point of view can be convincingly argued in the case of the numerical solution of initial value problems for systems of ordinary differential equations, in the absence of stiffness. However, when stiffness is present, the performance of conventional methods is not very impressive. The basic reasons for this are well-known; conventional methods can have adequate stability to cope with stiff systems only if they are implicit, and stiffness precludes the solution of the resulting implicit difference schemes by direct iteration, necessitating the use of some form of Newton iteration, with the attendant need to compute inverses of matrices. The real computational difficulty with stiff systems is centred round this need repeatedly to compute matrix inverses. This paper will describe two attempts to break out of this situation by considering unconventional classes of methods. One class, that of linear multistep methods with variable matrix coefficients, can be shown theoretically to possess adequate stability, and the methods appear to be computationally sound. However, with methods of this class, it is still necessary to invert a matrix, but only once per step. The second class, that of nonlinear methods, lacks the same level of theoretical backing; computationally, methods of this class appear to possess adequate stability, but there are difficulties, as yet unresolved, in controlling the level of local accuracy. Their advantage is that they require no matrix inversions.

It is in no way claimed that either class contains an ideal
method for stiff systems; but the results are sufficient to persuade the author that further study of unconventional methods is justified.

**LINEAR MULTISTEP METHODS WITH VARIABLE MATRIX COEFFICIENTS**

We shall describe this class only briefly, since a full derivation can be found in Lambert and Sigurdsson (1972A). The general members of the class is

\[
\sum_{j=0}^{k} [a(j,0)I + \sum_{s=1}^{S} a(j,s)h^{s}Q^{s}(n)]y_{n+j} = \\
\sum_{j=0}^{k} h \sum_{s=1}^{S-1} b(j,0)I + \sum_{s=1}^{S-1} b(j,s)h^{s}Q^{s}(n)]y_{n+j}
\]

where the system \( y' = f(x,y) \) has dimension \( N \), the \( a(j,s) \) and \( b(j,s) \) are scalars, \( I \) is the \( N \times N \) unit matrix, and \( Q(n) \) is an \( N \times N \) variable matrix which is arbitrary, apart from the requirements that \( ||Q(n)|| \) be bounded and the (matrix) coefficient of \( y_{n+k} \) be non-singular. If a method of this class has \( b_{ks} = 0 \), \( s = 0,1,\ldots,S-1 \), then the new value for \( y \) can be obtained directly at each step, at the cost of one matrix inversion; we shall call such methods **linearly implicit**. For the purposes of achieving adequate stability, we shall, in practice, choose \( -Q(n) \) to be an approximation to the current Jacobian \( \partial f/\partial y \), but it is important to note that the order of the method, defined in an obvious manner, is independent of the choice made for \( Q(n) \).

A **stabilizing condition**, generalizing that of Lambert (1970A), is postulated as follows. Let \( \zeta \), the principal root, and \( \zeta_p^p \), \( p = 2,3,\ldots,k \) be the spurious roots of the first characteristic polynomial

\[
\rho(\zeta) = \sum_{j=0}^{k} a(j,0)\zeta^{j}.
\]

Let \( y(j,s) = a(j,s) + b(j,s-1), s = 1,2,\ldots,S \). Then, in the case when the spurious roots of \( \rho \) are distinct, the stabilizing condition
requires that
\[ \sum_{j=0}^{k} y(j,s)\zeta_p^j = 0, \quad p = 2,3,\ldots,k; \quad s = 1,2,\ldots,S. \]

There is an obvious modification in the case when the spurious roots are not distinct. For example, if \( \zeta_q = \zeta_{q+1} \), the condition, for \( p = q, q+1 \), becomes
\[ \sum_{j=0}^{k} y(j,s)\zeta_q^j = \sum_{j=0}^{k} jy(j,s)\zeta_q^{j-1} = 0, \quad s = 1,2,\ldots,S. \]

A method is said to be stabilized if it satisfies the stabilizing condition and the strict root condition \( |\zeta_p| < 1, \quad p = 2,3,\ldots,k \).

The stabilizing condition has the following effect. If (1) is applied to the test system \( \dot{\mathbf{y}} = \lambda \mathbf{y}, \quad \Re \lambda < 0 \), and if \( -Q(n) \) is chosen to be \( \lambda I \), then a linear constant coefficient difference equation results. The characteristic polynomial of this difference equation - let us call it the stability polynomial - is an \( O(h) \) perturbation of the first characteristic polynomial \( \rho \). The stabilizing condition guarantees that the spurious roots of the stability polynomial coincide with the spurious roots of \( \rho \), which, if we assume in addition the strict root condition, lie strictly inside the unit circle. Thus, for all \( h \), no instability can arise on account of the spurious roots of the stability polynomial. Moreover, even in the case of a linearly implicit method, the principal root of the stability polynomial is a rational approximation to \( \exp(h\lambda) \), and we can choose coefficients such that this rational approximation has modulus less than unity, hence guaranteeing the A-stability of the method.

As an illustration, consider the stabilized second-order two-step method with \( S = 1 \),

\[ \text{\textsuperscript{†} It is of interest to note that the modified Adams-Bashforth methods proposed by Nørsett (1969A) turn out to be a stabilized linearly implicit sub-class of class (1), provided that the exponential terms in Nørsett's methods are replaced by appropriate rational approximations. The methods of Brunner (1967A) can also be considered as a sub-class (1); they do not, however, satisfy our definition of stabilization.} \]
\[ (1 - \frac{3}{2} \lambda h) y_{n+2} - \left[ (1+\alpha)I + hQ(n) \right] y_{n+1} + \left[ \alpha I + \frac{1}{2} hQ(n) \right] y_n = \frac{1}{2} h \left[ (3-\alpha) y_{n+1} - (1+\alpha) y_n \right], \quad -1 < \alpha < 1. \]

The roots of \( \rho \) are 1, \( \alpha \). On putting \( f = \lambda - y \), \( Q(n) = -\lambda I \), where \( \Re \lambda < 0 \), we obtain the difference equation

\[ (1 - \frac{1}{2} \lambda h) y_{n+2} - \left[ 1+\alpha + \frac{1}{2} (1-\alpha) h \lambda \right] y_{n+1} + \alpha (1+\frac{1}{2} \lambda h) y_n = 0. \]

The spurious root of the stability polynomial is easily seen to be \( \alpha \), verifying that the method is indeed stabilized, while the principal root is seen to be \( (1+\frac{1}{2} h \lambda)/(1-\frac{1}{2} h \lambda) \). The method is clearly \( A \)-stable.

It is convenient to introduce at this point a slightly more restrictive definition of \( A \)-stability. For methods of class (1), just as for linear multistep methods, the test equation \( y' = \lambda y \), \( \lambda \) a complex scalar with \( \Re \lambda < 0 \), is essentially the same as the test equation \( y' = Ay \), \( A \) a matrix with distinct eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_N \), all lying strictly in the left half-plane. Applying method (1) to each of these test equations, and putting \( Q(n) = -\lambda I \), \( Q(n) = -A \), respectively, we obtain difference equations of the form

\[ \sum_{j=0}^{k} P_j(h\lambda) y_{n+j} = 0, \quad \sum_{j=0}^{k} P_j(h\lambda I) y_{n+j} = O \tag{2} \]

respectively, where \( P_j(B) \) is a polynomial in the matrix \( B \). On making the transformation \( y = Hz \), where

\( H^{-1}AH = \lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N) \),

we obtain, on premultiplying the first of (2) by \( H^{-1} \),

\[ \sum_{j=0}^{k} H^{-1} P_j(h\lambda) H z_{n+j} = \sum_{j=0}^{k} P_j(h\lambda) z_{n+j} = 0, \]

which is essentially the same as the second of (2). Our reasons for distinguishing between these two test equations are twofold. Firstly, we can establish stability results for methods of class (1) without making the assumption that the eigenvalues of \( A \) are distinct, and, secondly, the two test equations turn out not to be equivalent, even in the case where the eigenvalues of \( A \) are distinct, for the class of nonlinear methods we discuss later in this paper. Accord-
TWO UNCONVENTIONAL CLASSES OF METHODS

...ingly, we make the following definitions, the first of which is precisely that of Dahlquist (1963A)

Definition 1. A method is $A$-stable if all solutions of the difference equation which results from applying it, with fixed positive $h$, to $y'_n = \lambda y_n$, $\lambda$ a complex scalar with negative real part, tend to zero as $n$ tends to infinity.

Definition 2. A method is $\tilde{A}$-stable if all solutions of the difference equation which results from applying it, with fixed positive $h$, to $y'_n = Ay_n$, $A$ a matrix all of whose eigenvalues lie strictly in the left half-plane, tend to zero as $n$ tends to infinity.

Definition 3. A method of class (1) is $\tilde{A}$-stable if it is $A$-stable for the choice $Q(n) = -A$.

Our results on stability can be summarised in the following theorem (Lambert and Sigurdsson 1972A):

Theorem 1. For any given $S$, there exist both fully implicit and linearly implicit $\tilde{A}$-stable methods of class (1) of order $p$, where $S \leq p \leq 2S$.

Order $2S$ is, in fact, always attainable. Note, however, that Theorem 1 makes no mention of stepnumber. Thus we find that with $S = 2$, for example, we can obtain a fourth-order $\tilde{A}$-stable fully implicit method with stepnumber 3; to get a fourth-order $\tilde{A}$-stable linearly implicit method we have to take stepnumber 4. A selection of $\tilde{A}$-stable methods with $S = 1, 2$ (in view of the need to calculate $S$-th powers of $Q(n)$, higher values of $S$ are rather impracticable) may be found in Lambert and Sigurdsson (1972A).

Although our computational experience with methods of class (1) has been satisfactory, the methods represent more of a theoretical rather than a practical improvement over conventional methods, in the following sense. Solutions to stiff systems are frequently computed by using conventional linear multistep methods which are $A$-stable or $A(\alpha)$-stable. Such methods are necessarily implicit (Dahlquist (1963A), Widlund (1967A)), and some form of Newton iteration is invariably resorted to. In theory, one ought to iterate to convergence in order to be certain of retaining the $A$- or $A(\alpha)$-stability, whereas, in practice, one usually applies the iteration only a small number of times—often only once—per step. (Indeed, in appropriate circumstances the inverse of the Jacobian may be held constant over a number of steps.) Thus, methods of class (1), although they guarantee us adequate stability at the cost of precisely one inversion per step, are, in terms of computational cost, roughly commensurate with conventional linear multistep methods as they are commonly applied in stiff situations. We have...
not yet attempted to develop methods of class (1) into a variable step algorithm with heuristics to determine whether or not to hold $Q(n)$ constant over a number of steps. Our guess is that such an algorithm would be roughly comparable with similar algorithms based on linear multistep methods. We are therefore motivated to make a more radical move away from linear multistep methods, and investigate nonlinear methods. In doing so, we must be prepared to accept a less substantial level of theoretical backing, in the hope of obtaining some practical advantage.

**NONLINEAR METHODS**

Linear multistep methods can be constructed by locally interpolating the data $(y_{n+j}, y'_{n+j})$, $j = 0, 1, \ldots, k$, in a Hermite sense, by a polynomial. It is well-known that explicit linear multistep methods, which correspond to local extrapolation by such polynomials, possess inadequate stability. We shall construct explicit nonlinear methods based on local extrapolation by rational functions, and investigate their stability properties. The technique is the same as that employed by Lambert and Shaw (1965A) although the methods derived there, which were specifically designed to deal with singularities, are unsuitable for stiff systems.

Let the theoretical solution $y(x)$ of the scalar initial value problem $y' = f(x, y)$, $y(a) = \eta$ be represented locally in the interval $[x_n, x_{n+1}]$ by the rational function $I(x) = A/(x+B)$. We impose the requirements

$$y_n = I(x_n), \quad y_{n+1} = I(x_{n+1}), \quad f_n = I'(x_n),$$

and eliminate $A$ and $B$ to obtain the nonlinear method

$$y_{n+1} - y_n = h y'_n f_n / (y_n - h f'_n), \quad (3)$$

a nonlinear analogue of Euler's Rule. We must clearly impose the restriction that $y(x)$ and $y'(x)$ must not vanish simultaneously. If, despite this restriction, $h$ is such that $y_n - h f'_n$ vanishes, we must choose another value for $h$.

Method (3) is component-applicable to the system $y' = \tilde{f}(x, y)$ in the sense that if

$$\tilde{\chi} = [y, y', \ldots, y^{N}]^T, \quad \tilde{\chi}' = [f, f', \ldots, f^{N}]^T,$$

we interpret method (3) to be
Applying (4) to the test equation $y' = \lambda y$, $\text{Re}\lambda < 0$, we find that

$$\frac{y_{n+1}^i}{y_n^i} = \frac{1}{1 - h\lambda}, \quad i = 1, 2, \ldots, N,$$

the $(0, 1)$ Padé approximation to $\exp(h\lambda)$. The method is thus $A$-stable; indeed it is $L$-stable (or stiffly $A$-stable). We cannot, however, prove it $A$-stable in the sense of Definition 2, even in the case when the matrix $A$ of the test equation has distinct eigenvalues. The most we can prove is the following.

**Theorem 1.** Let method (4) be applied to $y' = Ay$, $A$ a real matrix with distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$, and corresponding eigenvectors $z_1, z_2, \ldots, z_N$. Then the resulting (nonlinear) difference system has $N$ independent solutions

$$v_n^i = \frac{1}{1 - h\lambda_i^n} z_i^i \quad i = 1, 2, \ldots, N.$$

Note that, due to the nonlinearity of the difference system, Theorem 1 does not allow us to make any statement about the general solution of the system.

Similar remarks will apply to all of the nonlinear methods we present. Where a stability result can be proved, it will always be $A$- and not $A'$-stability. Since this essentially reduces the analysis to scalar level, we shall henceforth quote methods in scalar form, such as (3), rather in component-applicable form, such as (4).

Analogously to the usual procedure for linear multistep methods, we can associate a nonlinear operator with each nonlinear method. For method (3), the operator is

$$P[y(x);h] := \frac{y(x+h)-y(x) - hy(x)y'(x)/[y(x)-hy'(x)]},$$

where $y(x)$ is an arbitrary function in $C'$ such that $|y(x)| + |y'(x)| \neq 0$, for all $x$. If, for sufficiently differentiable $y(x)$,

$$P[y(x);h] = O(h^{p+1}),$$

we say the method has order $p$, and that the local truncation error $T_n$ is $P[y(x);h]$, where $y(x)$ is now taken to be the theoretical solution of the initial value problem. It is obvious from (3) that independently of $f$, $y_n = 0$ implies $y_{n+1} = 0$, and the method fails
to follow the solution through a zero. This phenomenon can be interpreted in terms of the local truncation error. It is

\[ T_{n+1} = \frac{[yy^{(2)} - 2y^{(1)}y^{(2)}]h^2/2! + [yy^{(3)} - 3y^{(1)}y^{(2)}]h^3/3! + O(h^4)}{y - hy^{(1)}} \bigg|_{x=x_n} \]

(5)

\[ = h^2\left[ \frac{y^{(2)}}{2} - \frac{y^{(1)}}{y} \right]_{x=x_n} + O(h^3). \]

(6)

The second of these two forms (which attempts to set up a principal error function, similar to those defined for Runge-Kutta methods) is a misleading one. It suggests that when the solution passes through a zero, the local truncation error will be infinite. (Recall that we have excluded the case \( y_n = y'_n = 0 \).) It is clear, however, that the local truncation error at such a point will be large, but finite. In practice, the numerical solution stays close to zero, and thus drifts away from the theoretical solution; the behaviour is similar to that of a linear multistep method which is inconsistent, not unstable. That inconsistency is indeed the explanation of the phenomenon is clear from (5). If \( y_n \neq 0 \), then \( T_{n+1} = O(h^{p+1}) \), \( p > 1 \), and the method has order at least one, whereas if \( y_n = 0 \) then, since \( y^{(1)}_n \) cannot also be zero, \( T_{n+1} = O(h) \), and the method has order zero, that is, is locally inconsistent.

Two points, which will apply to all our nonlinear methods, emerge from this discussion. One is that the notion of principal error function, in the style of (6), is always misleading; local truncation errors must always be put in the form (5) before order can be discussed. Secondly, although we can certainly overcome the specific difficulty that (4) has in following a solution through a zero, by constructing methods based on rational functions of higher degree, it will always be the case that the order of our methods may drop (but not always to zero) at a point where the solution and its derivatives satisfy particular relations.

Space does not allow us to give detailed arguments for choosing to develop nonlinear methods of specific structure, based on specific rational interpolants (see Lambert (1973B)). Nor can we give proofs of our stability results here. In the case of one-step methods, these proofs are trivial, since such methods applied to the test equation \( y' = \lambda y \) generate Padé approximations to \( \exp(h\lambda) \). For \( k \)-step nonlinear methods, with \( k > 1 \), the stability analysis is less trivial, and ad hoc techniques must be concocted (Lambert (1973B)). We shall quote below a selection of nonlinear methods. For each we will quote the order \( p \), any known stability results, the form of \( I(x) \) the rational interpolant (where appropriate), and a diagram to
indicate the structure of the method. In this diagram, a star (*) on the lowest line indicates that the method uses the value of $y_{n+j}$, one on the line above indicating the use of a value of $f_{n+j}$, one on the next line above a value of $f_{n+1}^{(1)}$ (:= $y_{n+1}$), and so on.

(For example, in the case of linear multistep methods, the diagrams for the four-step Adams-Bashforth method and for the three-step implicit backward differentiation method would be, respectively,

```plaintext
* * * *
  *   *
* * * *
```

All the methods we quote will be fully explicit. For completeness, we repeat method (3).

Method I

```
* I(x) = A/(x+B); \( y_{n+1} - y_n = h y_n f_n/(y_n - h f_n) \), |y| + |y'| \( \not\in 0 \).
```

\( p \geq 1 \) if \( y \not= 0 \), \( p = 0 \) if \( y = 0 \); L-stable.

Method II

```
* * I(x) = (Ax+B)/(x+C); \( y_{n+2} - y_{n+1} = \frac{h(y_{n+1} - y_n) f_{n+1}}{2(y_{n+1} - y_n) - h f_{n+1}} \).
```

\( p \geq 2 \) if \( y^{(1)} \not= 0 \); \( p = 0 \) if \( y^{(1)} = 0 \), \( y^{(2)} \not= 0 \).

If \( y \), obtained from \( y_0 \) by Trapezoidal Rule, method generates (1,1) Padé approximation; A-stable.

Method III

```
* y_{n+1} - y_n = 2h f_n/(2f_n - h f_{n+1}^{(1)}), |y^{(1)}| + |y^{(2)}| \( \not\in 0 \).
```

\( p \geq 2 \) if \( y^{(1)} \not= 0 \); \( p = 1 \) if \( y^{(1)} = 0 \); A-stable.

Method IV

```
* I(x) = (Ax+B)/(x^2+Cx+D);
```

```
y_{n+2} = \frac{4y_{n+1}^2 (y_{n+1} - y_n) - 4h y_{n+1} y f_{n+1} + h^2 y_n (2f_{n+1} - y_{n+1} f_{n+1})}{4y_{n+1} (y_{n+1} - y_n) - 4h y_{n+1} f_{n+1} + h^2 [2f_{n+1} - (y_{n+1} - 2y_n) f_{n+1}^{(1)}]}
```

A-stable.
Method V

\[ y_{n+1} = y_n + hf_n + \frac{h^2}{2} f_{n}^{(1)} + \frac{h^3}{6} \left( \frac{P f_n^{(2)} + h R n f_n^{(1)}}{P_n + h Q_n - \frac{h^2}{3} R_n} \right), \]

\[ P_n = y_n f_n^{(1)} - 2f^2_n, \quad Q_n = f_n^{(1)} - \frac{1}{3} y_n f_n^{(2)}, \quad R_n = \frac{2}{2} f_n^{(1)} - f_n^{(2)}. \]

\[ |y| + |y^{(1)}| + |y^{(2)}| \neq 0. \]

Method VI (Runge-Kutta type)

\[ y_{n+1} = y_n + hf_n \left( x_n + \frac{1}{2} h, \frac{y_n + \frac{h}{2} f_n^{(1)}}{y_n - \frac{1}{2} hf_n} \right), \]

\[ |y| + |y^{(1)}| \neq 0. \]

Method VII (Runge-Kutta type)

\[ y_{n+1} = \frac{y_n (c_1 k_1 + c_2 k_2)}{y_n - h(d_1 k_1 + d_2 k_2)}, \quad k_1 = f(x_n, y_n), \quad k_2 = f(x_n + ah, y_n + \frac{ah k_1 y_n}{y_n - ah k_1}), \]

\[ |y| + |y^{(1)}| \neq 0. \]

If \( c_1 = 1 - 1/2a, c_2 = 1/2a, d_1 = -d_2 = -c_1, \) then \( p \geq 3 \) in general, \( p \geq 2 \) if \( y^{(1)} = y^{(2)} = 0, \) \( p=1 \) if \( y = y^{(1)} = 0; \) A-stable.

Note that methods VI and VII are both second order in general; the latter has a stronger stability property, but suffers a greater loss of order at a zero of the solution. It is possible to construct a three-stage method similar to VII; it is L-stable and has order three in general, but is inconsistent when \( y = 0. \)

We conclude this section by summarising the theoretical shortcomings of our nonlinear methods. Firstly, the stability analysis is essentially scalar, and we have been unable to devise a technique for analysing stability when the test equation is \( \chi' = A\chi. \) Secondly,
all the methods can suffer a drop in order at special points of the solution. Moreover, the notion of principal local truncation error (or principal error function) is inappropriate for these methods, and consequently analogues of conventional techniques for estimating the local error fail. We comment in the next section on the extent to which these theoretical shortcomings manifest themselves in practice.

COMPUTATIONAL EXPERIENCE

Space does not permit us to present detailed numerical results for a representative selection of the methods discussed in this paper. Variable matrix coefficient methods have been tested on a number of nonlinear stiff systems, and appear to be computationally sound. Applied with fixed steplength, they are roughly equivalent in efficiency to implicit backward differentiation methods with Newton iteration applied just once per step, thus bearing out remarks made earlier in this paper.

The results of numerical tests with nonlinear methods have been interesting and somewhat unusual. In the preceding section we noted that there are two distinct areas of uncertainty, one concerning stability in the case of a system, and the other concerning local accuracy. Our numerical tests indicate clearly that it is the second area and not the first which gives trouble in practice. The conclusions reached from these tests are listed below.

1) Good results can usually be obtained if the steplength is sufficiently small. When the steplength is not small enough, we get solutions which are inaccurate (often widely so) but still stable; they are frequently over-damped.

2) Following a suggestion by W. Enright of Toronto, the methods were tested computationally for $A_-$, as opposed to $A$-stability, by using them to compute, for a range of transformations $P$, the vector $\chi$ where

$$\chi = Pz, \quad z' = P^{-1}APz$$

The accuracy of the computed vector varied with $P$; this is to be expected, since the points at which order loss will be experienced also vary with $P$. We were, however, unable to find any rotation for which the stability was lost.

3) For many problems, while a small step length produces a good global solution, a somewhat larger solution will produce a solution which is globally poor, but nonetheless manages to pick up the asymptotic solution with reasonable accuracy. (Consideration of the form of the integral curves for a typical stiff system shows that such a phenomenon is feasible.) Further increase of step length causes the asymptotic solution to be lost. This effect can be observed in both linear and nonlinear problems.

4) We have not been able to find any satisfactory method for estimating how small the step length has to be to produce the effects mentioned in 3). The maximum allowable step length for a "good" solution does not appear to be directly linked to stiffness, as is the case for linear methods. Thus, of three linear problems with stiffness ratios of 100, 1000, and 4000, the middle one was consistently the hardest to handle. It would appear that an oscillatory solution with modest stiffness ratio is harder to reproduce than a monotonic solution with high stiffness ratio. Consequently, for some problems the step length has to be comparable with that which would be demanded by an explicit linear method, while for others it can be an order larger.

5) Since the performance of these methods is more than usually problem-dependent, it is particularly difficult to pick out the best of the nonlinear methods. We feel that method IV is probably best.

Some of the above effects can be seen in the following examples. Table 1 shows solutions to the following linear stiff system due to Fowler and Warten (1967A):

\[
y' = \begin{pmatrix}
y_1 & y_2
\end{pmatrix}, \quad y_1(0) = 0, \quad 0 \leq x \leq 5.
\]

Both components of the theoretical solution are asymptotic to \(1 \times 10^{-5}\). All values in Table 1 have been multiplied by \(10^4\); each number pair is the vector \([y_1, y_2]^T\).
### TABLE 1

<table>
<thead>
<tr>
<th>x</th>
<th>Method III from x=h</th>
<th>Method IV from x=0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theor</td>
<td>h=.01   h=.025  h=.05</td>
</tr>
<tr>
<td>0.5</td>
<td>6.08 6.09 6.25 4.17</td>
<td>6.07 2.01 11.90</td>
</tr>
<tr>
<td></td>
<td>2.16 2.20 2.55 -0.48</td>
<td>2.15 1.76 2.21</td>
</tr>
<tr>
<td>1.0</td>
<td>6.94 6.97 7.24 4.18</td>
<td>6.94 4.87 3.50</td>
</tr>
<tr>
<td></td>
<td>3.89 3.96 4.51 -0.46</td>
<td>3.89 6.00 1.95</td>
</tr>
<tr>
<td>1.5</td>
<td>7.62 7.65 7.96 4.20</td>
<td>7.62 6.91 -5.57</td>
</tr>
<tr>
<td></td>
<td>5.24 5.32 5.96 -0.44</td>
<td>5.24 8.15 0.15</td>
</tr>
<tr>
<td>2.0</td>
<td>8.14 8.18 8.50 4.21</td>
<td>8.14 8.53 1.76</td>
</tr>
<tr>
<td></td>
<td>6.29 6.37 7.02 -0.42</td>
<td>6.29 8.72 -0.47</td>
</tr>
<tr>
<td>2.5</td>
<td>8.55 8.59 8.89 4.22</td>
<td>8.56 9.26 -1.59</td>
</tr>
<tr>
<td></td>
<td>7.11 7.19 7.80 -0.40</td>
<td>7.11 9.04 0.04</td>
</tr>
<tr>
<td>3.0</td>
<td>8.87 8.91 9.18 4.23</td>
<td>8.87 9.56 0.82</td>
</tr>
<tr>
<td></td>
<td>7.75 7.82 8.38 -0.38</td>
<td>7.75 9.27 0.00</td>
</tr>
<tr>
<td>3.5</td>
<td>9.12 9.15 9.40 4.24</td>
<td>9.12 9.70 -0.58</td>
</tr>
<tr>
<td></td>
<td>8.24 8.31 8.81 -0.36</td>
<td>8.26 9.45 0.00</td>
</tr>
<tr>
<td>4.0</td>
<td>9.32 9.34 9.56 4.25</td>
<td>9.32 9.78 0.20</td>
</tr>
<tr>
<td></td>
<td>8.63 8.69 9.12 -0.34</td>
<td>8.65 9.59 0.00</td>
</tr>
<tr>
<td>4.5</td>
<td>9.47 9.49 9.67 4.27</td>
<td>9.48 9.83 -0.05</td>
</tr>
<tr>
<td></td>
<td>8.93 8.99 9.35 -0.31</td>
<td>8.96 9.68 0.00</td>
</tr>
<tr>
<td>5.0</td>
<td>9.58 9.61 9.76 4.28</td>
<td>9.59 9.87 -0.06</td>
</tr>
<tr>
<td></td>
<td>9.17 9.22 9.52 -0.29</td>
<td>9.19 9.76 0.00</td>
</tr>
</tbody>
</table>

Note that method III starting from x=h (with theoretical solution there as starting value) has order 2 everywhere. Method IV starting from x = 0 has order 1 for the first step, and order 3 subsequently.

The following nonlinear stiff system is taken from Liniger and Willoughby (IBM Research Report RC-1970, 1967.)

\[
y_1' = 0.01 - (1+y_1 + 1000)(y_1 + 1)[0.01 + y_1 + y_2], \quad y_1(0) = 0
\]

\[
y_2' = 0.01 - (1+y_2^2)(0.01 + y_1 + y_2), \quad y_2(0) = 0.
\]
Numerical results by Method V are shown in Table 2. In order to avoid the order drop at $x=0$, the method was started from $x = 1$.

**TABLE 2**

<table>
<thead>
<tr>
<th>x</th>
<th>Theoretical</th>
<th>Method V from $x=1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$h=0.01$</td>
</tr>
<tr>
<td>10</td>
<td>-1.0975</td>
<td>-1.0423</td>
</tr>
<tr>
<td></td>
<td>0.0978</td>
<td>0.11540</td>
</tr>
<tr>
<td>100</td>
<td>-0.99164</td>
<td>-0.99898</td>
</tr>
<tr>
<td></td>
<td>0.98334</td>
<td>0.99395</td>
</tr>
</tbody>
</table>

**CONNECTION BETWEEN THE TWO CLASSES**

In view of the widely differing theoretical background and computational characteristics of the two classes of methods discussed in this paper, it is somewhat surprising to discover that the two sets of methods intersect. Consider, for example, the nonlinear method I,

$$y_{n+1} - y_n = h y_n f_n / (y_n - h f_n),$$

which, in its component-applicable form (4), may be written as

$$[I - h \text{ diag}(\frac{i f_n}{y_n})](y_{n+1} - y_n) = h f_n.$$  \hspace{1cm} (7)

On the other hand, the general linearly implicit one-step variable coefficient method with $S=1$ is

$$[I + a h Q(n)](y_{n+1} - y_n) = h f_n.$$ \hspace{1cm} (8)

Its order, 1, is independent of the choice made for $Q(n)$; it is A-stable $a > \frac{1}{2}$. It is clear that (7) is the special case of (8) for the choice

$$-Q(n) = \text{ diag}(\frac{i f_n}{y_n})$$

which we can interpret as an extremely crude approximation to the Jacobian. Note that the order of (7) is indeed 1 except when $y=0$; in this case our previous assumption that $Q(n)$ is bounded is invalid.

As a second example, the nonlinear method II can be written, after some manipulation, in the form
which is seen, in its component-applicable form, to be the special
case of the two-step second order linearly implicit variable
coefficient method

\[
[I + \frac{1}{2}h Q(n)]y_{n+2} - hQ(n)y_{n+1} + [-I + \frac{1}{2}h Q(n)]y_n = 2hf_{n+1},
\]

obtained by making the choice

\[
Q(n) = \text{diag} \left[ \begin{array}{c}
\frac{i f_{n+1} - (y_{n+1} - y_n)/h}{y_{n+1} - y_n} \\
\frac{i y_{n+1} - y_n}{y_{n+1} - y_n}
\end{array} \right] \tag{9}
\]

This somewhat strange-looking expression turns out to be meaning­
ful. The best approximation to \( f \) that can be achieved with the
data \( y_n, y_{n+1}, f_{n+1} \) is

\[
2(y_{n+1} - y_n)/h - f_{n+1} = f_n + O(h^2).
\]

Substituting this approximation for \( f_n \), we find

\[
\frac{\partial f}{\partial y} \approx \frac{f_{n+1} - f_n}{y_{n+1} - y_n} = \frac{f_{n+1} - (y_{n+1} - y_n)/h}{y_{n+1} - y_n} + O(h),
\]

and we can once more interpret the right side of (9) as an approxi­
mation to the Jacobian.

CONCLUSIONS

We do not claim to have found an ideal method for handling
stiff systems. The variable matrix coefficient methods appear to
be theoretically and computationally sound, but in practice are
not much different from implicit A-stable linear multistep methods
as they are commonly applied. The nonlinear methods appear, in
practice, always to have sufficient stability, but there is a
difficulty, as yet unresolved, in controlling the level of local
accuracy. In going from polynomials to rationals, we have traded
local accuracy for stability, and have possibly gone too far in
this direction. The connection between the two classes, described
above, suggests that the variable matrix coefficient methods may,
in practice, retain their stability properties even if the approxi­
mation to the Jacobian is a crude - and cheap - one; applied in
such a manner, they may be more competitive.

Acknowledgement. The numerical tests reported in this paper were programmed by Mr Stephen Driver and Mr Iain Watson of the University of Dundee.
INTRODUCTION

The practitioner is usually unaware of the nature of stiff systems and the associated numerical integration difficulties. Even arbitrary application of stiff methods is deemed significant enough in many areas of application to be suitable for publication. On the other hand, typical solution characteristics and the requirements on their elucidation may not be fully appreciated by the numerical analyst.

In what follows, the nature of the stiff problem as it occurs in practice is examined. The commonly used pseudo-steady-state approximation (pssa) to the model is shown to be a powerful technique when applied properly. An estimate of the domain of validity and the accuracy is derived from singular perturbation techniques for the general stiff system not necessarily in singularly perturbed form. Several examples demonstrate the viability of the approach.

STIFF SYSTEMS IN PRACTICE

Characteristic of the solution behavior of the stiff variables arising in many disciplines, and the type with which we shall deal, is a rapid change initially, not necessarily a simple decay. After this induction period, or boundary layer, transients settle to a slowly varying state, often a decay to low magnitudes in the quantity. The non-stiff variables may be essentially constant in the boundary layer, but this depends on the degree of stiffness. Although the stiff variable is often known a priori from physical
considerations, the degree of stiffness can vary considerably even among similar problems. Thus the variables of interest are usually the non-stiff ones outside of the boundary-layer. The nature of the application will dictate what aspect of the solution is to be explored, whether it be the steady-state distribution, relative maxima/minima in certain variables, the type of approach to one of several possible steady states, or bounds on the transient. Note also that rarely is the model an adequate enough representation of the phenomenon to require a numerical solution with better than moderate accuracy.

Chemical kinetics provides numerous illustrations of these points. The stiff variable corresponds to intermediate species that quickly react or are present in low concentrations. The (1973) paper by Edelson makes it clear that most kinetic study is model-fitting rather than fundamental chemistry, with the resulting highly approximate equations. They observe the general lack of knowledge of stiff methods, but the recent interest of some kineticians with the Gear (1971B) routine. This was also pointed out by Gelinas (1972) who examines the solution of a large, highly nonlinear kinetic set representing the development of photochemical smog with both the Gear method and the pssa. Schneider, et al. (1972) expend considerable effort in applying singular perturbation techniques to an autocatalysis scheme. The difficulties encountered with treating the turning point via their analytic approach could easily be avoided by employing suitable numerical techniques for this mildly stiff problem. Creighton (1) evangelizes stiff methods to the kinetician and discusses the stability problems of the very stiff chain reaction of hydrogen bromide. Branched free radical mechanisms as in the pyrolysis of propane (2) provide unusual cases where there may be stiff intermediates appearing and disappearing as the reaction proceeds.

Enzyme kinetics contains many further examples (3; Otten and Duysens, 1973). This area provides a simple illustration of why the pssa is employed in practice for reasons other than the removal of stiffness in the numerical solution. Consider the basic enzyme reaction


Systems Having Widely Varying Time Constants

\[
E + S \xrightarrow{k_1} (ES) \xrightarrow{k_2} E + P \xrightarrow{k_{-1}} E
\]

where the concentrations are

- **E** = enzyme
- **S** = substrate
- **(ES)** = enzyme-substrate complex
- **P** = product

The experimentalist may want to fit data on the rate of product formation to the simple first-order form

\[
\frac{dP}{dt} = k_2(ES) \quad , \quad P(0) = 0
\]

It is, however, very difficult to measure the complex concentration (ES). From

\[
\frac{d(ES)}{dt} = k_1SE - (k_{-1} + k_2)(ES) \quad , \quad (ES)(0) = 0
\]

the approximation \( d(ES)/dt = 0 \) is rationalized here by typical small enzyme concentrations \( E \) and in \( (ES) \), giving

\[
\frac{SE}{(ES)} = \frac{k_{-1} + k_2}{k_1} = K_M
\]

where \( K_M \) is the so-called Michaelis-Menten kinetic constant. Since the total amount of enzyme in free or complexed form is constant,

\[
E(0) = E + (ES)
\]

so that

\[
(ES) = \frac{E(0)}{K_M + S}
\]

Thus the rate of product formation is now

\[
\frac{dP}{dt} = \frac{k_2E(0)}{K_M + S}
\]

in terms of the easily measured substrate concentration. The only penalty is the introduction of one additional parameter \( K_M \); a more exact analysis would significantly complicate the rate expression and data gathering. The same pssa eliminates stiffness from the numerical solution of the full system equations and should be expected to be as successful in this context as the quite different context of Michaelis-Menten kinetics. Lim (1973) investigates the validity of the approximation for varying enzyme concentrations.
Polymerization kinetics with short-lived radicals are additional sources of highly stiff systems. Sena and Kershenbaum (4), in their study of the oxidation of isobutane, rather arbitrarily used several stiff methods to determine their relative merits. They conclude that the trapezoidal rule is very useful for their low accuracy requirements. Bisenberger and Capinpin (1972) examined free radical polymerization and the special numerical problems for "dead end", nonisothermal, and indred polymerization.

Chemical reactor dynamics reflect the stiffness of the reactions they contain. Although this may seem obvious, Schneider, et al. (1973) and Ray (1968) demonstrated it for the continuously stirred tank reactor. Actually, Jefferson and Smith (1973) and Ray found that the degree of stiffness could definitely change, the reactor equations being less stiff in both studies than the contained kinetic equations.

Various topics in transport phenomenon are naturally stiff when there is a significant difference in the transfer rate of coupled processes. For batch distillation, Distefano (1968) mentions the stiffness resulting from the time constant of the reboiler being very large compared with the time constants of the plates. Descloux (1970) uses the Gear routine on the heat equation reduced to a set of ordinary differential equations by the Galerkin method. Flow past an object produces a physical boundary layer over which some transfer rates can greatly vary over short distances as in Reiby (1972) and Ackerberg and Phillips (1973). The boundary layer region is of primary interest in Patterson and Creswell (1971), who examine heat and mass transfer with reaction in a catalyst pellet using collocation techniques within an "effective reaction zone" boundary layer approximation. It may occur that the domain of integration is so large as to make the problem stiff even if all the eigenvalues are of moderate absolute magnitude. For example, some iterative schemes involved with solving the two point boundary value problem of a tubular reactor with axial dispersion (Corbo and Lapidus, 1972) require continuation of the solution down the entire length of the reactor for each iteration, although process variables may reach essentially constant values near the entrance.

Many other areas have similar bases for the occurrence of stiffness. In circuit theory, the small time constants of certain circuit elements provide more examples (Russo, 1971; Richard et al., (1972)). In the case of process control, the fast acting controller

---

presents the problem (Jamshidi, 1972; Yackel, 1973); (5, 6) provide an account of examples in ballistics, nuclear reactor dynamics, and other fields.

AN APPROXIMATION TO THE MODEL THAT ELIMINATES STIFFNESS

The Pseudo Steady-State Approximation

Effectively describing a phenomenon in the simplest terms is a modeling problem; however, the conceptual simplifications invoked allow a rationale behind committing approximations to the model. As long as the solution characteristics are preserved, several percentages of error can usually be tolerated. The approximations may permit fewer parameter specifications, an analytic or easier numerical solution as shown by the enzyme kinetics example presented earlier. One of the most common type of model reductions, the pseudo steady-state approximation (pssa) involves the stiff system

\[ \dot{x} = f(x, y, \varepsilon, \mu) \]
\[ \varepsilon \dot{y} = g(x, y, \varepsilon, \mu) \]

where \( x \) and \( y \) are \( n \) and \( m \) nonstiff and stiff vectors, respectively; \( \varepsilon \) and \( \mu \) are small parameters. The pssa consists of setting \( \varepsilon = 0 \) (Jeffreson and Smith, 1973; Aris, 1970). This may be distinguished from letting \( \varepsilon = 0 \) (5), which can cause a more substantial error (Ray, 1969). It may be that the regular perturbing quantity \( \mu \) set equal to zero usefully reduces the dimensionality (Jamshidi, 1972; Brayton, et al., 1966), but the theoretical basis for this is not as finely developed as the pssa will be shown to be. The most important qualifications in the pssa, the region of applicability and the accuracy, will be examined in detail.

Singular Perturbation Theory for General Stiff Systems

There is an equivalence between stiff and singularly perturbed equations. This feature was exploited by Miranker (7) and Aiken and


Lapidus (8) to obtain numerical methods based on singular perturbation theory for the general stiff system not necessarily in perturbed form. Here we are considering

\[ x = f(x, y) \]  
\[ y = g(x, y) \]

where \( y \) contains the stiff components. Equations [2] can be identified with [1] if we define a local \( \varepsilon \) such that \( g = \varepsilon w \). The solution may then be found to the first order in the inner \((x_0, x_1, y_0, y_1)\) and outer \((x_0, x_1, y_0, y_1)\) terms as

\[ x \sim x_0 + \varepsilon x_1 + x_0 + \varepsilon x_1 \]
\[ y \sim y_0 + \varepsilon y_1 + y_0 + \varepsilon y_1 \]

These terms were numerically calculated in Aiken and Lapidus without the need for identification of the perturbing quantity \( \varepsilon \) and thus are applicable to the general form [2] of the stiff system. The zeroth-order outer solution,

\[ x_0 = f(x_0, y_0) \]
\[ 0 = w(x_0, y_0) \]

corresponds to the pssa. This has been recognized for systems originally in singular perturbed form (Yackel and Kokotovic, 1973; Bowen, et al., 1963); however, for stiff systems not in perturbed form, the corresponding basis has heretofore been missing, although the pssa has occasionally been applied when the stiff characteristics had been recognized (Otten and Duysens, 1973; Gelinias, 1972).

The applicability of the pssa for the general stiff system may be elucidated within a singular perturbation context. It should be expected invalid where the inner solution dominates the outer solution -- in the boundary layer; application within this region will destroy the remainder of the solution. The stiffer the problem, the smaller is \( \varepsilon \), and thus the better will be the accuracy of retaining only the zeroth-order terms; also the boundary layer


will be shorter, giving an increased region of applicability.

The problem of a priori finding the boundary layer thickness (here to predict the time after which the pssa is valid) is quite familiar to numerical analysts, since most integration methods face the difficulties of picking up the boundary-layer behavior for reasons of accuracy while taking large time steps and maintaining stability. Attempts at beginning the integration with small (how small?) steps to be increased when it is supposed that the boundary layer has been passed (what criterion?) is not recommended (Richard, et al., 1972). Aiken and Lapidus suggest a means of estimating this region based on their zeroth-order inner term for the stiff variables becoming negligible \( \{Y_0/Y_0(0) < 10^{-3}\} \):

\[
\frac{\partial Y}{\partial t} = Y_0(0)e^{-\frac{w_y}{Y_0}},
\]

where \( w_y \) is the Jacobian of [2b] evaluated at \( t = 0 \) and must have eigenvalues with negative real parts. Note that if \( w_i \) is linear in \( y_j \) \( (i,j = 1, \ldots, m) \) -- and this seems to be true for a majority of stiff systems arising in practice -- a very good estimate of the boundary layer can be made, as \( w \) will not vary much with \( y \) while \( x \) is relatively constant. If \( w_y \) is not linear in \( y_j \), the eigenvalues of \( w \) must be checked at the end of the initial boundary layer length estimate to assure they are not appreciably smaller in absolute magnitude.

The accuracy of the pssa outside the boundary layer is indicated by the size of the first-order outer terms. This sort of procedure has been proposed by Ponzo and Wax (1972) for equations in the singular perturbed form, but their conditions are not feasible for computation. The numerical scheme of Aiken and Lapidus will yield this information for the general stiff system.

Additional Remarks on the Solution of the PSSA Equations

The numerical solution of [4] is particularly simplified if an actual decomposition is affected, i.e., if \( w \) can be solved for \( Y_0 \) and substituted into \( f \). This may often be done for linear and certain nonlinear forms of \( w \) in \( Y_0 \) if the system is small or large and sparsely interconnected. When decomposition is not possible, \( w \) may be evaluated for \( Y_0 \) at the beginning of each step and this value used throughout the interval, since the variation in \( y \) will be small. Simultaneous integration of the differential algebraic system as in Hachtel, et al. (1971A) or Gear (1971C) is not recommended as the instability problems may persist (4).

Note that the validity of the pssa may be extended to cover the entire domain by proper choice of the initial conditions in a laboratory or numerical experiment, if this is feasible. Thus, if
Yo(0) satisfies
\[ w(x_0(0), Y_0(0)) = 0, \]
then
\[ Y_0(0) = 0, \]
and the boundary layer contribution to the solution is always small. Higher accuracy may then be achieved by employing suitable stiff techniques without concern for accuracy loss from the boundary layer.

Application of the PSSA with Boundary Layer and Accuracy Estimates

The viability of the pssa with the above boundary layer and accuracy estimates was examined on three applications representative of naturally occurring stiff systems. The non-stiff Runge-Kutta fourth-order (RK4) IBM Scientific Subroutine Package routine was used with the reduced model. The entire expansion [3] was also used throughout the interval to check the accuracy of dropping the first order terms. The solution was compared in each example with that obtained from the Gear (1971B) package. All computations were performed on the IBM 360/91 at the Princeton University Computer Center. Previous notation of \( \tilde{z} \) for the non-stiff and \( y \) for the stiff variables is continued. Note in each example \( w_i \) is linear in \( y_j \) \((i,j = 1, \ldots, m)\) and thus \( y \) can be eliminated from \( f \).

First Example: Proton Transfer in Hydrogen-Hydrogen Bond. Carbonell (9) has obtained the rate constants for the transition of a proton to various energy levels

\[ \begin{align*}
  x_1 & \to y \to x_2 \\
  k_1 & \to k_3 \to k_4
\end{align*} \]

within a hydrogen-hydrogen bound from quantum mechanical considerations:

\[ \begin{align*}
  \frac{dx_1}{dt} &= -k_1x_1 + k_2y, & x_1(0) &= 0 \\
  \frac{dx_2}{dt} &= -k_4x_2 + k_3y, & x_2(0) &= 1 \\
  \frac{dy}{dt} &= k_1x_1 + k_4x_2 - (k_2 + k_3)y, & y(0) &= 0
\end{align*} \]

Fig. 1. Proton transfer.
Fig. 2. Fluidized-bed catalyst temperature.
with

\[ k_1 = 8.4303270 \times 10^{-10} \]
\[ k_2 = 2.9002673 \times 10^{11} \]
\[ k_3 = 2.4603642 \times 10^{10} \]
\[ k_4 = 8.7600580 \times 10^{-6} \]

This extremely stiff linear three-dimensional example is of interest since the eigenvalues cannot be computed by any means due to the ill-conditioned Jacobian. The quickly reacting intermediate level obviously contains the stiff variable.

The estimated boundary layer length was \( 2.2 \times 10^{-10} \). Step sizes of \( 5 \times 10^{-6} \) were taken over the interval up to \( t = 8 \times 10^5 \) to produce Fig. 1. The achieved accuracy of 0.01% over the entire solution was limited only by the step size and RK4 used with the pssa, as the first order terms were found to always be completely negligible.

Second Example. Transient Behavior of a Catalytic Fluidized Bed. Luss and Amundson (1968) have proposed a simplified model for the dynamics of a catalytic fluidized bed:

\[
\frac{dx}{dt} = 1.30(y_2 - x) + 2.13 \times 10^6 k y_1 , \quad x(0) = 761 ,
\]
\[
\frac{dy_1}{dt} = 1.88 \times 10^3 [y_3 - y_1(1 + k)] , \quad y_1(0) = 0.0 ,
\]
\[
\frac{dy_2}{dt} = 1752 - 269y_2 + 267x , \quad y_2(0) = 600 ,
\]
\[
\frac{dy_3}{dt} = 0.1 + 320y_1 - 321y_3 , \quad y_3(0) = 0.1 ,
\]

where \( k = 0.006 \exp(20.7 - 15000/x) \) and \( x, y_1, y_2, y_3 \) represent the temperature (°R) and partial pressure (atm.) of the catalyst and interstitial fluid respectively. The problem is highly nonlinear in \( x \) and not in singular perturbed form. With the above selection of parameters three possible steady-states were reported. Severe numerical stability difficulties were encountered using RK4, leading Luss and Amundson to state that the use of the assumption \( x = y_2 \) and \( y_1 = y_3 \) "might be the only feasible way to carry out the transient computations". Actually, this causes the solution to proceed to the wrong steady-state.

It is known a priori from physical reasoning that the cata-
Fig. 3. Ozone decomposition with first-order inner-outer solution.
Fig. 4. Ozone decomposition with zeroth-order outer solution (PSSA).
lyst temperature $x$, the major variable of interest, will dictate the solution behavior, as its heat capacity is the main obstacle to change. Thus the other variables will quickly jump from their initial conditions to follow that of the particle. The transient of $x$ is shown in Fig. 2. The boundary layer was estimated to extend to $t = 5.3$. The pssa then gave better than 0.1% accuracy for $x$ with step sizes of 100 throughout the interval up to $t = 1400$. The first order terms were found to be very small contributions for all variables over the entire transient. The computation time required was only 0.4 second compared with a prohibitive number of hours using RK4 without the pssa.

**Third Example.** The Thermal Decomposition of Ozone. This was used by Bowen, et al. (1963) as an important system in singular perturbed form with which to derive an analytical solution approximation via matched asymptotic expansions. The accepted kinetic steps involved are described by

$$\frac{dx}{dt} = -x - xy + \varepsilon ky,$$

$$\varepsilon \frac{dy}{dt} = x - xy - \varepsilon ky,$$

where $x$ is the reduced ozone concentration and $y$ is the reduced oxygen concentration; $\varepsilon = 98$ and $k = 3$.

The authors expended considerable effort in obtaining the inner, outer, and "far out" solutions which, when combined, appear as in Fig. 3 for $y$, where the exact solution is also plotted. The relative error is seen to be in excess of 100% most of the time. Equation [5] gave a boundary layer estimate of 0.068. As can be seen from Fig. 3, this marks the end of an interval of rapid change; it also agrees with the point at which Bowen, et al. switched to their outer solution. After that point, a step size no larger than 0.5 could be used for 5% accuracy with the pssa, illustrated in Fig. 4. Smaller step sizes would not appreciably increase the accuracy. This is attributable to the first order outer terms of [3]. The execution time to $t = 240$ was 1.12 sec., only about 20 times faster than RK4. This only moderately stiff problem illustrates a point also indicated by Zeman (10): if a system may be integrated at all in a reasonable time, by standard techniques, it is not safe to apply singular perturbation techniques, particularly the pssa.

---

A STIFF SYSTEM PACKAGE BASED ON THE IMPLICIT MIDPOINT METHOD WITH SMOOTHING AND EXTRAPOLATION

Bengt Lindberg
Department of Computer Science
The Royal Institute of Technology
Stockholm, Sweden

INTRODUCTION

A procedure for the numerical solution of initial value problems for systems of stiff differential equations

\[ y' = f(t, y); \quad y(0) = y_0 \quad y \in \mathbb{R}^S \]

is presented.

The underlying method is the implicit midpoint method with smoothing and one extrapolation. Theoretical justification for the basic method is given in [Dahlquist (1963A), Lindberg (1971A), Dahlquist's paper] and by Lindberg in a number of recent reports from the Department of Information Processing, Computer Science, The Royal Institute of Technology, Stockholm, Sweden. A description of the procedure together with an ALGOL program and some numerical results can be found in Lindberg (1973A), IMPEX 2 a procedure for solution of systems of stiff differential equations, TRITA-NA-7303 in the series of reports mentioned above.

BASIC FORMULAS

To simplify the notation we regard henceforth the autonomous system

\[ (1) \quad y' = f(y); \quad y(0) = y_0 \quad y \in \mathbb{R}^S. \]
Denote by \( y(t,h) \) an approximation to the solution of (1) computed by the implicit midpoint method

\[
y(t_0,h) = y_0
\]

(2) \( y(t+h,h) - y(t,h) = h \cdot f([y(t+h,h)+y(t,h)])/2) \\
t = t_0, t_0+h, \ldots
\]

with the stepsize \( h \).

Theorem 1. The solution of (2) can be written in the form

\[
y(t_n,h) = y(t_n) + h^2 d_1(t_n) + h^4 d_2(t_n) + w_n(h)
\]

where

\[
y'(t) = f(y)
\]

(4) \( d_1'(t) - J(t)d_1(t) = \frac{1}{12} y'''(t) - \frac{1}{8} J'(t)y'(t) \).

(5) \( d_2'(t) - J(t)d_2(t) = -\frac{y'(t)}{120} + \frac{d''(t)}{12} - \frac{1}{2} f''(y(t)) (d_1(t) + y''(t))/8)^2 \\
- \frac{1}{384} (y''(t) - J(t)y''(t)) \\
- \frac{1}{8} (d'''(t) - J(t)d''(t) - y''(t))/12). \)

(6) \( w_{n+1}(h) - w_n(h) = h^2 J_{n+\frac{1}{2}} (w_{n+1}(h) + w_n(h))/2 + O(h^7) \).

Here \( J(t) = \{\partial f/\partial y\} y(t) \) and if \( w_n(h) = O(h^4) \) then

\[
J_{n+\frac{1}{2}} = \int_0^1 f'(y(t_{n+\frac{1}{2}}) + \theta (\frac{1}{2} (y_n + y_{n+1}) - y(t_{n+\frac{1}{2}}))) d\theta.
\]

This theorem can be proved with the same technique as in Dahlquist's paper. See that paper also regarding the use of ordo notation in connection with stiff systems.

As in Dahlquist's paper we shall say that a solution to the system of differential equations

\[
dx/dt = g(t,x)
\]

is smooth at time \( t_1 \) with respect to the time-constant \( \tau \) if for some prescribed constants \( \alpha, \beta \) we have the
following inequalities

\( \| x^{(v)}(t_1) \| \leq \alpha \cdot \tau^{-v} \quad v = 0, 1, \ldots p. \)

Briefly speaking we shall refer to such a solution as a smooth function. Assume in the sequel that \( y(*) \), \( d_1(*) \) and \( d_2(*) \) are smooth functions.

Contrary to the results of [Gragg (1965), Stetter (1965)] this theorem is valid without any restrictions on the product of the stepsize \( h \) and the Lipschitz constant \( L = \sup_{y \in D} \| \partial f / \partial y \| \), and thus is a firm basis for construction of extrapolation methods for stiff systems. For stiff systems the term \( w(h) \) in the expansion (3) contains an oscillating component, which although small when the starting vector for (2) is properly chosen [Dahlquist's paper] may cause difficulties for an extrapolation method. The amplitude of the oscillation may be decreased by the following smoothing formula: given \( y(t_k-h, h) \), \( y(t_k, h) \), \( y(t_k+h, h) \), compute

\[
\hat{y}(t_k, h) = \frac{y(t_k-h, h) + 2y(t_k, h) + y(t_k+h, h)}{4}.
\]

**Lemma 1.** Let \( \hat{y}(t_k, h) \) be defined by (8) then

\[
\hat{y}(t_k, h) = y(t_k) + [d_1(t_k)+y''(t_k)/4]h^2
+ [d_2(t_k)+d_1(t_k)/4+y''(t_k)/48]h^4
+ \acute{\hat{w}}_k(h) + O(h^6)
\]

where

\[
\hat{w}_k = \frac{w_{k-1}(h)+2w_k(h)+w_{k+1}(h)}{4}.
\]

**Proof:** By Taylor expansion around \( t = t_k \) we directly obtain this result.

An investigation of the effect of the smoothing (8) is given in [Lindberg (1971A)].

Note that the smoothed values have an expansion in even powers only of the stepsize \( h \).

**Theorem 2.** Let \( \acute{\hat{y}}(t_k, h) \) be defined by

\[
\acute{\hat{y}}(t_k, h) = \hat{y}(t_k, h/2) + [\hat{y}(t_k, h/2)-\hat{y}(t_k, h)]/3
\]
then

\[
\overline{y}(t_k,h) = y(t_k) - (d_2(t_k) + d_1(t_k)/4 + y''(t_k)/48)h^4/4 \\
+ \overline{w}_k(h) + O(h^6)
\]

where

\[
\overline{w}_k(h) = \hat{w}_k(h/2) + [\hat{w}_k(h/2) - \hat{w}_k(h)]/3.
\]

**Proof:** Follows directly from Lemma 1.

By the extrapolation we eliminate the \( h^2 \)-term of the expansion (9) but as \( \hat{w}_k(h) \) does not depend regularly on \( h \) \( \overline{w}_k(h) \) will not be significantly less than \( \hat{w}_k(h) \) or \( \hat{w}_k(h/2) \). Hence the necessity of the smoothing to decrease \( w_n(h) \) to \( \overline{w}_k(h) \).

Difficulties with the oscillating term \( w_n(h) \) for stiff systems forced us to use at most one extrapolation even after smoothing. Recent investigations [Dahlquist's paper] that reveal how the amplitude of the oscillations depend on the choice of starting-vector initially and after change of stepsize indicate how the influence of the oscillations may be decreased so extrapolation methods of higher order maybe can be constructed.

**ALGORITHM**

The procedure integrates from a given initial point to a prescribed final time \( T \). In doing so it repeats four basic computational steps:

a) proceed one large step

b) estimate the error

c) accept or reject the last step, choose a new stepsize

d) prepare for the next large step.

The initial step is a little different, it consists of four ordinary large steps followed by b, c and d.
In the following sections these computational steps are described.

THE LARGE STEP

Two independent solutions \( y(t, h) \) and \( y(t, h/2) \) are computed by the implicit midpoint rule.

\[
\begin{align*}
(14) \quad & y(t+h, h) - y(t, h) = h \cdot f([y(t+h, h) + y(t, h)]/2), \\
& t = t_0, t_0 + h, \ldots \\
(15) \quad & y(t+h/2, h/2) - y(t, h/2) = \frac{h}{2} \cdot f([y(t+h/2, h/2) + y(t, h/2)]/2), \\
& t = t_0, t_0 + h/2, \ldots
\end{align*}
\]

At the points \( t_k = t_0 + k \cdot h; \ k = 1, 2, \ldots \) passive smoothing is performed.

\[
\hat{y}(t_k, h) = [y(t_k, h, h) + 2 \cdot y(t_k, h) + y(t_k - h, h)]/4
\]

\[
\hat{y}(t_k, h/2) = [y(t_k, h/2, h/2) + 2 \cdot y(t_k, h/2) + y(t_k - h/2, h/2)]/4.
\]

From the smoothed values passive extrapolation is done

\[
\bar{y}(t_k, h) = \hat{y}(t_k, h/2) + [\hat{y}(t_k, h/2) - \hat{y}(t_k, h)]/3.
\]

Passive here means that the result is not used in the forthcoming computations, but only for printing.

The non-linear systems of equations are solved by a quasi-Newton method. Put \( z = y(t+h, h) \), \( y = y(t, h) \) and

\[
F(z) = z - y - h \cdot f([z+y]/2).
\]

The system (14) is then written \( F(z) = 0 \) and is solved iteratively by

\[
H(z^{i+1} - z^i) = - F(z^i)
\]

where \( H \) is an approximation to the matrix

\[
\{\partial F/\partial z\} = [I - h/2\{\partial f(\eta)/\partial \eta\}_{\eta=(z+y)/2}].
\]
The same $H$ is used for several time steps, as long as the rate of convergence is satisfactory. The rate of convergence is estimated by

$$\rho = \frac{\|z^i - z^{i-1}\|}{\|z^i - z^{i-2}\|} \quad i = 1, 2, \ldots$$

If $\rho \leq 0.2$ the rate of convergence is regarded as satisfactory. If $\rho > 0.2$ the iteration matrix $H$ is reevaluated.

The system (15) is solved accordingly.

The calculations of $y(t, h/2)$ and $y(t, h)$ are carried out in parallel; first two steps with stepsize $h/2$, then one with stepsize $h$.

To get starting values for the iterative solution of $y(t_{k-1} + h/2, h/2)$ and $y(t_k, h/2)$ quadratic extrapolation using the values $y(t_{k-3}, h/2)$, $y(t_{k-2}, h/2)$ and $y(t_{k-1}, h/2)$ is performed. As starting value for the solution of $y(t_k, h)$ we use $y(t_k, h/2) + (y(t_{k-1}, h) - y(t_{k-1}, h/2))$.

**Representation of Information**

Define

$$c(t_k, h) = \frac{(\hat{y}(t_k, h) - \hat{y}(t_k, h/2))}{3} - \frac{(y(t_k + h, h/2) - 2y(t_k, h/2) + y(t_k - h, h/2))}{16}.$$ 

Note that only the values of $y(t_v, h/2)$ for $t_v = (2v) \cdot h/2$, i.e. only the values corresponding to an even number of steps are used. From (3) and (9) we see that $c(t_k, h)$ is an approximation to $h^2 d_1(t_k)/4$ with an error $O(h^4)$. Thus $(c(t_k, h) + c(t_{k-1}, h))/2$ approximates $h^2 d_1(t_k - h/2)/4$ with an error $O(h^4)$ [cf. Lindberg (1972A)].
Thus
\[
\bar{y}(t_k - h/2, h) = \text{def} 2\left[\dot{y}(t_k - h/2, h/2) - (c(t_k, h) + c(t_k - h, h)/2\right]
- \left[\bar{y}(t_k, h) + \bar{y}(t_k - h, h)\right]/2
\]
is an approximation to \(y(t_k - h/2)\) which has an error \(O(h^4)\).

For printing, error estimates and change of step-size backward differences \((\nabla^i_{h/2} y, i = 0, 1, 2, 3, 4)\) of \(\bar{y}(, )\) are saved. Due to the interpolation above information from an interval of length \(2h\) is sufficient; if the interpolation would not be performed values from an interval of length \(4h\) would be necessary.

For error estimates and change of stepsize backward differences \((\nabla^i_y, i = 0, 1, 2, 3, 4)\) of \(c(t_k, h)\) are saved.

Introduce \(y_n = y(t_n, h)\), \(c_n = c(t_n, h)\) and
\[
Y_n = [y_n, \nabla_{h/2} y_n, \nabla^2_{h/2} y_n, \nabla^3_{h/2} y_n, \nabla^4_{h/2} y_n]
\]
\[
c_n = [c_n, \nabla_{h} c_n, \nabla^2_{h} c_n, \nabla^3_{h} c_n, \nabla^4_{h} c_n].
\]
The updating of old information when \(y_{n+1/2}\) and \(y_{n+1}\) has been computed is then given by two successive applications of
\[
Y_{\alpha+1/2} = Y_{\alpha}
\]
\[
\begin{bmatrix}
0 & -1 & -1 & -1 & -1 \\
0 & 0 & -1 & -1 & -1 \\
0 & 0 & 0 & -1 & -1 \\
0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
\[
+ [y_{\alpha+1/2}, y_{\alpha+1/2}, y_{\alpha+1/2}, y_{\alpha+1/2}, y_{\alpha+1/2}]
\]
for \(\alpha = n\) and \(\alpha = n + 1/2\).
When $c_{n+1}$ is given we update according to

$$
C_{n+1} = C_n + [c_{n+1} \quad c_{n+1} \quad c_{n+1} \quad c_{n+1} \quad c_{n+1}]
$$

ERROR ESTIMATES

The error estimates are based on formula (12) of Theorem 2, i.e.

(16) $y(t_k, h) = y(t_k) - (d_2(t_k) + d_1(t_k) + d_1(t_k) + d_2(t_k)/4 + y''(t_k)/48) \cdot h^4/4$

$+ \tilde{w}_k(h) + O(h^6)$

and the differential equations that define $d_1(\cdot)$ and $d_2(\cdot)$.

Assume in the sequel that $w_k(h), \tilde{w}_k(h/2)$ are negligible compared with $(d_2(t_k) + d_1(t_k) + d_1(t_k) + d_2(t_k)/4 + y''(t_k)/48) \cdot h^4/4$.

SMOOTHING ERROR

Define

(17) $\tilde{y}(t_k, h) = y(t_k, h/2) + (y(t_k, h/2) - y(t_k, h))/3$

then

(18) $\tilde{y}(t_k, h) = y(t_k) - d_2(t_k) h^4/4 + \tilde{w}_k(h) + O(h^6)$

where

(19) $\tilde{w}_k(h) = w_k(h/2) + (w_k(h/2) - w_k(h))/3$.

From (16) and (18) we see that the error due to the smoothing is

(20) $s(t_k) = -(d''(t_k) + y''(t_k)/12) h^4/16$. 
As the smoothing is done passively this contribution to the global error does not propagate, but depends on the values of \(d_1''(\cdot)\) and \(y''(\cdot)\) at the actual point. A crude estimate of the smoothing error can easily be obtained as

\[
(21) \quad \bar{s}(t_k) = \nabla^2 [\hat{y}(t,h/2) - \hat{y}(t,h)]/12.
\]

From Lemma 1 we directly get

\[
\bar{s}(t_k) = (d_1''(t_k) + y'''(t_k)/4) \cdot h^4/16 + O(h^6).
\]

**LOCAL ERROR**

Define the local error as

\[
(22) \quad \ell(t_k) = (d_2(t_k+h) - d_2(t_k)) \cdot h^4/4
\]

i.e. as the increase (or decrease) of that part of the global error that corresponds to \(d_2(\cdot)\). From formula (5) of Theorem 1 we get for linear systems with constant coefficients

\[
(23) \quad d_2'(t) - J(t)d_2(t) = -y''(t)/120 + d_1'''(t)/12.
\]

For non-linear systems (23) is approximately valid (cf. the section on global errors). By Taylor expansion we directly get from (23)

\[
\ell(t_k) = (d_2(t_k+h) - d_2(t_k))h^4/4
\]

\[
= [h(-y''(t_k)/120 + d_1'''(t_k)/12) + h^2/2 \cdot d_2''(\zeta_k)] \cdot h^4/4
\]

\[
t_k \leq \zeta_k \leq t_{k+1}.
\]

For smooth solutions \(y(\cdot), \ d_1(\cdot), \ d_2(\cdot)\) the term \(h^2/2 \cdot d_2''(\zeta_k)\) is small compared to the first term within the bracket [cf. Dahlquist's paper]. Thus

\[
(24) \quad \ell(t_k) = (-y''(t_k)/120 + d_1'''(t_k)/12)h^5 + O(h^6).
\]

An estimate of this local error by computed quantities is

\[
\ell*(t) = \nabla^3_h [\hat{y}(t,h/2) - \hat{y}(t,h)]/36
\]
where \( \nabla_h z(t) = z(t) - z(t-h) \). From Lemma 1 we get
\[
\varepsilon^*(t_k) = \left( \frac{d^4(t_k)}{48} + y'(t_k)/192 \right) h^5 + O(h^7).
\]

Studies on linear systems with constant coefficients indicate that \( \| \varepsilon(t_k) \| < \| \varepsilon^*(t_k) \| \) along smooth trajectories. In fact one gets \( \| \varepsilon(t_k) \| < 3 \| \varepsilon^*(t_k) \| \). Hence
\[
(25) \quad \overline{\varepsilon}(t_k) = \varepsilon^*(t_k)/3
\]
is a better estimate of the local error.

We do not aim at strict estimates of the local error, but rather want realistic estimates that can be used for choice of stepsize. Misjudgements of the local error at some points along the trajectory are not regarded as serious failures. Because, as will be shown in the next section, estimates of the global error will be computed from the local estimates, and in the long run an underestimate during short intervals of time does not seriously damage the global estimate, and will probably be compensated for by overestimates during other parts of the trajectory.

GLOBAL ERROR

The global error consists of the smoothing error (20) and the term \(-d_2(t_k) h^4/4\). The following discussion on how to determine \( d_2(t_k) \) leans heavily on the assumption that \( y(\cdot), d_1(\cdot) \) and \( d_2(\cdot) \) are smooth functions, see (7) and [Dahlquist's paper]. For given \( t_1 \) the quantities \( \alpha \) and \( \tau \) are constants but they may (and indeed do) vary along the trajectory. Assume, for ease of notation, that a piecewise constant scaling of time has been done, so that we can put \( \tau = 1 \), i.e. a smooth function \( x(t) \) satisfies \( \| x^{(v)}(t_1) \| \leq \alpha \)
\( \forall v = 0, 1, \ldots, p \). Assume further that \( \| d^2f/dy^2 \| \leq c_1 \) and \( \| J^{(v)}(t) \| \leq c_2, \ v = 1, 2, \ldots, p \) where \( c_1 \) and \( c_2 \) are small compared with \( \| J(t) \| \).

From formula (5) of Theorem 1 we then approximately get

From formula (5) of Theorem 1 we then approximately get
\( d_2'(t) - J(t)d_2(t) = -y''(t)/120 + d_1''(t)/12. \)

Hence, as \( \| J'(t) \| \) is small compared to \( \| J(t) \| \) we approximately get with

\[ g(t) = -y''(t)/120 + d_1''(t)/12 \]

\[ d_2(t+h) = e^{J(t)}h d_2(t) + e^{J(t)} h \int_0^h e^{-J(t)s} g(t+s) ds. \]

By the mean-value theorem for integrals we get

\[ d_2(t+h) = e^{J(t)}h d_2(t) + e^{J(t)}(h-\zeta) g(t+\zeta) \cdot h \quad 0 \leq \zeta \leq h. \]

Note that

\[ e^{J(t)}(h-\zeta) = I + (h-\zeta)e^{J\eta}J(t) \quad 0 \leq \eta \leq h. \]

Thus

\[ d_2(t+h) = e^{J(t)}h d_2(t) + hg(t+\zeta) + h(h-\zeta)e^{J(t)} J(t) g(t+\zeta). \]

Note that \( y''' = Jy'y \) and \( d_1' = Jd_1 + y''/12 - J'y'/8 \). By successive differentiation of these relations we get

\[ Jy'' = y''' - J''y'y - 4J''y'' - 6J'''y''' - 4J''y''' \]

and

\[ Jd_1'' = d_1''' - J'''d_1 - 3J''d_1' - 3J'd_1'' - y'''/12 + (J''y'+3J''y''+3J'''y'''+J'y''')/8. \]

Hence \( \| Jy'' \| \leq (1+15c_2)\alpha \) and \( \| Jd_1'' \| \leq (13/12+8c_2)\alpha. \)

Consequently \( \| Jg(t+\zeta) \| \leq (0.1+c_2)\alpha. \)

Note also that [Dahlquist (1959A) p. 14]

\[ e^{J(t)} \eta \leq e^{\mu[J(t)]} \cdot \eta \leq 1 \]

for stable systems. Thus

\[ d_2(t+h) = e^{J(t)}h d_2(t) + hg(t+\zeta) + O(h^2) \]

and as \( \| g'(t+\zeta) \| \leq \alpha \) we get

\[ d_2(t+h) = e^{J(t)}h d_2(t) + hg(t) + O(h^2). \]

Hence
If we want a strict bound the best value of $M$ is

$$M = \|e^{J(t)h}\| \leq e^{|J(t)|h}.$$  

If we are satisfied with an approximate bound we can do better.

Note that

$$d_1' - Jd_1 = y'''/12 - J'y'/8.$$

Thus $d_1(\cdot)$ is governed by the same kind of variational equation as $d_2(\cdot)$, and the forcing functions for both equations contain only smooth functions.

Disregard the term $J'y'/8$ and use the same argument as for the equation for $d_2(\cdot)$, then we get

$$e^{J(t)h}d_1(t) = d_1(t+h) - hy'''(t+h)/12 + O(h^2).$$

Let

$$e^{J(t)h}d_1(t) = K \cdot d_1(t)$$

and assume that

$$e^{J(t)h}d_2(t) \leq K \cdot d_2(t).$$

I.e. we assume that $d_2(\cdot)$ decreases more rapidly, or approximately with the same rate, as $d_1(\cdot)$. That this assumption is valid for linear systems with constant coefficients having only real negative eigenvalues can easily be verified.

From (29) and (30) $K$ can be estimated as

$$\overline{K} = \left\{ \frac{\| \hat{y}(t+h,h/2) - \hat{y}(t+h,h)\|}{3 + \| \hat{y}(t+h,h)\|} \right\}$$

and note that

$$\|e^{J(t)h}d_1(t)\| = K \cdot d_1(t).$$

Put $H(t) = \| -d(t)h^4/4 \|$ and note that $\ell(t) = g(t) \cdot h^5/4$ (cf. (24) and (27)). Thus

$$H(t+h) \leq \overline{K} \cdot H(t) + \| \ell(t) \|$$
is an approximation to one part of the global error. The total global error (cf. (16) and (21)) can then be estimated by

\[ E(t) = H(t) + \| \tilde{s}(t) \|. \]

**ACCEPTING A STEP, COMPUTING A NEW STEPSIZE**

The choice of stepsize is governed by the estimate \( L_n = \| \tilde{y}(t_n) \| \) of the local error. The strategy is to keep \( L_n < \varepsilon_{\text{loc}} \) where \( \varepsilon_{\text{loc}} \) is the maximally allowed local error. As the amount of work needed to change the stepsize is comparatively large only fairly large changes are allowed. Further the stepsize is allowed to increase only if \( L_n < L_{n-1} \).

Denote the old stepsize \( h \) and the new stepsize \( h_{\text{new}} \) then the table below gives the appropriate actions.

<table>
<thead>
<tr>
<th>( \frac{L_n}{\varepsilon_{\text{loc}}} )</th>
<th>( h_{\text{new}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1/5120</td>
<td>( \left( \frac{\varepsilon_{\text{loc}}}{5L_n} \right)^{1/5} \cdot h )</td>
</tr>
<tr>
<td>[1/5120,1/80]</td>
<td>2h</td>
</tr>
<tr>
<td>[1/80,1/2]</td>
<td>h</td>
</tr>
<tr>
<td>[1/2,1]</td>
<td>h/2</td>
</tr>
<tr>
<td>&gt;1</td>
<td>reject last step and put ( h_{\text{new}} = h/2 )</td>
</tr>
</tbody>
</table>

**ALGORITHM FOR CHANGE OF STEPSIZE**

Let the old stepsizes be \( h, h/2 \) and the new stepsizes \( H, H/2 \). The basic idea is to extrapolate from the solutions \( y(t,h), y(t,h/2) \) not to \( h = 0 \) but to \( H \) and \( H/2 \) to get starting values for smooth numerical solutions \( y(t,H) \) and \( y(t,H/2) \). The aim is
to keep the irregular errors (oscillations) so small that one gets fourth order accuracy by extrapolation from \( y(t,H) \) and \( y(t,H/2) \).

A rigorous treatment of this matter is given in [Dahlquist's paper]. There exists, in fact, a fairly large set of starting values that give smooth numerical solutions \( y(t,H) \), \( y(t,H/2) \). With the information at hand the best ones (i.e. the starting values whose distance from the exact solution are smallest) are given by (31), (32), (33) and (34) below.

Compute a minimal \( C_n \) approximately consistent with a smooth solution \( d_1(t) \) of (4) [cf. Dahlquist's paper] as

\[
C_n := C_n - \alpha \cdot T
\]

where \( T \) is the first column of the matrix \( T \) and

\[
\alpha = c_n T_1 / T_1 T_1
\]

With the information at hand the best ones (i.e. the starting values whose distance from the exact solution are smallest) are given by (31), (32), (33) and (34) below.

\[C_n := C_n - \alpha \cdot T\]

where \( T \) is the first column of the matrix \( T \) and

\[
\alpha = c_n T_1 / T_1 T_1\]

Change from backward differences for discretization steps \( h/2 \) and \( h \) to discretization steps \( H/2 \) and \( H \) for the four first components of \( Y \) and \( C \) accordingly (the fifth component is discarded).

\[
[Y_n]_{H/2} = [Y_n]_{h/2} \cdot M(\rho)
\]

\[
[C_n]_H = \rho^2 \cdot [C_n]_h \cdot M(\rho)
\]

where \( \rho = H/h \)

\[
M(\rho) =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & \rho & 0 & 0 & 0 \\
0 & (\rho - \rho^2)/2 & \rho & 0 & 0 \\
0 & \rho/3 - \rho^2/2 + \rho^3/6 & \rho^2 - 3 & \rho^3 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
Compute starting values for smooth numerical solutions \( y(t, H) \) and \( y(t, H/2) \) according to

\[
\begin{align*}
y_s(t_n, H) &= y_n + 4c_n \\
y_s(t_n, H/2) &= y_n + c_n.
\end{align*}
\]
CONSTRUCTION OF A FAMILY OF SECOND ORDER, A-STABLE K-STEP FORMULAS DEPENDING ON THE MAXIMUM NUMBER, 2K-2, OF PARAMETERS *

Werner Liniger
IBM Research Center
Yorktown Heights, N. Y.

Thierry Gagnebin
University of Neuchâtel

Abstract:

In the present paper we give, for arbitrarily large K, an explicit construction of a (2K-2)-parameter family of second-order, A-stable K-step formulas. The parameters in these formulas may be used, for example, for exponential fitting, to achieve strong damping of the stiff components, or to produce A-stable solutions of order p > 2 as weighted averages of second-order solutions.

* The results of this paper were obtained during the principal author’s stay as an invited professor at the Swiss Federal Institute of Technology, Lausanne, and the University of Neuchâtel. The work was supported in part by the Swiss National Research Foundation under Grant No. 2.642.72
A linear K-step formula,

\[ \sum_{j=0}^{K} \alpha_j x_{n+j} - h \sum_{j=0}^{K} \beta_j x_{n+j} = 0, \quad (1) \]

can be A-stable [Dahlquist (1963A)] only if its order of accuracy, \( p \), satisfies \( p \geq 2 \). There exists an A-stable formula with \( p = 2 \) and \( K = 1 \): the Trapezoidal Rule [Dahlquist (1963A)]. But it is useful to consider as well formulas with \( p = 2 \) and \( K \geq 2 \) containing free parameters. Such parameters may be used, for example, for exponential fitting [Liniger and Willoughby (1970A)] or to achieve strong damping of the stiff components, or to produce A-stable methods of order \( p > 2 \) by the method of averaging [Liniger and Odeh (1972A)].

The discussion of A-stability of formulas with relatively large step number \( K \) is rather difficult if the formula is parametrized by some of its coefficients \( \alpha_j \) and/or \( \beta_j \), or by parameters on which these coefficients depend linearly. The domain of A-stability in parameter space is defined by nonlinear inequalities in several dimensions \((2K-2)\) if \( p = 2 \). In the present paper we define another parametrization - more natural for discussing A-stability - and we give a method for constructing explicitly K-step formulas of order \( p = 2 \) which are A-stable in an open domain of the parameter space of maximum dimensionality, \( 2K-2 \).

If we apply (1) to the test equation for A-stability, \( x = \lambda x \), we are led to studying the characteristic equation

\[ \chi(w, q) = \rho(w) - q \sigma(w) = 0, \]

where

\[ \rho(w) = \sum_{j=0}^{K} \alpha_j w^j, \quad \sigma(w) = \sum_{j=0}^{K} \beta_j w^j, \quad (2) \]

are the well-known polynomials associated with (1) and where \( q = \lambda h \). Let \( \sigma_i, \ i=1, \ldots, K, \) denote the roots of \( \sigma(w) \). If \( |\sigma_i| < 1, \ i=1, \ldots, K, \) and if \( \|w\| = 1 \Rightarrow \text{Re} \left[ \rho(w)/\sigma(w) \right] \geq 0 \), then the formula (1) is A-stable [Liniger (1968A)]. Via the idempotent * mapping \( z = z(w) = (w + 1)/(w - 1) \) and the transformations \( r(z) = (z - 1)^K \rho(w(z)) \),

* i.e. the inverse map is \( w = w(z) = (z + 1)/(z - 1) \).
\[ s(z) = (z-l)^K \sigma(w(z)) \], the above sufficient conditions for A-stability are transformed into

i) \( \Re s_i < 0 \), \( i=1, \ldots, K \), and

ii) \( \Re z = 0 \Rightarrow \Re \left[ r(z)/s(z) \right] \geq 0 \),

where \( s_i, i=1, \ldots, K \), are the roots of \( s(z) \) and where the rational function \( r(z)/s(z) \) is sometimes referred to as the canonical fraction [Genin (1973A)]. We write

\[ r(z) = \sum_{j=0}^{K} a_j z^j, \quad s(z) = \sum_{j=0}^{K} b_j z^j. \]  

(3)

Condition i) may be discussed by the Routh-Hurwitz theory [Gantmacher (1959A)]. Consider the Routh tableau of Fig. 1,

\[
\begin{array}{cccc}
\gamma_K &=& b_K & \quad b_{K-2} & \quad \cdots \quad \cdots \\
\gamma_{K-1} &=& b_{K-1} & \quad b_{K-3} & \quad \cdots \quad \cdots \\
\gamma_{K-2} & & \cdots & \quad \cdots & \quad \cdots \\
& & \cdots & \quad \cdots & \quad \cdots \\
\gamma_0 &=& b_0 &=& 1 \\
\end{array}
\]

Fig. 1

formed of the coefficients of \( s(z) \), where the \( (K+1) \) entries of the first column are denoted by \( \gamma_0, \ldots, \gamma_K \), and where the formula (1) is normalized by setting \( \gamma_0 = b_0 = 1 \). The Routh tableau is constructed from the top down, with the entries of its first two rows, \( b_0 = 1, b_1, \ldots, b_K \), as starting values, by the recurrence relation

\[ E = (BC - AD) / B, \]  

(4)
where the relative positions of the entries A, B, C, D, and E are shown in Fig. 2

If the formula (1) is normalized as mentioned above, then the Routh criterion states that i) \( \gamma_i > 0 \), i = 1, ..., K. Unless K is very small, the inequalities \( \gamma_i > 0 \) represent very complicated nonlinear constraints on the coefficients \( b_i \). Therefore it is more natural to parametrize \( s(z) \) by \( \gamma_1, \ldots, \gamma_K \), rather than by \( b_1, \ldots, b_K \). This is possible as the construction of the Routh tableau can be reversed: One can write (4) in the form

\[
C = \frac{(AD + BE)}{B}
\]

and thereby construct the tableau from the bottom up with the entries of its first column, \( \gamma_0 = 1, \gamma_1, \ldots, \gamma_K \), as starting values. Then i) is satisfied if \( \gamma_i > 0 \), i = 1, ..., K. Note that both terms of (5) have positive sign so that, in starting from positive \( \gamma_i \), all entries in the tableau become positive, in particular the \( b_j \). This confirms the well-known fact that \( \text{Re } s_i < 0 \), i = 1, ..., K, implies that all the \( b_j \) are of the same sign.

The choice of the \( \gamma_i \), i = 1, ..., K, ties down K degrees of freedom in constructing the formula (1) in which altogether there are \( 2K+1 \) degrees of freedom. In the following we shall assume that a particular choice has been made of the \( \gamma_i \), i = 1, ..., K, and that therefore the \( b_i \), i = 1, ..., K, are given.

To discuss ii), let \( z = iy \), y real and \( \eta = y^2 \). Let a polynomial
FAMILY OF SECOND ORDER A-STABLE K-STEP FORMULAS

\[ R(\eta) = \sum_{i=0}^{K} e_i \eta^i \] be defined by

\[ R(\eta) = \text{Re} \left[ r(\eta) s(-\eta) \right]. \] (6)

Then ii) \( \Leftrightarrow R(\eta) \geq 0, \eta \geq 0 \). The coefficients of \( R(\eta) \) are found to be

\[
\begin{align*}
    e_0 &= a_0 b_0, \\
    e_1 &= -a_2 b_0 + a_1 b_1 - a_0 b_2, \\
    &\vdots \\
    e_{K-1} &= -a_{K-2} b_K + a_{K-1} b_{K-1} - a_K b_{K-2}, \\
    e_K &= a_K b_K.
\end{align*}
\] (7)

If we use the convention [Widlund (1967A)]: \( a_i = b_i = 0, \ i < 0, \ i > K \), then these \( (K+1) \) relations can be written as

\[ e_i = \sum_{j \geq 0} (-1)^{i+j} a_j b_{2i-j}, \ i = 0, \ldots, K. \]

For ii) to be satisfied it is then clearly sufficient that \( e_i \geq 0, \ i = 0, \ldots, K \). If there would not be any accuracy requirements on the formula (1) we could, after having specified arbitrary positive values for the \( K \) parameters \( \gamma_1, \ldots, \gamma_K \), complete the parametrization by selecting arbitrary non-negative values for the \( K+1 \) parameters \( e_i, \ i = 0, \ldots, K \), and then solve the \( K+1 \) equations (7) for the \( K+1 \) unknowns \( a_0, \ldots, a_K \). The resulting formula would then be A-stable. To make the desired formula accurate to order \( p=2 \), as well as A-stable, the procedure may be altered as indicated hereafter.

The linear multistep formula (1) is accurate to order \( p, p \geq 1 \), if and only if [Widlund (1967A)]
\[ a_i = 2 \sum_{j=0} b_{2j+1+i} / (2j+1), \quad i=K, K-1, \ldots, K-p, \quad (8) \]

where we have used again the convention mentioned above. In particular, for \( p=2 \), we have

\[
\begin{align*}
a_K &= 0, \\
a_{K-1} &= 2b_K, \\
a_{K-2} &= 2b_{K-1}.
\end{align*}
\]

(9)

With these constraints the system (7) becomes overdetermined and the \( e_i \) must satisfy three constraints. Substitution in (7) from (9) gives \( e_K = e_{K-1} = 0 \) and (7) may be replaced by the reduced system

\[
e_i = \sum_{j=0}^{K-3} (-1)^{i+j} a_j b_{2i-j} + c_i, \quad (10)
\]

where

\[
c_i = 2(-1)^{i+K-2}(b_{K-1} b_{2i-K+2} - b_{2i-K+1}) , \quad i=0, \ldots, K-2.
\]

(11)

The system (10) consists of \((K-1)\) equations for the \((K-2)\) unknowns \(a_0, \ldots, a_{K-3}\).

To find the last constraint between the \( e_i, i=0, \ldots, K-2 \), let \( b_{ij} = (-1)^{i+j} b_{2i-j} \), \( i=0, \ldots, K-2, j=0, \ldots, K-3 \), and let \( B = (b_{ij}, c_i) \) be the square matrix of order \( K-1 \) whose last column is formed of the \( c_i \). Let \( e = (e_0, e_1, \ldots, e_{K-2})^T \) and \( a = (a_0, a_1, \ldots, a_{K-3}, a)^T \) be \((K-1)\)-column vectors, where \( a \) is an auxiliary quantity. Then (10) can be written as \( Ba = e \), subject to the constraint \( a=1 \). Then \( a = B^{-1}e \) and, if the row vector \( u = (u_0, u_1, \ldots, u_{K-2}) \) represents the last row of \( B^{-1} \), then the constraint \( a=1 \)
takes the form

\[ u \cdot e = 1. \]  

(12)

By definition of \( u \) we have \( u \cdot B = (0, 0, \ldots, 1) \), or

\[ B^T u^T = (0, 0, \ldots, 1)^T \]  

(13)

which permits the calculation of \( u \).

The vector \( u \) is independent of \( K \) in the following sense: Let \( u \) be associated with \( \gamma_0, \gamma_1, \ldots, \gamma_K \) and \( u^+ \) with \( \gamma_0^+, \gamma_1^+, \ldots, \gamma_K^+, \gamma_{K+1}^+ \) (note that the first \( K \) \( \gamma \)-values determining \( u^+ \) are those determining \( u \)). Then \( u^+ = (u, u_{K-1}^-) \). We give the proof for \( K=3 \rightarrow K^+=4 \), which shows the general idea. The complete proof will be given elsewhere. The Routh tableaux for \( K=3 \) and \( K^+=4 \) are represented in Fig. 3.

\[
\begin{array}{ccc}
\gamma_3^+ = b_3^+ & b_1^+ & b_0^+ = b_0 \\
\gamma_2^+ = b_2^+ & b_0^+ & \\
\gamma_1^+ & b_0^+ & \\
\gamma_0^+ = b_0^+ & & \\
\end{array}
\]

\[
\begin{array}{ccc}
\gamma_4^+ = b_4^+ & b_2^+ & b_0^+ = b_0 \\
\gamma_3^+ = b_3^+ & b_0^+ & \\
\gamma_2^+ & b_0^+ & \\
\gamma_1^+ & b_0^+ & \\
\gamma_0^+ = b_0^+ & & \\
\end{array}
\]

\[ b_1 = (\gamma_3^+ / \gamma_2^+) b_0 + \gamma_1 \]

\[ c_0 = 0 \]

\[ c_1 = 2(\gamma_2 b_1 - \gamma_3 b_0) = 2 \gamma_1 \gamma_2 \]

\[ b_2^+ = (\gamma_4^+ / \gamma_3^+) b_1 + \gamma_2 \]

\[ c_0^+ = 0 \]

\[ c_1^+ = -2 \gamma_3 b_0 \]

\[ c_2^+ = 2(\gamma_3 b_2^+ - \gamma_4 b_1^+) = 2 \gamma_2 \gamma_3 \]

Fig. 3
Denote by $L_i$, $i=0, \ldots, K-2$, the left sides of the equations in (13) which, for $K=3$ and $K=4$, are

$$L_0 = b_0 u_0 - \gamma_2 u_1 = 0,$$

$$L_1 = 2 \gamma_1 \gamma_2 u_1 / c_1 = 1,$$

$$L_0^+ = b_0 u_0^+ - \left( \gamma_4 / \gamma_3 \right) b_1 + \gamma_2 u_1^+ + \gamma_3 u_2^+ = 0,$$

$$L_1^+ = b_1 u_1^+ - \gamma_3 u_2^+ = 0,$$

$$L_2^+ = -2 \gamma_3 b_0 u_1^+ + \gamma_3 \gamma_3 u_2^+ / c_1^+ + c_2^+ = 1,$$

respectively. Let $u = (u_0, u_1)$ satisfy $L_0 = 0$, $L_1 = 1$. Then we claim that $u^+ = (u_0^+, u_1^+, u_2^+)$ satisfies (13) for $K=4$ provided that $u_2^+$ is by definition such that $L_1^+ = 0$. In fact, by this definition and by the assumption on $u$ we have

$$L_0^+ = L_0^+ + \left( \gamma_4 / \gamma_3 \right) L_1^+ = L_0 = 0,$$

$$L_2^+ = L_2^+ + 2 \gamma_2 L_1^+ = L_1 = 1,$$

(note that $c_1 = 2(\gamma_2 b_1 - \gamma_3 b_0) = 2 \gamma_1 \gamma_2$).

For arbitrary $K$ the last element $u_{K-2}$ of $u$ is determined from the previous ones by the recurrence relation

$$\sum_{i=\lfloor k/2 \rfloor}^{k} (-1)^{k+i+1} b_{2i-k+1} u_i = 0, \quad k=1, 2, \ldots, K-2,$$

$$k \leq \lfloor k/2 \rfloor$$
which, as in the special case \( k=2 \), represents the second-to-last of equations (13). Here \([r]\) denotes the integer part of the real number \( r \) and, for each \( k \), the b's are those associated with \( k+2 \).

If one expresses the \( u_i \) directly by the \( \gamma_i \) one finds

\[
\begin{align*}
    u_0 &= (2 \gamma_0 / \gamma_1)^{-1}, \\
    u_1 &= (2 \gamma_1 / \gamma_2)^{-1}, \\
    u_2 &= (2 \gamma_1 / \gamma_2)^{-1} \left( \left( \gamma_0 / \gamma_2 \right) + \left( \gamma_1 / \gamma_3 \right) \right), \\
    u_3 &= (2 \gamma_1 / \gamma_2)^{-1} \left[ \left( \gamma_0 / \gamma_2 \right)^2 + \left( \gamma_1 / \gamma_3 \right)^2 + \left( \gamma_1 \gamma_2 / \gamma_3 \gamma_4 \right) \right].
\end{align*}
\]

In general, the \( u_i \) are rational expressions in the \( \gamma_i \)'s. For \( 0 \leq i \leq 4 \) it has been verified that in these expressions all terms are of positive sign. Therefore, if condition i) is satisfied and thus \( \gamma_i > 0, \ i=1, \ldots, K \), these \( u \)'s are all positive and one might conjecture that this is true in general.

The relation (12) can be written as

\[
    u_0 e_0 = 1 - \sum_{j=1}^{K-2} u_j e_j
\]

and, if arbitrary non-negative values are chosen for the \( K-2 \) parameters \( e_j, \ j=1, \ldots, K-2 \), the system (10) for determining the \( a \)'s is consistent if \( e_0 \) is determined from (16). To guarantee A-stability we must have \( e_0 \geq 0 \). But, since i)\( u_0 > 0 \) by equation (15), this inequality is clearly satisfied if the \( e_j, \ j=1, \ldots, K-2 \), are sufficiently small, regardless of whether or not the conjecture about the sign of the \( u \)'s is true. We have thus proved the following

**Theorem:**

Let \( K \geq 3 \). In the \((2K-2)\)-dimensional space of the parameters \( \gamma_1, \ldots, \gamma_K; e_1, \ldots, e_{K-2} \) of a \( K \)-step formula of order \( p=2 \) there exists a set \( D \) of maximum dimension such that all
formulas associated with points in $D$ are A-stable. $D$ can be found by the following construction.

**Construction:**

1) Let $\gamma_i > 0$, $i=1, \ldots, K$, normalize: $\gamma_0 = b_0^0 = 1$. Compute the entries $b_1, \ldots, b_K$ in the first two rows of the Routh tableau by the recurrence relation (5).

2) Let $U_0 = (2 \gamma_0 \gamma_0)^{-1}$. For $k=1, \ldots, K-2$ compute $u_k$ recursively by (14), using the $b$'s associated with $k+2$.

3) Select $e_i$, $i=1, \ldots, K-2$, $e_i \geq 0$, such that $\sum_{j=1}^{K-2} u_j e_j \leq 1$. From (16) determine the non-negative value of $e_0$.

4) Solve any $(K-2)$ of the $(K-1)$ equations (10) with respect to $a_j$, $j=0, \ldots, K-3$.

5) Let $a_{K-2}$, $a_{K-1}$, and $a_K$ take the values indicated by (9).

The theorem is valid for $K=2$ as well; in this case, the parameter space is $\left(\gamma_1, \gamma_2\right)$, $D$ is the open first quadrant, and steps 2) through 4) of the construction do not apply.

The constraint on $e_i$, $i=1, \ldots, K-2$, stated in step 3) defines the simplex shown schematically in Fig. 4:

![Figure 4](image-url)
The domain $D$ is the Cartesian product of this simplex and of the first "quadrant" of $(\gamma_1, \ldots, \gamma_K)$. The original polynomials defined by (2) can be recovered by the inverse transformations

$$
\Phi(w) = 2^{-K}(w-1)^K r(z(w)) , \quad \sigma(w) = 2^{-K}(w-1)^K s(z(w))
$$

(17)

where $z(w) = (w+1)/(w-1)$.

As an example we apply the above theory to the backward differentiation formula with $K=3$. In this case $\sigma(w) = w^3$; $s(z) = (z+1)^3$; $b_0 = \gamma_0 = 1$, $b_1 = \gamma_2 = 3$, $b_2 = \gamma_3 = 1$, $b_1 = 3$, $\gamma_1 = \frac{8}{3}$; $u_0 = 3/16$, $u_1 = 1/16$. If we write $e = e_1$, then $a_0 = e_0 = (16-e)/3$; $a_1 = 6$, $a_2 = 2$, $a_3 = 0$. The $A$-stability interval is $0 \leq e \leq 16$ (in this case $e_0 \geq 0$, $e_1 \geq 0$ are necessary, as well as sufficient, for ii) to hold). In its usual form, the formula in question, after multiplying by 24, becomes:

$$-(4-e)x_n + (24-3e)x_{n+1} - (60-3e)x_{n+2} + (40-e)x_{n+3} - 24hx_{n+3} = 0 .
$$

(18)

One easily calculates that the local truncation error of this formula is

$$-(4+e)h^3 x^{(3)}(t) + O(h^4) .
$$

For $e = -4$ one obtains the usual backward differentiation formula with $p=3$ which falls outside of the stability interval, as it should.
The method of lines (MOL) is a flexible algorithm for the numerical integration of systems of simultaneous nonlinear partial differential equations (PDE's) with an initial value independent variable. Our experience has indicated that it can be applied to a spectrum of scientific and engineering problems and therefore we have for several years undertaken a developmental effort to implement the MOL in general FORTRAN-based programs for PDE's. Basically, the spatial derivatives (i.e., the derivatives with respect to the boundary value independent variable(s)) are replaced with finite difference approximations. The resulting system of initial value ordinary differential equations (ODE's) is then integrated numerically to obtain a solution along lines of constant spatial independent variable(s). Multidimensional PDE's (i.e., with more than one spatial variable) can be handled as well as simultaneous PDE's with differing numbers of spatial independent variables. Since the MOL is oriented toward initial value ODE's, it is particularly well suited for the numerical integration of systems of mixed ODE/PDE's. These mixed systems occur in physical problems in which the ODE's are boundary conditions for the PDE's*.

Since the MOL is generally useful in scientific and engineering problems, yet has been discussed to only a limited extent in the existing texts on the numerical integration of PDE's, a study was

undertaken to better understand the computational characteristics and requirements of the algorithm. This paper summarizes the principal conclusions, particularly with regard to convergence and stability. The study is primarily numerical since this approach permitted a broader study than would be practical with a strictly analytical approach. In particular, a sliding difference algorithm is discussed which couples up to all of the points in the finite difference grid in approximating the spatial derivatives at a particular grid point.

THE FINITE DIFFERENCE APPROXIMATION AND ITS ORIGIN IN THE TAYLOR SERIES

Consider a one dimensional spatial domain which is divided into equal segments each \( \Delta x \) in length. It is desired to construct a finite difference approximation to the \((n-1)\) spatial derivatives at the kth point in the domain. Let the kth point be \( p \) points from the left of the set of \( n \) points which are to be coupled by the approximation. (If \( n=3 \) and \( p=2 \), we construct the three-point central difference at the kth point for \( y'_k \) and \( y''_k \).)

The value of \( y_i \) at any point within the set of \( n \) points may be approximated in terms of the function \( y_k \) and its \( n-1 \) derivatives, \( y_k \ldots y_k^{(n-1)} \), by the truncated Taylor series

\[
y_i \approx y_k + y'_k (i - k) \Delta x + y''_k \frac{(i - k)^2 (\Delta x)^2}{2!} + \ldots \\
+ y_{(n-1)} \frac{(i - k)^{n-1} (\Delta x)^{n-1}}{(n-1)!} ; i = k - p + 1, \ldots \\
k - p + n; i \neq k
\]

This results in \( n-1 \) equations which may be expressed as

\[
\begin{bmatrix}
(1-p)\Delta x & \frac{(1-p)^2(\Delta x)^2}{2!} & \ldots & \frac{(1-p)^{n-1}(\Delta x)^{n-1}}{(n-1)!} \\
\vdots & \vdots & \ddots & \vdots \\
(n-p)\Delta x & \frac{(n-p)^2(\Delta x)^2}{2!} & \ldots & \frac{(n-p)^{n-1}(\Delta x)^{n-1}}{(n-1)!}
\end{bmatrix}
\begin{bmatrix}
y'_k \\
\vdots \\
y_{(n-1)}
\end{bmatrix}
\approx
\begin{bmatrix}
y_{k-p+1} - y_k \\
\vdots \\
y_{k-p+n} - y_k
\end{bmatrix}
\]

(2)
Or, factoring the matrix
\[
\begin{bmatrix}
(1-p) & \frac{(1-p)^2}{2!} & \ldots & \frac{(1-p)^{n-1}}{(n-1)!} \\
\vdots & \ddots & \ddots & \ddots \\
(n-p) & \frac{(n-p)^2}{2!} & \ldots & \frac{(n-p)^{n-1}}{(n-1)!}
\end{bmatrix}
\times
\begin{bmatrix}
y'_k \\
y''_k \\
\vdots \\
y^{(n-1)}_k
\end{bmatrix}
\approx
\begin{bmatrix}
y_{k-p+1} - y_k \\
y_{k-p+2} - y_{k-p+1} \\
\vdots \\
y_{k-p+n} - y_{k-p+n}
\end{bmatrix}
\tag{3}
\]

Solving for the vector of the \( n-1 \) derivatives at the \( k \)th point in terms of the adjacent \( n \) points the following is obtained:
\[
\begin{bmatrix}
y'_k \\
y''_k \\
\vdots \\
y^{(n-1)}_k
\end{bmatrix}
\approx
\begin{bmatrix}
\frac{1}{\Delta x} & 0 & \ldots & 0 \\
\frac{1}{(\Delta x)^2} & \frac{1}{\Delta x} & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots \\
0 & \ldots & \frac{1}{(\Delta x)^{n-1}}
\end{bmatrix}
\times
\begin{bmatrix}
y'_k \\
y''_k \\
\vdots \\
y^{(n-1)}_k
\end{bmatrix}
\tag{4}
\]

From (4) we may obtain the coefficients of the dependent variable \( y \) at each of the coupled points to obtain the finite difference approximation to the first \( n-1 \) spatial derivatives at the \( k \)th point.

In general the \( i \)th derivative at the \( k \)th point of the set of \( n \) coupled points, with the \( k \)th point \( p \) points from the left end of the set of coupled points is
\[
y^{(i)}_k \approx \frac{1}{(\Delta x)^i} \sum_{j=1}^{n} r_{ij} y_{k-p+j}
\tag{5}
\]
The coefficients $r_{ijp}$ are obtained from the inverted matrix in (4). For all but the coefficient of $y_k$, $r_{ijp}$ is equal to the appropriate element of the row of the matrix corresponding to the desired derivative. The coefficient of $y_k$ is, simply, minus the sum of the elements of the appropriate row.

THE CLASSICAL APPROACH

If one considers the approximation of the spatial derivatives in a linear PDE along a grid of $N$ points, a system of $N$ initial value ODE's results. The eigenvalues of this system (which will be termed "temporal eigenvalues" since they pertain to the variation of the solution with respect to the initial value independent variable) then approximate the first $N$ eigenvalues of the original continuous (i.e., PDE) system. Therefore, one approach to the analysis of the MOL is the comparison of the temporal eigenvalues of the approximating ODE's with the exact eigenvalues of the continuous problem. Differences between the exact and approximate eigenvalues for increasing numbers of grid points gives a direct indication of the convergence characteristics of a particular MOL algorithm. Also, a comparison of the smallest and largest eigenvalues for the approximating system of ODE's gives an indication of the stability characteristics of the ODE's with respect to numerical integration (i.e., the stiffness of the ODE's).

This latter point, the stiffness, is an important consideration. The continuous problem generally has an infinity of temporal eigenvalues and therefore is infinitely stiff. Consider, for example, the separation of variables solution of the linear one-dimensional heat conduction equation

$$u_t = u_{xx}$$

with the Neumann boundary conditions

$$u_x(0,t) = u_x(1,t) = 0$$

The temporal eigenvalues for this problem are $\lambda_n = -(nm)^2$, $n = 0, 1, 2, 3, \ldots$. The approximating ODE's for an $N+1$ point grid will become stiffer as $N$ increases since the $N$ temporal eigenvalues approximate the first $N$ exact eigenvalues of the continuous problem. Consequently, as one attempts to improve the accuracy of the finite difference approximation for the spatial derivatives in the MOL by increasing the grid size, the approximating ODE's increase in number and stiffness. This significantly increases the computational requirements in the numerical integration of the ODE's when either an implicit or explicit algorithm is used.
To illustrate some of these ideas, consider the use of the second-order approximation

\[ u_{xx} = (u_{k+1} - 2u_k + u_{k-1})/\Delta x^2, \quad k = 0, 1, 2, \ldots N \]  

(9)
in equation (6). Boundary conditions (7) and (8) can also be approximated by central finite differences

\[ u_x(0, t) = (u_1 - u_0)/(2\Delta x) \]  

(10)

\[ u_x(1, t) = (u_{N+1} - u_{N-1})/(2\Delta x) \]  

(11)

so that the entire MOL is based on second-order correct central differences (CD). This second-order behavior is demonstrated in Figure 1 where the error in the first ten temporal eigenvalues (i.e., \( \log((\lambda_n, \text{approx} - \lambda_n, \text{exact})/\lambda_n, \text{exact} \times 100) \), \( n = 1, 2, 3, \ldots 10 \)) is plotted vs the number of grid points (i.e., \( \log(N) \)).

![Figure 1: Error of the First Ten Eigenvalues vs Number of Grid Points for Equations (6) to (8); CD algorithm](image)

Boundary conditions (7) and (8) can also be approximated by the second-order three-point noncentral finite differences
In this approach, the fictitious points \( u_{-1} \) and \( u_{N+1} \) in equations (10) and (11) are avoided; rather only internal and boundary grid points are used so that the method is termed "internal differencing" (ID). The error in the first ten eigenvalues for ID is plotted in Figure 2. The departure from linearity for small \( N \) occurred when the eigenvalues of the approximating ODE's were complex. In this case, the absolute value of the complex numbers was used in constructing Figure 2.

To obtain an indication of the stiffness characteristics of the CD and ID algorithms, the absolute value of the ratio of the largest to smallest eigenvalues, subsequently termed the "stiffness ratio" (SR), is plotted vs \( N \) logarithmically in Figures 3 and 4. The limiting values of SR are also indicated beneath the graphs.

Figures 1 to 4 demonstrate the well-known conflicting requirements of accuracy (or convergence to the exact solution) and stability. Although they apply to only a single, simple linear example, equations (6) to (8), the conclusion is generally valid, that is, improving the accuracy by increasing the grid also aggravates the stability problem.

\[
\begin{align*}
    u_x(0,t) &= (-3u_0 + 4u_1 - u_2)/(2\Delta x) \quad (12) \\
    u_x(1,t) &= (3u_N - 4u_{N-1} + u_{N-2})/(2\Delta x) \quad (13)
\end{align*}
\]

In this approach, the fictitious points \( u_{-1} \) and \( u_{N+1} \) in equations (10) and (11) are avoided; rather only internal and boundary grid points are used so that the method is termed "internal differencing" (ID). The error in the first ten eigenvalues for ID is plotted in Figure 2. The departure from linearity for small \( N \) occurred when the eigenvalues of the approximating ODE's were complex. In this case, the absolute value of the complex numbers was used in constructing Figure 2.

To obtain an indication of the stiffness characteristics of the CD and ID algorithms, the absolute value of the ratio of the largest to smallest eigenvalues, subsequently termed the "stiffness ratio" (SR), is plotted vs \( N \) logarithmically in Figures 3 and 4. The limiting values of SR are also indicated beneath the graphs.

Figures 1 to 4 demonstrate the well-known conflicting requirements of accuracy (or convergence to the exact solution) and stability. Although they apply to only a single, simple linear example, equations (6) to (8), the conclusion is generally valid, that is, improving the accuracy by increasing the grid also aggravates the stability problem.

\[
\begin{align*}
    u_x(0,t) &= (-3u_0 + 4u_1 - u_2)/(2\Delta x) \quad (12) \\
    u_x(1,t) &= (3u_N - 4u_{N-1} + u_{N-2})/(2\Delta x) \quad (13)
\end{align*}
\]

In this approach, the fictitious points \( u_{-1} \) and \( u_{N+1} \) in equations (10) and (11) are avoided; rather only internal and boundary grid points are used so that the method is termed "internal differencing" (ID). The error in the first ten eigenvalues for ID is plotted in Figure 2. The departure from linearity for small \( N \) occurred when the eigenvalues of the approximating ODE's were complex. In this case, the absolute value of the complex numbers was used in constructing Figure 2.

To obtain an indication of the stiffness characteristics of the CD and ID algorithms, the absolute value of the ratio of the largest to smallest eigenvalues, subsequently termed the "stiffness ratio" (SR), is plotted vs \( N \) logarithmically in Figures 3 and 4. The limiting values of SR are also indicated beneath the graphs.

Figures 1 to 4 demonstrate the well-known conflicting requirements of accuracy (or convergence to the exact solution) and stability. Although they apply to only a single, simple linear example, equations (6) to (8), the conclusion is generally valid, that is, improving the accuracy by increasing the grid also aggravates the stability problem.
Figure 3: Stiffness Ratio vs Number of Grid Points for Equations (6) to (8); CD Algorithm, $10 \leq N \leq 60$; $40.9 \leq SR \leq 1459$

Figure 4: Stiffness Ratio vs Number of Grid Points for Equations (6) to (8); ID Algorithm, $10 \leq N \leq 60$, $39.1 \leq SR \leq 1458$
A further comparison of the CD and ID algorithms is reported elsewhere as well as a detailed discussion of the associated eigenvalue analysis.*

THE SLIDING DIFFERENCE ALGORITHM

The preceding implementation of the MOL is based on relatively low-order finite difference approximations which span only a few grid points in the neighborhood of the point for which the approximation is written. For example, the central difference approximation of equation (9) spans three points. However, as shown above, the Taylor series can be used to derive finite difference approximations of higher-order accuracy which span a larger number of points. In fact, by using noncentral difference approximations such as equations (12) and (13), one can consider the use of approximations which involve all of the grid points when writing replacements for the spatial derivatives at each grid point. Thus in this limiting case, each grid point has a different finite difference approximation. Furthermore, a FORTRAN program has been developed which automatically obtains the finite difference approximation of a given order with a prescribed number of active grid points to the left and right of the point of interest. To reiterate the basic procedure:

1. A grid of N+2 points is constructed where N is now the total number of internal grid points.
2. At each of the internal grid points, the spatial derivatives are approximated in terms of n coupled points. n will always be odd.
3. Whenever possible, central difference approximations are used.
4. The approximations "slide" left or right whenever a boundary point is encountered so that only N+2 points are used.
5. The value of the dependent variable at the boundary points is computed from the boundary conditions (and not the PDE) in terms of the dependent variable at the internal points using the appropriate n-point forward difference at the left boundary and backward difference at the right boundary.

Applying rules 1 through 3 to equation (5) we obtain the results shown in Table I. This approach is termed the "sliding difference" method (SD).

The eigenvalue analysis can then be performed as explained previously for the case of low-order approximations. The essential difference, however, is that instead of considering the variation of convergence and stability characteristics with the total number of grid points, one can now consider how these characteristics change with the number of coupled points in the finite difference approximation for a grid of a fixed size. For example, the ID and SD algorithms can be compared which have the following characteristics:

1. ID algorithm
   - Number of coupled points: \( n = 3 \)
   - Number of active grid points: \( 29 < N < 59 \)

2. SD algorithm
   - Number of coupled points: \( 3 < n < 31 \)
   - Number of active grid points: \( N = 29 \)

Note that the two algorithms are equivalent when \( N = 29 \) for ID and \( n = 3 \) for SD.

If the SD algorithm is applied to the basic linear problem, equations (6) to (8), the per cent error in the first ten eigenvalues varies with the number of coupled grid points, \( n \), as indicated in Figure 5. In this case, the absolute value of the per cent error is plotted to eliminate the excursions around zero error which tend to obscure the trends in the numerical data. The higher accuracy with increasing number of coupled points (labelled NC in Figure 5) is evident. Note that the grid size has remained fixed at 29 internal points; only the degree of coupling between the approximating 29 ODE's changes.

One might expect in view of the preceding discussion that in order to achieve this higher accuracy, a concomitant increase in the stiffness ratio (SR) would occur. However, this turns out not to be the case. The SR for this same problem is plotted in Figure 6 for the SD algorithm while the same data are plotted logarithmically in Figure 7. The maximum in SR with increasing number of coupled points is particularly noteworthy. Thus improved accuracy can be achieved without an irreversible increase in the severity of the stability problem.
<table>
<thead>
<tr>
<th>$y^{(1)}_k$</th>
<th>$p$</th>
<th>$k$</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{(\Delta x)^I} \sum_{j=1}^n r_{ijp} y_{k-p+j}$</td>
<td>$\frac{n+1}{2}$</td>
<td>$\frac{n+1}{2} \leq k \leq N - \frac{n-1}{2}$</td>
<td>Central Difference No Boundary Points</td>
</tr>
<tr>
<td>$\frac{1}{(\Delta x)^I} \left[ r_{i1p} y_o + \sum_{k=2}^n r_{ijp} y_{k-p+j} \right]$</td>
<td>$k+1$</td>
<td>$1 \leq k \leq \frac{n-1}{2}$</td>
<td>Sliding Left Left boundary point encountered</td>
</tr>
<tr>
<td>$\frac{1}{(\Delta x)^I} \left[ \sum_{j=1}^{n-1} r_{ijp} y_{k-p+j} + r_{inp} y_{N+1} \right]$</td>
<td>$n-N+k-1$</td>
<td>$N - \frac{n-1}{2} + 1 \leq k \leq N$</td>
<td>Sliding Right Right boundary point encountered</td>
</tr>
<tr>
<td>$\frac{1}{(\Delta x)^I} \left[ r_{i1p} y_o + \sum_{j=2}^{n-1} r_{ijp} y_{k-p+j} + r_{inp} y_{N+1} \right]$</td>
<td>$k+1$</td>
<td>$1 \leq k \leq N$</td>
<td>Continuous Sliding All points coupled; $n = N + 2$</td>
</tr>
</tbody>
</table>

Table I
Figure 5: Error of the First Ten Eigenvalues for Equations (6) to (8); SD Algorithm, $3 \leq n \leq 31$, $N = 29$

Figure 6: Stiffness Ratio vs Number of Coupled Grid Points for Equations (6) to (8); SD Algorithm, $3 \leq n \leq 31$, $N = 29$
Finally, an attempt was made to assign a single figure of merit to the ID and SD algorithms so that they can be compared by a simple visual inspection. To this end, a cost factor, CF, is defined as

$$CF = \text{(time for 100 calls to the derivative subroutine for the approximating ODE's) x (stiffness ratio)} \quad (14)$$

CF defined by equation (14) was selected primarily for explicit integration algorithms. Thus as the total number of grid points for the ID algorithm or the number of coupled points for the SD algorithm is increased, the amount of arithmetic required to evaluate the derivatives of the approximating ODE's increases thereby adding to the cost of the computer solution. This is reflected in the first factor in CF. Similarly, as the stiffness ratio increases, more evaluations of the ODE derivative vector are required since the integration step size for a stable numerical solution must decrease. This is reflected in the second factor in CF.

The ID and SD algorithms can now be compared with respect to the CF for a given accuracy. This is demonstrated in Figures 8 and 9 where the CF is plotted against the error in the first and fifth
Figure 8: Cost Factor vs Accuracy of the First Eigenvalue for Equations (6) to (8); SD Algorithm, $3 \leq n \leq 31$, $N = 29$; ID Algorithm, $29 \leq N \leq 59$, $n = 3$

Figure 9: Cost Factor vs Accuracy of the Fifth Eigenvalue for Equations (6) to (8); SD Algorithm, $3 \leq n \leq 31$, $N = 29$; ID Algorithm, $29 \leq N \leq 59$, $n = 3$
eigenvalues for the linear problem, equations (6) to (8). Clearly the SD algorithm is superior with regard to the CF for a given accuracy, particularly for the lower (i.e., first) eigenvalue. In general this conclusion was true for all of the eigenvalues of the test problem which were studied; this superior cost-accuracy performance of the SD algorithm is noteworthy since the lower eigenvalues most significantly affect the transient solution.

The same general conclusion would probably also apply to implicit numerical integration of the approximating ODE's. Specifically, if acceptable accuracy can be achieved with a smaller total number of grid points via the SD algorithm, then the computational effort required in the matrix inversion for the implicit algorithm will be reduced. Thus unless some efficient sparse matrix algorithm for the matrix inversion can be applied, the inversion of the fuller, but smaller, matrix in the SD method may be less time consuming. Additionally, the user need not be concerned with tailoring a sparse matrix routine to the particular problem of interest.

CONCLUSION

The MOL algorithm when implemented in the conventional sense via low-order finite difference approximations for the PDE spatial derivatives leads to systems of ODE's which may be stiff if good accuracy is required with respect to the spatial independent variables. The stiff differential equations can be integrated by implicit algorithms to circumvent the stability constraint. The approximating ODE's will be only locally coupled so that some sparse matrix procedure for the implicit algorithm can be used to good advantage. In general, this approach is frequently dictated by the physical problem which defines the sparsity (i.e., degree of coupling) of the ODE's. For example, the topology of an electrical circuit will usually define a sparse mathematical structure.

However, in the case of PDE's, the degree of coupling is open to the analyst. In fact, as the coupling increases, the accuracy improves since the spatial derivatives can be approximated by higher order finite differences. Fortunately, as this study demonstrated for one test problem, the stability problem does not become more severe as the coupling increases so that a classical explicit integration algorithm may be satisfactory. If this is not the case, then an implicit algorithm may be computationally more efficient since fuller but smaller matrices are involved in the SD algorithm for a given accuracy.

However, all of the results presented are empirical and for only one linear PDE test problem (i.e., equations (6) to (8)). Similar results were observed for three other linear problems for
which exact eigenvalues are known.* The study, however, should be extended to variable coefficient and nonlinear PDE's which are linearized locally. One of the authors** has analyzed the eigenvalues for a PDE system which models a nonlinear reactor. Linearization followed by a systematic search was performed to obtain the temporal eigenvalues for the simultaneous nonlinear PDE's.

Finally, the authors are now considering the implementation of the SD algorithm in a general FORTRAN-based system for PDE's. This work will be discussed in a future paper.

*1973 SCSC paper cited previously

ECONOMICAL GLOBAL ERROR ESTIMATION

Hans J. Stetter
Technical University, Vienna, Austria

1. Introduction

This paper has no immediate relation to stiff differential equations; in fact, modifications which may be necessary to adapt some of the following techniques to badly stiff systems have not been elaborated. The main feature of the paper is the analysis of a new technique for global error estimation [Zadunaisky (1970)] which should be of genuine interest to everybody concerned with the numerical solution of differential equations.

At first, we will survey the general principles of global error estimation, with some emphasis on a technique which avoids the integration of an "error equation" [Stetter (1971)]. Next, a non-standard approach to estimating the local discretization error will be presented which uses a residual in the differential equation rather than a difference residual. This "differential estimate" of the local error will be the basis for obtaining a global error estimate which will turn out to be exceedingly accurate (asymptotically as h→0) in a sufficiently smooth situation.

A close analogy between this technique and the well-known difference correction technique [e.g. Pereyra (1967)] will be established and it will be shown that our new "differential correction" technique may also be used in an iterated fashion.
2. The Principles of Global Error Estimation

We have the following task: Given an initial value problem (the interval has been normalized)

\[ y(0) = y_0, \quad y' = f(t, y) \quad \text{in} \quad [0, T], \quad y(t) \in \mathbb{R}^s, \]

and its discretization obtained by a method M with a certain stepsize parameter \( h \) (which is assumed to characterize the setting of a step control). By \( z, \zeta_h, \) and \( \tilde{z}_h \) we denote the solutions of the ODE (2.1), the accurate solution of the discretization, and a computational solution of the discretization resp. We want a vector-valued function \( \tilde{\varepsilon} \) such that one of the following relations holds componentwise

\[ ||\zeta_h(t) - z(t)|| \leq \tilde{\varepsilon}(t) \quad (\forall t) \]

or

\[ \zeta_h(t) - z(t) \approx \tilde{\varepsilon}(t). \]

(2.2) is called a global error bound (absolute or relative),
(2.3) a global error estimate.

Ideally, one should be able to specify \( \tilde{\varepsilon} \) and to determine the step control such that (2.2) holds. Practically, one may at best hope to determine \( \tilde{\varepsilon} \) a posteriori for a given computation.

It may seem that the request for an appraisal of the global error \( \zeta_h - z \) is not appropriate: A backward error analysis would rather suggest the construction of a differentiable function \( \tilde{z} \) which takes the same values as \( \zeta_h \) on the "grid" \( G_h := \{ t \} \) and which satisfies

\[ \tilde{z}'(t) = \tilde{f}(t, \tilde{z}(t)) \]

where (componentwise)

\[ ||\tilde{f}(t, \tilde{z}(t)) - f(t, \tilde{z}(t))|| \leq \delta(t) \quad (\forall t) \]

or

\[ \tilde{f}(t, \tilde{z}(t)) - f(t, \tilde{z}(t)) \approx \delta(t). \]

Thus the components of \( \delta \) would interpret the effects of
the discretization on the solution values as the effect of a perturbation of the right-hand side of (2.1); one might claim that the establishment of the smallness of $\delta$ relative to actual data uncertainties would be sufficient.

This argument is not fully valid:

a) Perturbations of similar magnitude but different properties (e.g. constant versus oscillating) may cause considerably different effects on the solution of (2.1).
b) One may not be able to judge the qualitative and quantitative effects of perturbations of (2.1) at all.

Global error bounds. In principle, it is well feasible to obtain strict bounds (2.2) for the error of a numerical solution $\bar{z}_h$. The methods proposed by various authors [Moore (1966), Kahan (1966), and others] bound the generated error per step and carefully consider the propagation of the errors along the field of possible solutions. However, in most practical applications, particularly with large $s$, the bounds obtained over a longer interval of integration become unrealistically large, even when a tremendous effort has been spent in their calculation. In the following, we will not consider methods for generating strict global error bounds.

Global error estimation. To be able to display the principal approach in a "shorthand" notation let us denote the difference equation into which (2.1) is discretized (and which defines $\bar{z}_h$) by

\begin{equation}
F_h\bar{z}_h = 0.
\end{equation}

The computational solution $\bar{z}_h$ leaves a residual

\begin{equation}
F_h\bar{z}_h^* = \rho \quad \text{"local round-off error"}
\end{equation}
in (2.4) while the analytic solution $z$ of (2.1), or rather its "reduction" $\Delta_h z$ to the grid $G_h$, leaves another residual

\begin{equation}
F_h \Delta_h z = l_h \quad \text{"local discretization error"}.
\end{equation}

From (2.5) and (2.6) we obtain the "error equation"

\begin{equation}
F_h' (\zeta_h) \left( \bar{z}_h - \Delta_h z \right) = \rho - l_h.
\end{equation}
where $F'_h(\zeta_h)$ denotes the variational equation associated with our difference equation (2.4) along $\zeta_h$ and where terms of higher order in $|\bar{\xi}_h - \Delta_h z|$ have been neglected. Neglection of $\rho$ relative to $1_h$ and the corresponding assumption $\bar{\xi}_h \approx \xi_h$ yield the linear difference equation

$$(2.7) \quad F'_h(\zeta_h)(\zeta_h - \Delta_h z) = -1_h.$$  

Obviously, the computation of an estimate for $\bar{\xi}_h - \Delta_h z$ on the basis of (2.7) requires

(i) an estimation of the local discretization error $1_h'$,
(ii) an integration of the error equation (2.7).

The computational effort for (ii) is generally of the same order as that necessary for the computation of the approximate solution $\xi_h'$. Some economization may be obtained through the following observation:

If the situation (problem and discretization) is sufficiently smooth, there exists a "principal error function" $\varphi : \mathbb{R} \times \mathbb{R}^S \to \mathbb{R}^S$ such that ($p$ is the order of our method $M$)

$$(2.8) \quad -1_h'(t_{\nu}) = h^p \varphi(t_{\nu}, \xi_h(t_{\nu})) + O(h^{p+1}).$$

Furthermore, the solution $e$ of

$$(2.9) \quad e' - f_y(t, z(t)) e = \varphi(t, z(t)),$$

with appropriate initial conditions, satisfies

$$(2.10) \quad \xi_h(t_{\nu}) - z(t_{\nu}) = h^p e(t_{\nu}) + O(h^{p+1}).$$

If we consider only lowest order terms in $h$, we may replace the integration of (2.7) by an integration of (2.9) and use $h^p e$ as our estimate, see (2.10). This numerical integration of (2.9) may be carried out with a simpler method and on a wider grid than used for the computation of $\xi_h'$; this may considerably reduce the "cost" involved.

There is even a possibility to avoid the integration at all [Stetter (1971A)]: For certain discretization methods, (2.9) becomes an "exact" differential equation, i.e. there exists a function $Q : \mathbb{R} \times \mathbb{R}^S \to \mathbb{R}^S$ such that
(2.11) \[ \frac{d}{dt} Q(t, e(t)) = 0 \] for a solution of (2.9).

Thus

\[ Q(t_v, e(t_v)) = Q(0, e(0)) \quad \text{for all } t_v \in G_h. \]

This looks like the ideal solution to the problem of global error estimation. Unfortunately, however, these methods have a number of shortcomings which restrict their general use. Further investigations will have to establish whether methods of this type may be reasonably implemented and perhaps even used with stiff systems.

3. Estimation of the Local Discretization Error

As we have seen in section 2 (cf. (2.7)-(2.9)) it is necessary to have estimates of the local discretization error if one wants to obtain global error estimates (the exceptional estimation method based on (2.11) will no longer be considered in the following). We will therefore shortly survey the principal approaches to the generation of such local estimates. (It should be mentioned that we always consider the local error per unit step.)

Virtually all existing estimation techniques employ a second difference operator \( F_h \) for the estimation of the local error \( l_h \) generated by the original operator \( F_h \) (cf. (2.6) for the definition of \( l_h \)). The underlying assumption is that the solution \( \zeta_h \) of the discretization (2.4) possesses - at least locally - an asymptotic expansion

\[ \zeta_h = \Delta_h [z + h^P e_P] + O(h^{P+1}). \]

This assumption is not trivial; it does not necessarily hold for multistep methods or immediately after step changes.

From (2.6), (2.8), and (3.1) we have (under suitable smoothness assumptions)
When we apply another difference operator $F_h$ of order $p$ or $p+1$ to $\zeta_h$ we obtain

\[
F_{h} \zeta_h = F_{h} \Delta_h [z + h^p e_p] + O(h^{p+1})
\]

(3.3)

Since both $F_h$ and $F_{h}$ are consistent of order $p$, we must have (for sufficiently smooth $e_p$)

\[
F_{h} (\Delta_h z) e_p = F_{h} (\Delta_h z) e_p + O(h^p).
\]

(3.4)

With (3.4), subtraction of (3.2) from (3.3) yields

a) if $F_h$ is of order $p$ and $\bar{\phi} = \overline{\phi}$

\[
F_h \zeta_h = (1-\bar{c}) h^p \phi + O(h^{p+1}),
\]

b) if $F_h$ is of order $p+1$

\[
F_h \zeta_h = h^p \phi + O(h^{p+1}).
\]

Thus (cf. (2.8))

\[
-1_{h}(t_v) = \begin{cases} 
\frac{1}{1-\bar{c}} F_h \zeta_h(t_v), \\
F_h \zeta_h(t_v).
\end{cases}
\]

(3.5)

In many cases, $F_h \zeta_h$ takes the form of $\zeta_h - \zeta_h$ where $\zeta_h$ is another approximation at the grid point $t_v$ in question.

A different approach which will become important in the sequel is the following: We construct a differentiable function $\tilde{z}$ which satisfies - at least locally -

\[
\tilde{z} = z + h^p e_p + O(h^{p+1})
\]

(3.6)

\[
\tilde{z}' = z' + h^p e_{p}' + O(h^{p+1}),
\]
e.g. by (local) interpolation of the $\zeta_h$-values. Then

$$\tilde{z}'(t) - f(t,\tilde{z}(t)) = z'(t) - f(t,z(t))$$

$$+ h^p (e'_p(t) - f_y(t,z(t)) e_p(t)) + O(h^{p+1})$$

$$= h^p \varphi(t,z(t)) + O(h^{p+1}) \text{ by (2.9).}$$

This implies (cf. (2.8))

$$(3.7) \quad l_h(t_v) = \tilde{z}'(t_v) - f(t_v,\tilde{z}(t_v)) + O(h^{p+1}).$$

We will call (3.7) a differential estimate of the local error and (3.5) a difference estimate. If $\tilde{z}$ depends linearly on the values of $\zeta_h$ and if $f$ is linear, (3.7) is equivalent to (3.5) with some usual difference operator $F_h$. Otherwise, the difference operators which would be defined by the equivalence of (3.7) and (3.5) are generally of a non-standard form.

The computational effort for the evaluation of (3.7) will normally consist of arithmetic operations (for the interpolation of $\tilde{z}'(t_v)$ and $\tilde{z}(t_v)$ from $\zeta_h$-values) and of one evaluation of $f$. Exactly the same effort is usually needed for the evaluation of (3.5).

4. A Differential Estimate for the Global Error

Normally, an estimate of the local error at the $t_v$ is used in the error equation (2.7) or in a suitable discretization of (2.9). In the case of a differential estimate (3.7), however, this estimate is defined as a function of $t$ on the whole interval $[0,T]$ so that we may assume the function $\varphi(t,z(t))$ in (2.9) as known for arbitrary $t \in [0,T]$. Furthermore, with a sufficiently accurate interpolation of the $\zeta_h$-values the function $\tilde{z}$ represents the asymptotic expansion of $\zeta_h$ correctly to higher terms than just $h^p e'_p$.

Thus, with a sufficiently accurate discretization of the differential equation

$$(4.1) \quad w(0) = 0^*, \quad w' - f_y(t,\tilde{z}(t))w = d_h(t),$$

where

*) This assumes $\zeta_h(0) = z(0)$ which is normally the case. Other starting conditions may be taken into account.
we may expect to obtain an estimate of the global error \( \zeta_h - \Delta_h z \) which is correct to a higher order of \( h \) than \( h^{p+1} \).

**Theorem 1:** With \( p \in \mathbb{N}, \ p < \tilde{p} \leq 2p \), assume that

(i) the solution \( \zeta_h \) of our discretization of order \( p \) possesses an asymptotic expansion to order \( \tilde{p}-1 \) in \([0, T]\):

\[
\zeta_h(t_v) - z(t_v) = \sum_{j=\tilde{p}}^{p-1} h^j e_j(t_v) + O(h^{\tilde{p}}); \tag{4.3}
\]

(ii) the interpolation function \( \tilde{z} \) satisfies in \([0, T]\)

\[
\tilde{z}(t) - z(t) = \sum_{j=\tilde{p}}^{p-1} h^j e_j(t) + O(h^{\tilde{p}}), \tag{4.4}
\]

\[
\tilde{z}'(t) - z'(t) = \sum_{j=\tilde{p}}^{p-1} h^j e_j'(t) + O(h^{\tilde{p}}). \tag{4.5}
\]

Consider a discretization method \( M \) of order \( \tilde{p} - p \) which discretizes linear differential equations into linear difference equations*). Then the solution \( \omega_h \) of the discretization of (4.1) by \( M \) satisfies

\[
\zeta_h(t_v) = \omega_h(t_v) + O(h^{\tilde{p}}). \tag{4.6}
\]

**Sketch of proof**: \( \tilde{w} = \sum_{j=\tilde{p}}^{p-1} h^j e_j \) satisfies (4.1) with an error of \( O(h^{\tilde{p}}) \): By (4.4) and (4.5), we have

\[
\tilde{w}'(t) = f_y(t, \tilde{z}(t)) w(t)
\]

\[
= (\tilde{z}(t) - z(t))' + O(h^{\tilde{p}}) - f_y(t, \tilde{z}(t)) \cdot (\tilde{z}(t) - z(t)) + O(h^{\tilde{p}})
\]

\[
= \tilde{z}'(t) - f_y(t, \tilde{z}(t)) - z'(t) + f_y(t, \tilde{z}(t)) + z'(t) + f_y(t, z(t)) \cdot (\tilde{z}(t) - z(t))
\]

\[
- f_y(t, z(t)) \cdot (\tilde{z}(t) - z(t)) + O(h^{\tilde{p}})
\]

\[
= d_h(t) - \left[ f_y(t, \tilde{z}(t)) - f_y(t, z(t)) \right] \cdot (\tilde{z}(t) - z(t)) + O(h^{\tilde{p}})
\]

\[
= d_h(t) + O(h^{\tilde{p}})
\]

for Lipschitz-continuous \( f_y \) since \( \tilde{z} - z = O(h^{\tilde{p}}) \) and \( 2p \geq \tilde{p} \).

Thus the correct solution of (4.1) satisfies

*) A more concise formulation will be given in a forthcoming paper by the author.
ECONOMICAL GLOBAL ERROR ESTIMATION

(4.7) \[ w(t) = \tilde{w}(t) + O(h^{\bar{p}}) = \sum_{j=p}^{\bar{p}} h^j e_j(t) + O(h^{\bar{p}}). \]

By our assumption on \( \text{M} \), \( \omega_h \) will reproduce \( w \) with a relative error of \( O(h^{\bar{p}-p}) \):

\[ \omega_h(t) = w(t) (1 + O(h^{\bar{p}-p})) = \sum_{j=p}^{\bar{p}} h^j e_j(t) + O(h^{\bar{p}}) \]

by (4.7). This implies (4.6) by virtue of (4.3).

Theorem 1 shows (take \( \bar{p} = 2p \)) that in a sufficiently smooth situation and with a sufficiently good interpolation, integration of (4.1) by our original method \( \text{M} \) of order \( p \) leads to an estimate (4.6) of the global error which is correct except for terms of \( O(h^{2p}) \). In any case, the efforts spent in the computation of \( z \) and \( z' \) and in the numerical solution of (4.1) may be adapted to each other and to the smoothness of the original problem (2.1) and of the method \( \text{M} \) both of which determine the size of \( p \) in (4.3).

5. Zadunaisky's Approach

The estimation technique described in the previous section 4, with \( \bar{p} = p+1 \), was contained in a paper presented by P.E. Zadunaisky at the '73 Dundee Conference. The main content of his paper was, however, an elaboration of an idea which he had previously presented at Astronomers' conferences [Zadunaisky (1966), (1970)]. This is the essence of his approach:

Together with (2.1) consider another differential equation

(5.1) \[ \tilde{y}(0) = y_0', \quad \tilde{y}' = \tilde{f}(t,\tilde{y}), \quad \tilde{y} \in \mathbb{R}^n, \]

on the same interval \([0,T]\); (5.1) must be constructed such that its analytic solution \( \tilde{z} \) is known and that \( \tilde{f} \) is "close" to \( f \). Due to this closeness it may be expected that the global error generated by our discretization method \( \text{M} \) for (5.1) is nearly the same as that generated for (2.1). Hence we may use the difference between the numerical solution \( \tilde{z} \) of (5.1) and \( \tilde{z} \) as an estimate for the global error of \( \text{M} \) on (2.1):

*) A more concise formulation will be given in a forthcoming paper by the author.
For the construction of $\tilde{f}$, Zadunaisky suggests to choose $\tilde{z}$ and to form

$$f(t,y) = f(t,y) + \bar{d}_h(t)$$

where

$$\bar{d}_h(t) := \tilde{z}'(t) - f(t,\tilde{z}(t)).$$

Obviously $\tilde{z}$ is the analytic solution of (5.1) with (5.3)/(5.4). If $\bar{z}$ is chosen as an interpolant of the $\zeta_h$-values, $\bar{d}_h$ will be small so that $f$ and $\bar{f}$ are close.

The success of this intuitive approach is due to the fact that the most standard discretization methods $M$ possess the following property:

**Property (E):** Let $M$ be of order $p$ and assume that both (2.1) and (5.1)/(5.3) are sufficiently smooth so that their numerical solutions by $M$ possess an asymptotic expansion to order $\bar{p}-1$, $p<\bar{p} \leq 2p$, i.e. that we have (4.3) and

$$\zeta_h(t,v) - \bar{z}(t,v) = \sum_{j=p}^{p+1} \frac{h^j}{j!} e_j(t,v) + O(h^{\bar{p}}).$$

$M$ is said to possess property (E) if

$$\bar{d}_h(t) = O(h^{\bar{p}-p}) \text{ uniformly in } [0,T]$$

implies, for $t \in [0,T],$

$$e_j(t) = e_j(t) + O(h^{\bar{p}-j}), \quad j = p(1)\bar{p}-1.$$

**Theorem 2:** With $p \in \mathbb{N}$, $p<\bar{p} \leq 2p$, assume that

(i) (4.3) holds for the solution $\zeta_h$ of the discretization of (2.1) by $M$;

(ii) $\bar{z}$ in (5.4) has been chosen such that the solution $\zeta_h$ of the discretization of (5.1)/(5.3) by $M$ satisfies (5.5) and that $\bar{d}_h$ defined by (5.4) satisfies (5.6);

(iii) $M$ has property (E).

Then

$$\zeta_h(t,v) - z(t,v) = \zeta_h(t,v) - \bar{z}(t,v) + O(h^{\bar{p}}).$$
**Proof:** The assertion (5.8) follows immediately from (5.5), (5.7), and (4.3). *

Let us note at first that with a function $\tilde{z}$ which satisfies (4.4)/(4.5) we have $\tilde{d}_h(t) = O(h^p)$. Hence, if we are able to choose $p = 2p$ in (4.3)-(4.5) we achieve once more an estimate with an error of only $O(h^{2p})$; cf. the remark after the proof of Theorem 1. Thus the technique described above is just as powerful as that of section 4, in a sufficiently smooth situation and with sufficiently accurate interpolation. If $z$ satisfies (4.4)/(4.5) $\tilde{d}_h$ is, of course, a differential estimate of the error and (5.8) may again be called a differential estimate of the global error.

There is, however, a fundamental difference between the two approaches when $p \leq 2p$. In this case, in Theorem 1 we could choose a method $M$ of the lower order $p < p$ without invalidating the estimate (4.6) as long as the interpolation assumption (4.4)/(4.5) was satisfied. In Theorem 2, on the other hand, we have to stick to the method $M$ while we may considerably weaken the assumptions on $\tilde{z}$. In fact, $\tilde{z}$ may be chosen completely independent of $\zeta_h$ as long as it is close enough to $z$ in the sense of (5.4)/(5.6). Interpolation of $\zeta_h$-values is now merely a convenient way of constructing $\tilde{z}$. Thus, $\tilde{d}_h$ need not be an estimate of the local truncation error at all - as was assumed by Zadunaisky in the analysis that he gave of his method.

6. Implementation

Zadunaisky has reported excellent results of his method in real-life problems [Zadunaisky (1966), (1970), Dundee conference]. For $\tilde{z}$ or $z$ he seems to have used high degree interpolation polynomials; details of the implementation could not be gathered from his reports.

In any case it is clear that for practical applications the asymptotic results of section 4 and 5 can only serve as a guideline. More important than the asymptotic

*) This shows that the real difficulty lies in the establishment of property (E) for a given method $M$ in a given situation. In the forthcoming paper (mentioned earlier), we will indicate the range of validity of property (E).
behavior of the error of the estimate is its actual size for a given realistic value of \( h \).

Experiments by the author of this paper have not yet led to valid conclusions. The main difficulty of the approach of Theorem 1 lies in the fact that a very accurate interpolation is required if \( M \) is a higher order method. Furthermore, the formation of \( f_0 \) may be undesirable for large systems. In the approach of Theorem 2 one may get away with ordinary cubic spline interpolation. On the other hand, the effort for the numerical integration of (5.1) with the aid of the original method \( M \) is as large as that for the computation of \( \zeta_h \), except when local iterations are needed where the value of \( \zeta_h \) may be used as a good guess for \( \zeta_h \).

In both approaches, the arithmetic accuracy of the computations has to be such that round-off effects are negligible compared to discretization effects. (This is, of course, a necessary prerequisite for all estimation techniques.)

The optimal treatment of the starting phase in multistep methods, of stepsize changes, or of order changes in variable-order methods has not yet become clear. The intuitive reasoning given at the beginning of section 5 would indicate that the best results are to be expected when the "history" of \( M \) (step change, order change, etc.) is simply repeated in the integration of (5.1). The application of the whole approach to stiff systems will certainly require extra precautions.

It is hoped that a more comprehensive report on the implementation of the new technique and on its merits and pitfalls may be presented in the near future.

7. Iterated differential correction

The old paradox of error estimation arises also in the present case: If a good estimate is available, it may be used to correct the original approximation; this generates an improved approximation but leaves us again without an error estimate.

On the other hand, the process of estimation and correction may be repeated; the feasibility of this procedure - originally conceived by L. Fox - has been analyzed and demonstrated by Pereyra in various papers.
ECONOMICAL GLOBAL ERROR ESTIMATION

[Pererya (1966, 1967, ..., 1973)]. In analogy to Pereyra's "iterated difference correction" which uses "difference estimates" of the local error (cf. section 3) we will call a similar technique which is based on a differential estimate and the approach of sections 4 or 5 "iterated differential correction".

Here is a short description of such a process:

A) Based on the estimation procedure of section 4:

Let \( \zeta_h^{[1]} := \zeta_h; \) for \( m = 1, 2, ... \) do:

(i) Form \( \tilde{z}^{[m]} \) by interpolation of \( \zeta_h^{[m]} \);

(ii) form \( d_h^{[m]} := \tilde{z}^{[m]} - f(t, z_h^{[m]}) \);

(iii) solve \( w' = f(t, \tilde{z}^{[m]}) w + d_h^{[m]} \) by a method \( \tilde{y} \) to obtain \( \omega_h^{[m]} \);

(iv) form \( \zeta_h^{[m+1]} := \zeta_h^{[m]} - \omega_h^{[m]} \).

B) Based on the estimation procedure of section 5:

Let \( \zeta_h^{[1]} := \zeta_h; \) for \( m = 1, 2, ... \) do:

(i) Form \( \tilde{z}^{[m]} \) by interpolation of \( \zeta_h^{[m]} \);

(ii) form \( d_h^{[m]} := z^{[m]} - f(t, z_h^{[m]}) \);

(iii) solve \( y' = f(t, \tilde{z}^{[m]}) - \sum_{\mu=1}^{m} d_h^{[\mu]} \) by the method \( \tilde{y} \) to obtain \( \zeta_h^{[m+1]} \).

The remarks on the choice of \( \tilde{z} \) and \( z \) in step (i) and on the choice of the method in step (iii) apply again. Depending on these choices and on the smoothness of the situation, an improvement of order \( O(h^{p-p}) \) may be expected in each cycle of the iteration. Thus under optimal conditions one may obtain

\( \zeta_h^{[m]}(t_{\nu}) - z(t_{\nu}) = O(h^{mp}) \), resp. \( \zeta_h^{[m]}(t_{\nu}) - z(t_{\nu}) = O(h^{mp}) \).

To what extent an actual improvement for a given \( h > 0 \) will correspond to this asymptotic improvement, will strongly depend on the situation. With initial value problems for ODEs where higher order methods are readily available, one would not expect that much can be gained from a repeated iteration of the differential correction procedure.
8. Conclusions

It is obvious that the intuitive approach of Zadunaisky may be used on a much wider class of problems than initial value problems for ODE. The main prerequisites are the existence of an asymptotic expansion for the solution of the discretization and the possibility of generating sufficiently good interpolants of the discrete solution values. But even in cases where a theoretical analysis may fail the method may prove successful in an "engineering" fashion.

Thus it may be expected that reasonable global error estimates may be obtained with Zadunaisky's method for discretizations of two-point boundary value problems, of various problems with partial differential equations, and of integral equations. In all these cases the necessary computational effort for the error estimation should be markedly smaller than that for the generation of the discrete solution since almost throughout the corresponding discretization methods involve iterative processes which may be started with much better guesses in the second pass of the method.

Thus it may be hoped that Zadunaisky's technique and its refinements may make it possible to obtain realistic error estimates for the results of discretization methods in a wide class of problems at an expense which is not prohibitively high.

REFERENCE

1966

Zadunaisky, P. E., Title unknown, in Proc. IAU Symp. No. 25, Thessaloniki.
In this paper we consider one-step discretizations for stiff differential equations. We study the problem of the convergence of the approximate solution, obtained by means of a discretization method, to the solution of the differential equation. We introduce the stiffness of the differential equation by means of a small positive parameter as in the theory of singular perturbation problems. Then we ask to what extent the convergence properties are independent of the stiffness. This question is important in view of the use of "large" stepsizes, i.e. stepsizes not determined by the stiffness, but only by an accuracy requirement. In order to do so, we decompose a solution of the differential equation in a slowly varying, regular component and a rapidly decaying, singular component. A similar decomposition is proved for the solution of a recursion.

In our error-analysis we then compare the slowly varying component of the solution of the differential equation with the corresponding component of the discretization. This requires a new concept of consistency and, as a consequence, a new definition of stability. It turns out that bad consistency properties may force small stepsizes, even if the stability properties of the method are very good.

Usually the solution to be obtained will consist of a regular component and a singular one. Then the initial-value is decomposed into an initial-value for the regular component and an initial-
value for the singular one. These decompositions might be different for the case of a differential equation and a discretization. The decomposition errors describe this situation. This theory enables us to give a realistic description of the global error if "large" stepsizes are used.

2. DECOMPOSITION OF A SOLUTION

Let \( R \) denote a two dimensional real linear space. Let a basis have been given in \( R \). Any element \( x \in R \) is denoted with respect to this basis by

\[
    x = ( E_1 x, E_2 x )^T
\]

with \( E_1 \) and \( E_2 \) projections on the first and the second basis vector respectively. Let \( \| \cdot \| \) denote a norm on \( R \).

Consider the differential equation

\[
    \frac{dx}{dt} = [ \Lambda(\varepsilon) + B(t) ] x + g(t)
\]

with \( g(t) \in R, \Lambda(\varepsilon), B(t) \) \( 2 \times 2 \) matrices and \( g \) and \( B \) defined and continuous for all \( t \in IR \) (the set of real numbers). \( \Lambda(\varepsilon) \) denotes the \( 2 \times 2 \) matrix

\[
    \Lambda(\varepsilon) = \begin{pmatrix}
        \lambda/\varepsilon & 0 \\
        0 & \mu
    \end{pmatrix}, \quad \varepsilon > 0, \lambda < 0.\tag{3}
\]

It is assumed that for all \( t \in IR \) and \( i, j = 1, 2 \) \( |B_{ij}(t)| \leq m \).

So the differential equation (2) is stiff if \( \varepsilon \lambda j \mu \) is sufficiently small. Usually a solution is required on a segment \([0, T]\). Our assumptions require the differential equation to have been defined for all \( t \in IR \), because the required decomposition of a solution is proved by means of an unbounded domain of definition. As an example, consider the case \( B = 0 \) in (2). Then, if \( g \) is slowly varying with respect to \( t \), any rapidly decaying component is given by a multiple of

\[
    ( \exp(\lambda t/\varepsilon), 0 )^T \tag{4}
\]

If \( B \neq 0 \), it is to be expected that a rapidly decaying component still exists and that such a component, a solution of the homogeneous part of (2), is close in some sense to a multiple of (4). By the variation of constants formula we have:

\[
    E_1 x(t) = \exp(\lambda t/\varepsilon)E_1 x(0) + \int_0^t \exp(\lambda(t-s)/\varepsilon)E_1 B(s)x(s)ds \tag{5}
\]

\[
    E_2 x(t) = \exp(\mu t)E_2 x(0) + \int_0^t \exp(\mu(t-s))E_2 B(s)x(s)ds. \tag{6}
\]

Now, if \( x \) is a rapidly decaying solution with a growth character approximately given by the growth character of (4), we have for \( t \to \infty \)
in (6):
\[ E_2 x(0) = - \int_0^\infty \exp(-us) E_2 B(s) x(s) ds \] (7)

With this formula we eliminate \( E_2 x(0) \) in (6). There results an integral equation on \([0, \infty)\). We then look for a suitable Banach space, such that in this space the integral equation has a unique solution. The existence of a unique solution is shown by a slightly modified fixed-point theorem in a pseudo-metric space, cf. [Collatz (1964)]. By a similar process, first for \( t \in (-\infty, 0] \), then for \( t \in \mathbb{R} \), an integral equation involving \( g \) is constructed for the slowly varying components. Our process is not new, cf. [Coddington and Levinson (1955)]. In order to summarize our results, we start with some definitions.

Let \( C(\alpha, \beta) \) denote the Banach space of continuous mappings \( x \) of \( \mathbb{R} \) into \( \mathbb{R} \) with norm

\[ \|x\|_{\alpha, \beta} = \max \{1, u, b, \|\exp(-\alpha t)x(t)\|, 1, u, b, \|\exp(-\beta t)x(t)\| \} \] (8)

Let \( C(\gamma) \) denote the Banach space of continuous mappings \( x \) of \([0, \infty)\) into \( \mathbb{R} \) with norm

\[ \|x\|_{\gamma} = \max \{1, u, b, \|\exp(-\gamma t)x(t)\| \} \] (9)

Put

\[ M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \] (10)

We assume that for all \( \varepsilon \in (0, \varepsilon_0] \) the matrix \( \Lambda(\varepsilon) + M \) has two different real eigenvalues \( \lambda_1(\varepsilon) \) and \( \lambda_2(\varepsilon) \), \( \lambda_1(\varepsilon) < \lambda_2(\varepsilon) \). Moreover, assume \( g \in C(\alpha, \beta) \) for some \( \alpha < \lambda_1(\varepsilon), \lambda_2(\varepsilon) \) and some \( \beta \), majorizing the eigenvalues of \( \Lambda(\varepsilon) + |M|, |M| = \left| \begin{pmatrix} M_{11} \\ M_{12} \end{pmatrix} \right| \). We then have for \( \varepsilon \in (0, \varepsilon_0] \):

Any solution \( x(\cdot, d, \varepsilon) \) of (2), \( x(0, d, \varepsilon) = d \) the initial-value, may be written for \( t \geq 0 \) as

\[ x(\cdot, d, \varepsilon) = u(\cdot, \xi, \varepsilon) + v(\cdot, \eta, \varepsilon) \] (11)

The **regular component** \( u(\cdot, \xi, \varepsilon) \) is a solution of the differential equation (2), and as such it is uniquely determined by

\[ E_2 u(0, \xi, \varepsilon) = \xi, u(\cdot, \xi, \varepsilon) \in C(\alpha, \beta) \] (12)

The **singular component** \( v(\cdot, \eta, \varepsilon) \) is a solution of the homogeneous part of (2), and as such it is uniquely determined by

\[ E_1 v(0, \eta, \varepsilon) = \eta, v(\cdot, \eta, \varepsilon) \in C(\gamma) \] (13)

for \( \gamma \in (\lambda_1(\varepsilon), \lambda_2(\varepsilon)) \). Moreover, for \( \delta(\varepsilon) \) and \( \mathcal{J}(\varepsilon) \) depending only on \( \varepsilon \), we have

\[ E_1 u(0, \xi, \varepsilon) = \delta(\varepsilon) \xi + E_1 u(0, 0, \varepsilon), E_2 v(0, \eta, \varepsilon) = \mathcal{J}(\varepsilon) \eta \] (14)

Hence, \( d \) and \( (\eta, \xi) \) are related by

\[ \begin{bmatrix} 1 & \delta(\varepsilon) \\ \mathcal{J}(\varepsilon) & 1 \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = d - \begin{bmatrix} E_1 u(0, 0, \varepsilon) \\ 0 \end{bmatrix} \] (15)
and this equation is uniquely solvable since \( \mathcal{A}(\varepsilon), \mathcal{J}(\varepsilon) \) are bounded uniformly in \( \varepsilon \) and \( |\mathcal{A}(\varepsilon)\mathcal{J}(\varepsilon)| < 1 \).

These results establish the decomposition of a solution. We still need the "smoothness" of a regular component. We have found:

If for \( k = 0, 1, 2, \ldots, N \) the \( k \)-th derivative of \( g \) is in \( C(\alpha, \beta) \), and if the \( k \)-th derivative of \( B_{i,j} \), \( i, j = 1, 2 \), is bounded uniformly on \( \mathbb{R} \), then the \( k \)-th derivative \( \mathcal{L}^{k}\psi \) \((k = 0, 1, 2, \ldots, N \) ) of any regular component is in \( C(\alpha, \beta) \) with a norm independent of \( \varepsilon \).

Finally we consider the case that the differential equation (2) is only defined for \( t \in [-a, T + a], T > 0, a > 0 \). Then, by considering the modified differential equation

\[
\frac{dx}{dt} = \mathcal{A}(\varepsilon)x + \psi'(t)\mathcal{B}(\psi(t))x + \psi'(t)g(\psi(t))
\]

with \( \psi \) a \( N+1 \) times continuously differentiable function satisfying

\[
\psi(t) = t \text{ for } t \in [0, T], \quad \psi'(t) \in [0, 1] \text{ for } t \in \mathbb{R}, \quad \lim_{t \to -\infty} \psi(t) = -a, \quad \lim_{t \to +\infty} \psi(t) = T + a
\]

we may apply our theory, since the differential equation is now defined on all of \( \mathbb{R} \). Clearly, the decomposition now depends on the choice of \( \psi \), but it can be shown that different \( \psi \) cause differences of order \( O(\varepsilon^N) \), \( \varepsilon \to 0 \), in the components restricted to \([0, T]\). This agrees with the results of the theory of singular perturbation problems, cf. [Wasow(1965)].

I have also extended the above considerations to non-linear differential equations in \( n \)-dimensional real linear space.

3. RECURSIONS

In this section we consider the decomposition of the solution of the recursion

\[
x_i = A_{i-1}x_{i-1} + B_{i-1} = \begin{bmatrix} a_{i-1} & b_{i-1} \\ c_{i-1} & d_{i-1} \end{bmatrix} x_{i-1} + B_{i-1}
\]

(17)

It is assumed that the recursion has been defined for all \( i \in \mathbb{Z} \) (the set of integers); since we shall use our results when the recursion is obtained by means of a discretization, this is no real restriction in view of the modification process sketched in the previous section.

By \( \{x\} \) we denote the sequence of vectors in \( \mathbb{R} \)

\[
\cdots , x_{-2} , \cdots , x_{-1} , x_0 , x_1 , \cdots , x_i , \cdots
\]

By \( [x] \) we denote the sequence of vectors in \( \mathbb{R} \)

\[
x_0 , x_1 , \cdots , x_i , \cdots
\]
Let $X(\rho, \nu)$ denote the Banach space of sequences $\{x\}$ with the norm
\[ \|\{x\}\|_{\rho, \nu} = \max_{i \geq 0} \{ \text{l.u.b. } \|\rho^{-i} x_i\|, \text{l.u.b. } \|\nu^{-i} x_i\| \} \] (18)
and let $X(\gamma)$ denote the Banach space of sequences $\{x\}$ with norm
\[ \|\{x\}\|_{\gamma} = \text{l.u.b. } \|\gamma^{-i} x_i\| \] (19)

Now assume that for all $i \in \mathbb{Z}$:
\[ |a_i| \leq \sigma_1, \ |b_i| \leq \sigma_1, \ |c_i| \leq \sigma_2, \ d_i \geq \sigma_2 > \sigma_1 . \]

Put
\[ M(\alpha) = \begin{pmatrix} \sigma_1 & \alpha \sigma_1 \\ -\alpha \sigma_2 & \sigma_2 \end{pmatrix} , \] (20)

and assume that for all $\alpha \in [0,1]$ the matrix $M(\alpha)$ has two different real eigenvalues, for $\alpha = 1$ denoted by $\lambda_1, \lambda_2, \lambda_1 < \lambda_2$. Also assume $\{g\} \in X(\rho, \nu)$ with $\rho \in (\lambda_1, \lambda_2)$ and $\nu$ majorizing the eigenvalues of $|M(1)|$. Then:

Any solution $\{x(d)\}, x_0(d) = d$, of the recursion (17) may be written uniquely as
\[ \{x(d)\} = \{u(\xi)\} + \{v(\eta)\} . \]

Here $\{u(\xi)\}$ is the restriction to $i \geq 0$ of the regular component $\{u(\xi)\}$; $\{u(\xi)\}$ is in $X(\rho, \nu)$ and it is a solution of (17). Moreover, $E_1 u_0(\xi) = \xi$. These properties uniquely determine $\{u(\xi)\}$, and hence $\{u(\xi)\}$, for given $\xi \in \mathbb{R}$.

The singular component $\{v(\eta)\}$ is in $X(\gamma), \gamma \in (\lambda_1, \lambda_2)$, and it is a solution of the homogeneous part of (17). Moreover $E_1 v_0(\eta) = \eta$. These properties uniquely determine $\{v(\eta)\}$ for given $\eta \in \mathbb{R}$.

We also have:
\[ E_1 u_0(\xi) = \delta_0 \xi + E_1 u_0(0), \ E_2 v_0(\eta) = \delta_0 \eta \]
with $|\delta_0| < 1$. Hence $(\eta, \xi)^T$ and $d$ are related by
\[ \begin{pmatrix} 1 & \delta_0 \\ \delta_0 & 1 \end{pmatrix} \begin{pmatrix} \eta \\ \xi \end{pmatrix} = d - \begin{pmatrix} E_1 u_0(0) \\ 0 \end{pmatrix} \] (21)

The analogy with the results of the previous section is obvious. Similar results have been obtained by Van der Sluis$^\dagger$.

It is possible to give an upper bound for a regular component. In fact, if $\{r(\xi)\}$ denotes the regular component of the recursion
\[ x_i = M(1)x_{i-1} + \rho^{-i-1} \|\{g\}\|_{\rho, \nu} (1,-1)^T \] (22)

\[ ^\dagger \text{A. van der Sluis, Estimating the Solution of Slowly Varying Recursions, to appear. A preprint may be obtained on request (Mathematical Institute, University of Utrecht).} \]
determined by \( E_2 r_0(\xi) = \xi \), then we have for \( i \leq 0 \) (and not for \( i \geq 0 \)):
\[
|E_j u_i(\xi)| \leq E_j r_i(|\xi|), \quad j = 1, 2.
\]
For \( i \geq 0 \), \( \{u(\xi)\} \) (the restriction of \( \{u(\xi)\} \) to \( i \geq 0 \)) is majorized pointwise and coordinateswise by the solution ( \( i > 0 \) ) of the recursion
\[
x_i = |M(1)| x_{i-1} + \nu^{i-1} ||g|| \rho \nu (1, 1)^T
\]
with initial-value \( r_0(|\xi|), \{r(|\xi|)\} \) as above. The calculation of these upperbounds is possible since \( M(1) \) and \( |M(1)| \) are constant matrices.

It should be noted that the above theory also applies if the recursion is non-linear and implicit (such recursions are frequently met as discretizations).

4. CONSISTENCY

In this section we consider the linear differential equation in \( \mathbb{R}^2 \)
\[
\frac{dz}{dt} = A(t, \varepsilon)z + b(t, \varepsilon), \quad \varepsilon \in (0, \varepsilon_0].
\]
We suppose that a transformation \( x(t) = \xi(t, \varepsilon)z(t) \) exists which transforms (24) into such a differential equation, that the results of section 2 hold. Moreover, we assume for all \( t \in \mathbb{R} \) and all \( \varepsilon \in (0, \varepsilon_0] \) that
\[
\| \frac{d^k}{dt^k} F(t, \varepsilon) \|, \quad \| \frac{d^k}{dt^k} F^{-1}(t, \varepsilon) \|
\]
are uniformly bounded in \( t \) and \( \varepsilon, k = 0, 1, 2, \ldots, N \). In fact, this property has to be required for \( t \in [-a, T + a] \) only; the modification process of section 2 (by means of a suitable function \( \psi \)) then may be applied to modify (24) into such a differential equation that this property holds for all \( t \in \mathbb{R} \).

Let \( x(\cdot, \xi, \varepsilon) \) denote a regular component of the transformed differential equation. Hence \( x(\cdot, \xi, \varepsilon) \in C(\alpha, \beta) \) for suitable \( \alpha \) and \( \beta \) (cf. section 2). It can be shown that \( \alpha \) and \( \beta \) may be chosen independently of \( \varepsilon \). Put
\[
\mathcal{F}(\xi) = \{z(\cdot, \xi, \varepsilon) | z(t, \xi, \varepsilon) = F^{-1}(t, \varepsilon)x(t, \xi, \varepsilon) \text{ for all } t \in \mathbb{R} \}.
\]
Any element of \( \mathcal{F}(\xi) \) is in \( C(\alpha, \beta) \) and any element of \( \mathcal{F}(\xi) \) is a solution of (24). Let
\[
z_i = D_i (\varepsilon, h) z_{i-1} + h b_i (\varepsilon, h)
\]
denote a recursion obtained as the discretization of the differential equation (24) with stepsize \( h \). Put
\[
\delta_i (\xi, \varepsilon, h) = \frac{1}{h} \{z(ih, \xi, \varepsilon) - D_i (\varepsilon, h) z((i-1)h, \xi, \varepsilon) - h b_i (\varepsilon, h)\}
\]
the local discretization error in \( t_i = (i-1)h \), \( z(\cdot, \xi, \varepsilon) \in \mathcal{F}(\xi) \). By \( \{\delta(\xi, \varepsilon, h)\} \) we denote the sequence of local discretization errors.

**Definition I**  The method (26) is called \( R \)-consistent for the differ-
consistency and stability for one-step discretizations

The method is called R-consistent and of order $p$, if $c(h) = O(h^p)$.

This definition differs from the usual one, cf. [Henrici (1962)], by the restriction to regular components and the uniformity with respect to $\varepsilon$. The threshold $k$ relates the maximum stepsize to the stiffness, i.e., a desirable situation is a constant, non-zero threshold.

We defined R-consistency for explicit methods. The explicit form is not essential, but because of the uniformity with respect to $\varepsilon$ the choice of a standard form for the recursions obtained by discretization is important. E.g., the recursion (26) and the recursion

$$z_i = \left(1 + \frac{h}{\varepsilon}\right) D - \frac{h}{\varepsilon} z_{i-1} - \frac{h}{\varepsilon} z_i + \frac{1+h}{\varepsilon} b_{i-1}(\varepsilon, h)$$

yield the same approximation, but whereas (26) may be R-consistent with a constant threshold, (28) is never R-consistent with a constant threshold.

It is easily proved that consistency and R-consistency with constant threshold yield the same order of accuracy for the method

$$z_i - z_{i-1} - \alpha h f(t_i, z_{i-1}) - \beta h f(t_i, z_i) = 0$$

$\alpha + \beta = 1$, and $\alpha, \beta$ constants, cf. [Liniger and Willoughby (1970)]. If the method is exponentially fitted at the eigenvalue with the larger negative real part (we apply the method to (24)), then it is R-consistent with constant threshold and of order 1. Note that the exponentially fitted method is consistent and of order 2.

The method

$$z_i = z_{i-1} + [I - hJ(t_i-1, z_{i-1})]^{-1} f(t_i, z_{i-1})$$

where $J(t, z)$ denotes the Jacobian matrix of $f$ evaluated at $(t, z)$, is R-consistent and of order 1 with threshold $k(\varepsilon) = c. \varepsilon, c$ a constant, if applied to (24); e.g., the threshold is not a constant if the method is applied to the scalar differential equation

$$\frac{dz}{dt} = [- z + \exp(-t)]/\varepsilon,$$

the regular solution of which is given by $\exp(-t)/(1-\varepsilon)$.

5. Stability

The order of accuracy, as defined in the previous section, makes sense if we know how the global error is related to the local error. We make some assumptions, necessary for our stability definition.

The recursion (26) is first transformed by

(i) $\{\delta(\xi, \varepsilon, h)\} \in X(\exp(\alpha h), \exp(\beta h))$
(ii) $\text{l.u.b.} \|\delta(\xi, \varepsilon, h)\|_{\exp(\alpha h), \exp(\beta h)} \equiv c(h), \lim_{h \to 0} c(h) = 0.$
\[ x_i = \mathcal{L}(t_i, \varepsilon) z_i \]  
(\(\mathcal{L}(t, \varepsilon)\) as in (25)). Then we assume that for all \(\varepsilon \in (0, \varepsilon_0]\) and \(h \in (0, h_0(\varepsilon)]\) regular and singular components exist. Let \(\{x(\xi, \varepsilon, h)\}\) denote the regular component determined by \(E_2 x_0(\xi, \varepsilon, h) = \xi\). Then \(\{z(\xi, \varepsilon, h)\}\) defined by

\[ z_i(\xi, \varepsilon, h) = \mathcal{L}^{-1}(t_i, \varepsilon) x_i(\xi, \varepsilon, h), \quad i \in \mathbb{Z} \]

is called a regular component of (26). Moreover for some \(\rho(\varepsilon, h)\) and \(\nu(\varepsilon, h)\) (cf. section 3) we have \(\{z(\xi, \varepsilon, h)\} \in X(\rho(\varepsilon, h), \nu(\varepsilon, h))\).

**Definition II** The discretization, given by (26), is called **R-stable with stability threshold \(k\), if for all \(\varepsilon \in (0, \varepsilon_0]\), \(h \in (0, k(\varepsilon)]\), \(\xi \in \mathbb{R}\) and all \(\{b(\varepsilon, h)\} \in X(\rho(\varepsilon, h), \nu(\varepsilon, h))\):

(i) \(X(\exp(\alpha h), \exp(\beta h)) \subset X(\rho(\varepsilon, h), \nu(\varepsilon, h))\) with \(\alpha\) and \(\beta\) belonging to (24).

(ii) \(\nu(\varepsilon, h) \leq 1 + hl\) for some constant \(l\).

(iii) \(\|z(\xi, \varepsilon, h)\| \leq C_1 |\xi| + C_2 \|b(\varepsilon, h)\|\) for non-negative constants \(C_1\) and \(C_2\).

This stability definition is applied with the sequence of **global errors** \(z_i(\xi, \varepsilon, h) - z(t_i, \xi, \varepsilon)\) in stead of \(\{z(\xi, \varepsilon, h)\}\) and the local discretization error in stead of \(\{b(\varepsilon, h)\}\). Then we easily obtain the convergence result for regular components:

**Theorem** If the discretization (26) is **R-consistent** (and of order \(p\)) with threshold \(k\), and if it is **R-stable with stability threshold \(k\)**, then

\[ 1. u. b. \quad \| z_i(\xi, \varepsilon, h) - z(t_i, \xi, \varepsilon) \| \leq c(h) (1 + hl)^i, \quad i \geq 0 \]

with \(c(h) = o(1)\) (\(O(h^p)\)) for \(h \in (0, k(\varepsilon)]\) and \(h \to 0\).

Note that we obtain convergence only for \(t \geq 0\), due to the requirement \(\nu(\varepsilon, h) \leq 1 + hl\). We are not interested in \(t < 0\), but convergence for \(t < 0\) might be obtained also.

We mention here that the Euler backwards method and the exponentially fitted method (29) are **R-stable with constant threshold**, if applied to (24). These results may be obtained by applying the estimates given in section 3 for a regular component. We may even develop the global error in an asymptotic series in \(h\), cf. [Stetter (1965)]. We have for \(i \geq 0\):

\[ z_i(\xi, \varepsilon, h) - z(t_i, \xi, \varepsilon) = h(\alpha - \frac{1}{2}) e(t_i, \varepsilon) + r_i(\varepsilon, h) \]

where \(\cdot e(\cdot, \varepsilon)\) is the regular solution of

\[ \frac{du}{dt} = A(t, \varepsilon) u + z''(t, \xi, \varepsilon) \]
CONSISTENCY AND STABILITY FOR ONE-STEP DISCRETIZATIONS

267
determined by $E_2L^{-1}(0,\epsilon)e(0,\epsilon) = 0$ and where

$$
l.u.b. \| r_i(\epsilon,h) \| (1 + hl)^{-i} = O(h^2), \ h \to 0.
$$

For the Euler backwards method we have $a=0$; for the exponentially fitted method $a$ depends on the point at which the fit is made.

For the method (30), applied to the homogeneous part of (24), the global error is $O(h)$ uniformly in $\epsilon, \epsilon \in (0, \epsilon_0], h \in (0, \ h_0), \ h_0$ a constant. Here the situation is much more complicated than for the method (29), since the method (30) need not to be $R$-consistent with constant threshold. Therefore a somewhat sharper stability result is required. The necessary calculations are very awkward, but the tools for these calculations may be found in section 3.

Now we consider an example of an inhomogeneous differential equation, that satisfies our assumptions. The differential equation is:

$$
\frac{dz}{dt} = \frac{1}{\epsilon} \begin{pmatrix} -1 & 0 \\ \sin(t) & 0 \end{pmatrix} z + \frac{1}{\epsilon} \begin{pmatrix} -1 \\ \sin(t) \end{pmatrix} \exp(\frac{1}{2}t).
$$

With

$$
L(t,\epsilon) = \begin{pmatrix} 1 & 0 \\ \sin(t) & 1 \end{pmatrix}
$$

the transformation $x(t) = L(t,\epsilon)z(t)$ transforms (33) into

$$
\frac{dx}{dt} = \begin{pmatrix} -1/\epsilon & 0 \\ \cos(t) & 0 \end{pmatrix} x + \begin{pmatrix} -\exp(\frac{1}{2}t)/\epsilon \\ 0 \end{pmatrix}
$$

If the method (30) is applied to (33), and if the transformation (32) is applied, there results:

$$
x_i = \begin{pmatrix} (1 + h/\epsilon)^{-1} \\ \sin(t_i) - \sin(t_{i-1}) \end{pmatrix} x_{i-1} - \frac{h}{\epsilon} \begin{pmatrix} (1 + h/\epsilon)^{-1} \\ \sin(t_i) - \sin(t_{i-1}) \end{pmatrix} \exp(\frac{1}{2}t).
$$

The second coordinate of the inhomogeneous term is not bounded uniformly in $\epsilon$; hence any regular component is not bounded uniformly in $\epsilon$. Thus the global error is not bounded uniformly in $\epsilon$ and $h$, $\epsilon \in (0, \epsilon_0], h \in (0, h_0], h_0$ a constant. So "large" stepsizes yield useless results, although the method is very stable.

This bad behaviour of the method (30) is not to be expected for all differential equations; the results of my thesis may be used to predict whether or not such a bad behaviour might occur.
6. DECOMPOSITION ERRORS

In general an initial-value is given such that the resulting solution contains a singular component. For easier presentation we shall restrict ourselves in this section to the differential equation (24) with \( b(t, \varepsilon) = 0 \) for all \( t \in \mathbb{R} \) and all \( \varepsilon \). The same assumptions as in the previous sections should hold for (24) (now homogeneous) and its discretization (26) (also homogeneous). Moreover we assume \( \ell(t, \varepsilon) = 1 \) for all \( t \) and \( \varepsilon \). Then (cf. section 2), an initial-value \( \delta \in \mathbb{R} \) at \( t = 0 \) determines \( \xi, \eta \in \mathbb{R} \) by

\[
\begin{bmatrix}
1 & \delta(\varepsilon) \\
J(\varepsilon) & 1
\end{bmatrix}
\begin{bmatrix}
\eta \\
\xi
\end{bmatrix} = \delta
\] (35)

while the regular component of the solution is determined by \( \xi \) and the singular one by \( \eta \). For the recursion we have the corresponding equation (cf. section 3)

\[
\begin{bmatrix}
1 & \delta^0(\varepsilon, h) \\
\sigma^0(\varepsilon, h) & 1
\end{bmatrix}
\begin{bmatrix}
\eta' \\
\xi'
\end{bmatrix} = \delta
\] (36)

In the notation of the sections 2 and 3, we have

\[
x(t_i, \delta, \varepsilon) = x_i(\delta, \varepsilon, h) = [u(t_i, \xi, \varepsilon) - u_i(\xi', \varepsilon, h)] + [v(t_i, \eta, \varepsilon) - v_i(\eta', \varepsilon, h)].
\] (37)

In many instances we shall have for \( t_i \) sufficiently large in relation to \( \varepsilon \):

\[
\|v(t_i, \eta, \varepsilon)\| + \|v_i(\eta', \varepsilon, h)\| \ll \varepsilon,
\] (38)

eps the required tolerance for \( t = t_i \). Then the error is given in good approximation by

\[
u(t_i, \xi, \varepsilon) - u_i(\xi', \varepsilon, h)
\] (39)

and an upperbound for this error may be obtained, if the discretization is R-stable and if a majorant for \( |\xi - \xi'| \) is known. Now \( \xi - \xi' \) is determined by the decomposition errors \( \delta(\varepsilon) - \delta^0(\varepsilon, h) \) and \( J(\varepsilon) - J^0(\varepsilon, h) \).

We have in the notation of the previous sections with \( x \) in stead of \( z \) (because of \( \ell(t, \varepsilon) = 1 \)):

\[
\delta(\varepsilon) - \delta^0(\varepsilon, h) = E_1 x(0, 1, \varepsilon) - E_1 x_0(1, \varepsilon, h)
\] (40)

\( x(., 1, \varepsilon) \) and \( \{x(1, \varepsilon, h)\} \) regular components. Hence \( \delta(\varepsilon) - \delta^0(\varepsilon, h) \) is obtained by means of R-consistency and R-stability without any new difficulty. To obtain \( J(\varepsilon) - J^0(\varepsilon, h) \) we have the following duality theorems:

Any regular component \( x(., \xi, \varepsilon) \) of the (adjoint) differential equation

\[
\frac{dx}{dt} = A^T (-t, \varepsilon) x
\] (41)
satisfies $E(x(0,\xi,\varepsilon)) = J(\varepsilon)\xi$.

Any regular component $\{x(\xi',\varepsilon,h)\}$ of the (adjoint) recursion

$$x_i = D^{-1}_i(\varepsilon,h) x_{i-1} \quad \text{(42)}$$

satisfies $E(x_0(\xi',\varepsilon,h)) = J'(\varepsilon,h)\xi'$.

Hence, to obtain $J(\varepsilon) - J_0(\varepsilon,h)$ we may proceed as for $\delta(\varepsilon) - \delta_0(\varepsilon,h)$, but now with the adjoint differential equation and the adjoint recursion. E.g., for the Euler backwards method we have

$$D_i(\varepsilon,h) = [I - hA(t_{i+1},\varepsilon)]^{-1} \quad \text{(43)}$$

Hence (42) is given by

$$x_i = [I - hA(t_{i-1},\varepsilon)]^{-T} x_{i-1} \quad \text{(44)}$$

But this recursion also results if the method (30) is applied to (41). Hence, by the previously mentioned results:

$$\text{l.u.b.} \ |J(\varepsilon) - J_0(\varepsilon,h)| = O(h) \ , \ h \rightarrow 0$$

$$\varepsilon \in (0,\varepsilon_0]$$

and

$$\text{l.u.b.} \ |\delta(\varepsilon) - \delta_0(\varepsilon,h)| = O(h) \ , \ h \rightarrow 0$$

$$\varepsilon \in (0,\varepsilon_0]$$

for $h \in (0,\varepsilon_0], \varepsilon_0$ a constant. From this result it follows that the global error for the Euler backwards method (but also for the method (30)), if applied to (24) with $b(\cdot,\varepsilon) = 0$, yields a global error given by $O(h)$, uniformly in $\varepsilon$, and the difference of some rapidly decaying components. Thus, for $t_i$ sufficiently large (in practice outside the boundary layer) the global error is in good approximation $O(h)$, even if "large" stepizes are used.

For more examples I may refer to my thesis.

7. DISCUSSION

It is believed that the theory, summarized above, enables a realistic approach of stiff differential equations from the theoretical point of view, i.e. our theory describes what happens if the stepsize is controlled by the behaviour of the "smooth" solution rather than by the small parameter (stiffness). Therefore, in our theory the stepsize may very well be "large" with respect to the Lipschitz constant of the right-hand side of the differential equation. Some simple examples illustrate this theory. For more complicated methods, e.g. the higher order methods of [Liniger and Willoughby(1970)], or the Runge-Kutta methods of [Van der Houwen(1972)] an investigation of R-consistency and R-stability with realistic threshold seems technically difficult. For multistep methods also there are quite a few technical difficulties.
Acknowledgement

I am indebted to Prof. A. van der Sluis for many hours of stimulating discussion and many critical remarks.

REFERENCE

1973

BIBLIOGRAPHY

1973


271
1973 (cont.)


Lim, H. C., "On kinetics behavior at high enzyme concentrations," AIChE J. 19, pp. 659-


1973 (cont.)


Ström, T., "Practical error estimates for repeated Richardson extrapolation schemes," BIT 13, pp. 196-205.


1972


1972 (cont.)


Gautschi, W., "Numerical aspects of recurrence relations" (German), Computing 9, pp. 107-126.


1972 (cont.)


IBM Journal of Research and Development 16 (July), Special Issue on New Techniques in the Mathematics of Numerical Computation.

IEEE Proceedings 60 (Jan.) Special Issue on Computers in Design.


Jamshidi, M., "A near-optimal controller for cold-rolling mills," Intern. J. Control 16, pp. 1137-


1972 (cont.)

Pouzo, P. J., and Wax, N., "Relaxation oscillations, parasitics, and singular perturbations," IEEE Trans. CT-19, pp. 623-


Schneider, D. R. (et al), "On a mechanism for autocatalysis," Chem. Eng. Sci. 27, pp. 895-


1971


1971 (cont.)


BIBLIOGRAPHY

1971 (cont.)


IEEE Transactions on Circuit Theory CT - 18 (Jan.) Special Issue on Computer-Aided-Design.


1971 (cont.)


1971 (cont.)


1970


Aris, R., "Chemical kinetics and the ecology of mathematics," Amer. Scientist 58, pp. 419-


Daniel, J.W., and Moore, R.E., Computation and Theory in Ordinary Differential Equations, Freeman, San Francisco.
1970 (cont.)


1970 (cont.)


1969 (cont.)


Il'in, A. M., "Differencing scheme for a differential equation with a small parameter affecting the highest derivative" (Russian), Mat. Zametki 6, pp. 237-248 (Journal is translated as: Mathematical Notes of the Academy of Sciences USSR).


Mentzen, B., "Use of electronic computers in problems of basic and applied kinetics" (French), Chemie et Ind. - Genie Chim. 109, pp. 37-53.


1969 (cont.)


1968


1968 (cont.)


(A) Distefano, G. P., "Mathematical modeling and numerical integration of multicomponent batch distillation equations," AIChE J. 14, pp. 130-


Garfinkel, D., "A machine-independent language for the simulation of complex chemical and biochemical systems," Computers in Biomedical Research 2, pp. 31-44.


1968 (cont.)


Luss, D., and Amundson, N. R., "Stability of batch catalytic fluidized beds," AIChE J. 14, pp. 211-


1968 (cont.)


1967


Gautschi, W., "Computational aspects of three-term recurrence relations," SIAM Rev. 9, pp. 24-82 (Extensive bibliography).


Heinekin, F.G. (et al), "On the mathematical status of the pseudo-steady state hypothesis of biochemical kinetics," Math. Biosciences 1, pp. 95-

Hicks, J.S., and Wei, J., "Numerical solution of parabolic partial differential equations with two point boundary conditions by use of the method of lines," JACM 14, pp. 549-


1967 (cont.)


1967 (cont.)


1966


Glasmacher, W., and Dietmar, S., Implicit Runge-Kutta Formulas, Westd. Verlag, Köln.


1966 (cont.)


Walsh, J. (Editor), Numerical Analysis: an Introduction (IMA Conf.) Thompson Book Co., Washington, D.C.


1965


1965 (cont.)

Campbell, E. S., "Solution of the hydrodynamic equations for laminar time-independent flames with arbitrarily large deviations from the kinetic state," Quart. J. Combustion Institute 9, pp. 43-52.

Coppel, W., Stability and Asymptotic Behavior of Differential Equations, Heath, Boston.


Liskovets, O. A., "The method of lines" (review), Differential Equations 1, pp. 1308-


BIBLIOGRAPHY

1965 (cont.)


Wasow, W., Asymptotic Expansions for Ordinary Differential Equations, Interscience, New York.


1964

(A) Butcher, J. C., "Implicit Runge-Kutta processes," Math. Comp. 18, pp. 50-64.

(B) Butcher, J. C., "Integration processes based on Radau quadrature formulas," Math. Comp. 18, pp. 233-244.


Walter, W., Differential- and Integral- Inequalities (German), Springer, Berlin (English translation in 1970).


Bowen, J. R. (et al), "Singular perturbation refinement to quasi-steady state approximation in chemical kinetics," Chem. Eng. Sci. 18, pp. 177-


1963 (cont.)


1962


Kreiss, H.-O., "On the definition of stability for difference equation approximations to partial differential equations" (German), BIT 2, pp. 153-181.

1962 (cont.)


1961


1961 (cont.)


1960


1959

1959 (cont.)


1958


1957


1956


1955


(A) Crandall, S. H., "Implicit vs explicit recurrence formulas for the linear diffusion equation," JACM 2, pp. 42-49.

1955 (cont.)


Rubinoff, M., "Digital computers for real-time simulation," JACM 2, pp. 186-204.

1954

Deissler, R.G., "Heat transfer and fluid friction for fully developed turbulent flow of air and supercritical water with variable fluid properties," ASME Trans. 76, pp. 73-85.


1953


1952


1949


1947


1942


1937


1929

Perron, O., "On stability and asymptotic convergence of systems of integral and differential equations" (German), Math. Z. 29, pp. 129-160.

1927


1925

1911

Perron, O., "On linear difference equations" (German), Acta. Math. 34, pp. 109-137.

1910


1909

Perron, O., "On a theorem of Mr. Poincaré" (German), J. Reine Angew. Math. 136, pp. 17-37.

1885

AUTHOR LIST* FOR BIBLIOGRAPHY

Abbett, M. J. 65 (De Groat) 
Achs, M. J. 68 (Rhoads) 
Ackerberg, R. C. 73 (et al) 
Amundson, N. R. 68 (et al) 68 (Luss) 61 (Liu) 
Anderson, W. H. 60 (et al) 
Andria, G. D. 73 (et al) 
Ansorge, R. 70 (et al) 
Aris, R. 70 65 
Aube, G. 72 (Richard) 
Avenhaus, J. 71 
Axelsson, O. 72 69 
Babuska, I. 68 P 66 (et al) 
Bahvalov, N. S. 69 
Bailey, P. B. 68 (et al) 
Ball, R. B. 60 (Anderson) 
Ball, W. E. 70 (Papian) 
Barandes, M. 68 (Berman) 
Bard, Y. 69 (et al) 
Barton, D. 71(A) (et al) 71(B) P (et al) 
Bauer, F. L. 63 P (et al) 
Baxter, D. C. 68 (Gagne) 
Becker, D. V. 68 (Berman) 
Belevitch, V. 68 
Bellman, R. 63 (et al) 
Bennett, M. M. 67 (Riley) 
Benua, R. 68 (Berman) 
Benyon, P. R. 68 S 
Berman, M. 68 (et al) 65 
Bickart, T. A. 73 (et al) 73 (Sloate) 
Biesenberger, J. A. 72 (et al) 
Birkhoff, G. 66 P, S 65 S (et al) 
Bjurel, G. 72 
Blanch, G. 52 
Blue, J. L. 70 (et al) 
Boggs, P. T. 
Bowen, J. R. 63 (et al) 
Bowers, J. C. 71 (et al) 

* 73 refers to book published in 1973, 73 P means author is editor of proceedings or collection of articles, 73 P refers to article in proceedings, 73 S refers to survey article, 73 (et al) means author is first of several authors - the others each list name of first author.
<table>
<thead>
<tr>
<th>Author</th>
<th>Year(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxer, R.</td>
<td>56 (et al)</td>
</tr>
<tr>
<td>Boyer, D. W.</td>
<td>62 (Eschenroeder)</td>
</tr>
<tr>
<td>Boyle, J. H.</td>
<td>73 (Smith, B. T.)</td>
</tr>
<tr>
<td>Branin, F. H.</td>
<td>71 (et al) 67 S</td>
</tr>
<tr>
<td>Brayton, R. K.</td>
<td>72 (A) (et al) 72 (B) (et al) 71 P 71 (A) (Hachtel) 68 67 (et al) 66 (et al)</td>
</tr>
<tr>
<td>Brennan, R. D.</td>
<td>68 P, S</td>
</tr>
<tr>
<td>Brent, R. P.</td>
<td>72</td>
</tr>
<tr>
<td>Brock, P.</td>
<td>52 (et al)</td>
</tr>
<tr>
<td>Brown, K. M.</td>
<td>68 (et al)</td>
</tr>
<tr>
<td>Broyden, C. G.</td>
<td>71 69</td>
</tr>
<tr>
<td>Brunner, H.</td>
<td>72 (A) 72 (B) 70 67</td>
</tr>
<tr>
<td>Bulirsch, R.</td>
<td>66 (A) (et al) 66 (B) (et al)</td>
</tr>
<tr>
<td>Butcher, J. C.</td>
<td>72 64 (A) 64 (B)</td>
</tr>
<tr>
<td>Byrne, G. D.</td>
<td>73 (Andria)</td>
</tr>
<tr>
<td>Cadzow, J. A.</td>
<td>70 (et al)</td>
</tr>
<tr>
<td>Calahan, D. A.</td>
<td>71 69 (et al) 68 (A)</td>
</tr>
<tr>
<td>Calahan, D. A. (cont'd)</td>
<td>68 (B) 67 67 S (Temes)</td>
</tr>
<tr>
<td>Campilongo, E. D.</td>
<td>71</td>
</tr>
<tr>
<td>Campbell, D. F.</td>
<td>53 (Hirschfelder)</td>
</tr>
<tr>
<td>Campbell, E. S.</td>
<td>66 S (et al) 65</td>
</tr>
<tr>
<td>Capiapin, R.</td>
<td>72 (Biesenberger)</td>
</tr>
<tr>
<td>Carasso, A.</td>
<td>72</td>
</tr>
<tr>
<td>Cash, J. R.</td>
<td>72</td>
</tr>
<tr>
<td>Cermak, I.</td>
<td>71 (A) 71 (B) (et al)</td>
</tr>
<tr>
<td>Certaine, J.</td>
<td>60 P</td>
</tr>
<tr>
<td>Cesari, L.</td>
<td>63</td>
</tr>
<tr>
<td>Ceschino, F.</td>
<td>66 (et al)</td>
</tr>
<tr>
<td>Chance, B.</td>
<td>70 (B) S (Garfinkel, D.)</td>
</tr>
<tr>
<td>Chance, E. M.</td>
<td>72 (Curtis)</td>
</tr>
<tr>
<td>Chang, J. S.</td>
<td>73 (et al)</td>
</tr>
<tr>
<td>Chartres, B.</td>
<td>72 (et al)</td>
</tr>
<tr>
<td>Chien, C. W.</td>
<td>60 P (Neiburger)</td>
</tr>
<tr>
<td>Chipman, F. H.</td>
<td>73 71</td>
</tr>
<tr>
<td>Christensen, H. N.</td>
<td>69 P, S</td>
</tr>
<tr>
<td>Clark, R. L.</td>
<td>72 (et al)</td>
</tr>
<tr>
<td>Clenshaw, C. W.</td>
<td>63 (et al)</td>
</tr>
<tr>
<td>Coddington, E. A.</td>
<td>55 (et al)</td>
</tr>
<tr>
<td>Cole, J. D.</td>
<td>68</td>
</tr>
</tbody>
</table>
Collatz, L.
66
60

Conley, C. C.
72 P (Brayton)

Cooke, C. H.
72

Cooke, K. L.
63 (Bellman)

Cooley, J. W.
71 (B) Hachtel

Cooper, G. J.
71
69
68

Coppel, W.
65

Corbo, V. J.
72 (et al)

Crandall, S. H.
55 (A)
55 (B)

Crank, J.
47 (et al)

Crossley, T. R.
70 S (et al)

Crosswell, D. L.
71 (Paterson)

Cryer, C. W.
73
72

Curtis, A. R.
72 (et al)

Curtiss, C. F.
53 (Hirschfelder)
52 (et al)

Dahlquist, G.
73 P
68 P
66 P
63 (A)
63 (B) P
59
56

Daniel, J. W.
70 (et al)
68 (A)
68 (B) (et al)
66

Danley, D. E.
60 (Huckaba)

Davidenko, D. F.
53

Davis, C. B.
71 (B) (Klopfenstein)

Davis, H. T.
62

Davison, E. J.
68
67

Day, J. T.
66

De Groat, J. J.
65 (et al)

Decell, H. P., Jr.
66 (et al)

Deissler, R. G.
54

Dejon, B.
67

Dennis, J. E.
68 (Brown)

Dennis, S. C. R.
60

Descloux, J.
70

De Tar, D. L. F.
67 (A)
67 (B)

Dietmar, S.
66 (Glasmacher)

Dill, C.
71 (et al)

Distefano, G. P.
68 (A)
68 (B)

Donelson, J., III
71 (et al)

Dorr, F. W.
73 S (et al)
71

Duewer, W. H.
73 (Chang)

Duffin, R. J.
69 S

Dvysens, L. N. M.
73 (Otten)
Edelson, D.  
73
Ehle, B. L.  
68
Elgerd, O. I.  
67
El-Sherif, H. H.  
68
Enright, W.  
73
72 (Hull)
Eschenroeder, A. Q.  
62 (et al)
Falkenberg, J. C.  
68
Feldstein, A.  
73 (et al)
Fellen, B. M.  
72 (Hull)
Flatt, H. P.  
68 P, S
Fletcher, N. H.  
62
Fowler, M. E.  
67 (et al)
65
63
Fox, L.  
62(A)
62(B)
57 (et al)
49 (et al)
Fox, P.  
72
Freeze, R. A.  
71
Freudenstein, F.  
63 (et al)
Gagne, R. E.  
68 (et al)
Gantmacher, F. R.  
59
Garbow, B. S.  
73 (Smith, B. T.)
Garfinkel, D.  
70(A)
70(B) S (et al)
Garfinkel, D. (cont'd)  
68
68 (Rhoads)
67
61 (et al)
Garfinkel, L.  
70(B) S (Garfinkel, D.)
Gautschi, W.  
72
67 S
61
Gavurin, M. K.  
63
Gear, C. W.  
73 (et al)
71 (Dill)
71(A)
71(B)
71(C)
71(D)
70
69 (Calahan)
68 P
67
Gelinas, R. J.  
72
Genin, Y.  
73
Giese, C.  
67
Gilbert, E. G.  
66
Glasmacher, W.  
66 (et al)
Godunov, S. K.  
64 (et al)
Goodman, R.  
73 (Feldstein)
Goodwin, E. T.  
49 (Fox, L.)
Gordon, M. K.  
73(B) Shampine)
Gourlay, A. R.  
70
Gragg, W. P.  
72 S
65
Gray, H. J., Jr.  
55
54
Green, S.  
70 (B) S (Garfinkel, D.)
Greenspan, D.  
66 P
Greenwood, R. E.  
49
Greville, T. N. E.  
69 P
Griepentrog, E.  
70
Groner, G. F.  
72 (Clark)
Grosswald, E.  
59
Grover, S. S.  
67
Guderley, K. G.  
72 (et al)
Gummel, H. K.  
70 (Blue)
Gupta, G. K.  
73 (Wallace)
Gurk, H. M.  
55
Guseman, L. F., Jr.  
66 (Decell)
Gustafsson, B.  
69
Gustavson, F. G.  
72 (A) (Brayton)
71 (A) (Hachtel)
70 (et al)
66 (Brayton)
Haas, R.  
70 (Ansorge)
Hachtel, G. D.  
72 (A) (Brayton)
71 (A) (Hachtel)
71 (B) (et al)
67 (et al)
Haines, C. F.  
69
Hakberg, B.  
70
Hall, J. G.  
62 (Eschenroeder)
Hamming, R. W.  
62
Hansen, E.  
71 (Donelson)
Hansen, K.  
65 (et al)
Harris, W. A., Jr.  
62
Hartree, D. R.  
37 (et al)
Hayashi, K.  
69
Heinekin, F. G.  
67 (et al)
Heinen, F. J.  
66 S (Campbell, E. S.)
Hemker, H. C.  
72 P (et al)
Hemker, P. W.  
72 P
Henrici, P. J.  
69
65
62
Hess, B.  
72 P (Hemker, H. C.)
Hicks, J. S.  
67 (et al)
Higgins, J. J.  
61 (Garfinkel, D.)
Hirschfelder, J. O.  
63 P, S
53 (et al)
52 (Curtiss)
Hill, D. R.  
73 (Andria)
Ho, C. W.  
73
Hodgkins, W. R.  
69 P
Hoff, E.  
68 (Berman)
Hogsett, G. R.  
71 (Branin)
Hoppensteadt, F.  
73
Houwen, P. J., van der  
72
71
Hsu, C.-C.  
72 (Guderley)
<table>
<thead>
<tr>
<th>Author</th>
<th>Year(s)</th>
<th>Journal(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huckaba, C. E.</td>
<td>60 (et al)</td>
<td></td>
</tr>
<tr>
<td>Hull, T. E.</td>
<td>73 P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>72 (et al)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>68 P, S</td>
<td></td>
</tr>
<tr>
<td>Hulme, B. L.</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Hwang, M.</td>
<td>70 S (Seinfeld)</td>
<td></td>
</tr>
<tr>
<td>Il'in, A. M.</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Ikebe, Y.</td>
<td>73 (Smith, B. T.)</td>
<td></td>
</tr>
<tr>
<td>Iri, M.</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Jain, R. K.</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Jamshidi, M.</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Jeffreaseon, C. P.</td>
<td>73 (et al)</td>
<td></td>
</tr>
<tr>
<td>Joy, R. C.</td>
<td>71(B) (Hachtel)</td>
<td></td>
</tr>
<tr>
<td>Joyce, D. C.</td>
<td>71 S</td>
<td></td>
</tr>
<tr>
<td>Kahan, W.</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>Kastlunger, K.</td>
<td>72 (et al)</td>
<td></td>
</tr>
<tr>
<td>Keller, H. B.</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>Kemper, G. A.</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Kirby, D. B.</td>
<td>71 (Cermak)</td>
<td></td>
</tr>
<tr>
<td>Kizner, W.</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Klema, V. C.</td>
<td>73 (Smith, B. T.)</td>
<td></td>
</tr>
<tr>
<td>Klopfenstein, R. W.</td>
<td>71(A)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>71(B) (et al)</td>
<td></td>
</tr>
<tr>
<td>Koen, B.</td>
<td>65 (Hansen, K.)</td>
<td></td>
</tr>
<tr>
<td>Kokotovic, P.V.</td>
<td>73 (Yackel)</td>
<td></td>
</tr>
<tr>
<td>Koutras, D. A.</td>
<td>68 (Berman)</td>
<td></td>
</tr>
<tr>
<td>Kraiko, A. N.</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Kreiss, H.- O.</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>Krogh, F. T.</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>Krogh, F. T.</td>
<td>73(A) P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>73(B)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>68 P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>Kugel, L. E.</td>
<td>71 (Braun)</td>
<td></td>
</tr>
<tr>
<td>Kuntzmann, J.</td>
<td>66 (Ceschin)</td>
<td></td>
</tr>
<tr>
<td>Kwan, T. V.</td>
<td>69 (Weinstein)</td>
<td></td>
</tr>
<tr>
<td>Laasonen, P.</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>Lambert, J. D.</td>
<td>73(A)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>73(B) P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>72 (et al)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66 (et al)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>65 (et al)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>62 (et al)</td>
<td></td>
</tr>
<tr>
<td>Lancaster, P.</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>Lanning, W. D.</td>
<td>65 (Richards)</td>
<td></td>
</tr>
<tr>
<td>Lapidus, L.</td>
<td>72 (Corbo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>71 (et al)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>70 S (Seinfeld)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>69 (Bard)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>67 (et al)</td>
<td></td>
</tr>
<tr>
<td>Lawson, J. D.</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Lea, R. N.</td>
<td>66 (Decell)</td>
<td></td>
</tr>
<tr>
<td>Lee, H. B.</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>Legras, J.</td>
<td>66</td>
<td></td>
</tr>
</tbody>
</table>
Lehmann, N. J. 67
Lether, F. G. 66
Levin, J. 54 (et al)
Levinson, N. 55 (Coddington) 54 (Levin)
Lim, H. C. 73
Lindberg, B. 72
71
Liniger, W. 72 (et al) 71 71 (Odeh) 70 (Gustavson) 70 (et al) 69 P 68 66 (Brayton)
Liou, M. L. 66
Liskovets, O. A. 65 S
Little, W. 65 (Hansen, K.)
Liu, S. 61 (et al)
Loscalzo, F. R. 69 P, S
Lozinskii, S. M. 58
Lunde, R. L. 71 (Branin)
Luss, D. 68 (Amundson) 68 (et al)
Luus, R. 67 (Lapidus)
Lyche, T. 72
Lyusternik, L. A. 57 (Vishik)
McCann, M. J. 69 (Robertson)
McCormick, E. 67 (Riley)
MacMillan, D. B. 68
Mah, R. 62 (et al) Mäkelä, M. 71
Makinson, G. J. 68
Martens, H. R. 70 (Cadzow)
Mentzen, B. 69 S
Michaelson, S. 62 (Mah)
Midgley, J. E. 66
Miller, J. C. P. 66 P, S
Milne, W. E. 70
Miranker, W. L. 73
Mitchell, A. R. 62 (Lambert) 57 (Fox, L.)
Moler, C. B. 73 (Smith, B.T.)
Moore, R. E. 70 (Daniel) 66
Moretti, G. 65
64
Morrell, A. J. H. 68 P
Morris, J. Ll. 69 P
Morrison, N. 69
Morton, K. W. 67 (Richtmyer)
Murphy, C. H. 65
Murray, F.J. 52 (Brock)
Neiburger, M.
60 P (et al)
Nicholson, P.
47 (Crank)
Nigro, B. J.
69
Nordsieck, A.
62
Nørsett, S. P.
73
69(A)
69(B) P
Norton, H. J.
63 (Clenshaw)
Nosrati, H.
73
Obrecht, N.
42
Odeh, F.
72 (Liniger)
71 (et al)
O'Malley, R. E., Jr.
71 S
70
68 S
O'Regan, P. G.
70
Ortega, J. M.
73
70 (et al)
Osborne, M. R.
68 P
66
Otten, H. A.
73 (et al)
Papian, L. E.
70 (et al)
Parter, S. V.
73 S (Dorr)
Paterson, W. R.
71 (et al)
Payne, H. J.
68
Pearson, C. E.
68
Pearson, W. E.
67
Pereyra, V. L.
73 P
68(B) (Daniel)
67
66
Perron, O.
29
11
09
Peschard, J.
72 (Richard)
Peterson, L.
68 (Rhoades)
Phillips, J. H.
73 (Ackberger)
Pinel, Z.
73 (Bickart)
Piotrowski, P.
69 P
Poincaré, H.
1885
Pope, D. A.
63
Porsching, T. A.
66
Pouzo, P. J.
72 (et al)
Prager, M.
66 (Babuska)
Prater, C. D.
62 (Wei)
Pring, M.
70(B) S (Garfinkel, D.)
Radok, J. R.
59
Rall, L. B.
65 P
Ralston, A.
67 P (et al)
65
60 P (et al)
Ray, W. H.
69
Reid, J. K.
70 P
Reiley, N.
72
AUTHOR LIST

Reinch, C.
71 (Wilkinson)

Rheinboldt, W. C.
70 (Ortega)

Rhoads, D. G.
68 (et al)

Rice, J. R.
71 P

Richard, S.
72 (et al)

Richards, P. I.
68 (et al)

Richardson, L. F.
27
25
10

Richtmyer, R. D.
67 (et al)

Riha, W.
72

Riley, J. D.
67 (et al)

Robertson, H. H.
69 (et al)
66 P
60 P

Rohrer, R. A.
67 (Hachtel)

Rose, D. J.
71 P (et al)

Rosenbrock, H. H.
66 (et al)
63

Roth, B.
63 (Freudenstein)

Rubinoff, M.
55

Russo, P. M.
71

Rutishauser, H.
63 P (Bauer)

Rutledge, J. D.
61 (Garfinkel, D.)

Ryabenki, V. S.
64 (Godunov)

Samarskii, A. A.
64

Sandberg, I. W.
68 (et al)

Sargent, R.
62 (Mah)

Schalit, L. M.
66 S (Campbell, E. S.)

Schneider, D. R.
73 (et al)
72 (et al)

Schumaker, L. L.
68 (Daniel)

Scott, M. R.
70

Sedgwick, A. E.
72 (Hull)

Sedore, S. R.
71 (Bowers)
67

Seinfeld, J. H.
71 (Lapidus)
70 S (et al)

Shampine, L. F.
73 S (Dorr)
73(A)
73(B) (et al)
72 (Watts)
71 (et al)
69 (et al)
68 (Bailey)

Sharma, A.
72 S

Shaw, B.
66 (Lambert)
65 (Lambert)

Shichman, H.
70
68 (Sandberg)

Sigurdsson, S. T.
72 (Lambert)
71 P

Silverberg, M.
68

Simha, R.
66 (Zimmerman)

Slifkin, M. A.
70 S (Crossley)

Sloate, H. M.
73 (et al)
Smith, B. T.
73 (et al)
Smith, J. M.
73 (Jeffreson)
Snow, R. H.
66
Solem, R. R.
62
Sonenberg, M.
68 (Berman)
Soodak, H.
61 P, S
Sörensen, J.
69 (Villadsen)
Spijker, M. N.
71(A)
71(B)
67
66
Steeleman, R.
72 (Chartres)
Stetter, H. J.
73
71(A)
71(B) P
70
69 P
65
Stewart, W. E.
67 (Villadsen)
Stiefel, E.
63 P (Bauer)
Stineman, R. W.
65
Stoer, J.
66(A) (Bulirsch)
66(B) (Bulirsch)
Stone, H. L.
69 (Weinstein)
68
Storey, C.
66 (Rosenbrock)
Strang, G.
64
62
Ström, T.
73
72
Temes, G. C.
67 S (et al)
Thaler, S.
56 (Boxer)
Thibault, R.
72 (Richard)
Thomas, J. M.
67 (et al)
Thomas, W. J.
67 (Thomas, J. M.)
Thomée, V.
67
Thompson, W. E.
67
Timlake, W. P.
65
Torrey, M. D.
65 (Richards)
Treanor, C. E.
66
Tsuchida, E.
65
Urabe, M.
70
61
Van Ness, J. E.
72
Van Wyk, R.
70
Varga, R. S.
65 S (Birkhoff)
61
Vasil'eva, A. B.
63 S
Verner, J. H.
70
69(A)
69(B) P
Villadsen, J.
69 (et al)
67 (et al)
Vishik, M. I.
57 (et al)
Vitásek, E.
69 P
66 (Babuska)
Voss, J. R.
60 (Anderson)
<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Journal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waddell, E. D.</td>
<td>68</td>
<td>Walston</td>
</tr>
<tr>
<td>Wallace, C. S.</td>
<td>73</td>
<td>(et al)</td>
</tr>
<tr>
<td>Walsh, J.</td>
<td>66</td>
<td>P</td>
</tr>
<tr>
<td>Walston, D. E.</td>
<td>68</td>
<td>(et al)</td>
</tr>
<tr>
<td>Walter, W.</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Waltman, P. E.</td>
<td>68</td>
<td>Bailey</td>
</tr>
<tr>
<td>Wanner, G.</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>72</td>
<td>(Kastlunger)</td>
</tr>
<tr>
<td>Warten, R. M.</td>
<td>67</td>
<td>Fowler</td>
</tr>
<tr>
<td>Wasow, W.</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>Watanabe, D. S.</td>
<td>73</td>
<td>Gear</td>
</tr>
<tr>
<td>Watson, G. A.</td>
<td>73</td>
<td>P</td>
</tr>
<tr>
<td>Watts, H. A.</td>
<td>72</td>
<td>(et al)</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>(Shampine)</td>
</tr>
<tr>
<td></td>
<td>69</td>
<td>(Shampine)</td>
</tr>
<tr>
<td>Wax, N.</td>
<td>72</td>
<td>(Pouzo)</td>
</tr>
<tr>
<td>Wei, J.</td>
<td>67</td>
<td>(Hicks)</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>(et al)</td>
</tr>
<tr>
<td>Weinstein, H. L.</td>
<td>69</td>
<td>(et al)</td>
</tr>
<tr>
<td>Widlund, O. B.</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>Wilf, H. S.</td>
<td>67</td>
<td>P (Ralston)</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>P (Ralston)</td>
</tr>
<tr>
<td>Wilkinson, J. H.</td>
<td>71</td>
<td>(et al)</td>
</tr>
<tr>
<td>Willers, I. M.</td>
<td>71</td>
<td>(et al)</td>
</tr>
<tr>
<td></td>
<td>71(A)</td>
<td>(Barton)</td>
</tr>
<tr>
<td></td>
<td>71(B)</td>
<td>P (Barton)</td>
</tr>
<tr>
<td>Willoughby, R. A.</td>
<td>71</td>
<td>P (Rose)</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>(Gustavson)</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>(Liniger)</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>(Brayton)</td>
</tr>
<tr>
<td>Wimp, J.</td>
<td>70</td>
<td>S</td>
</tr>
<tr>
<td>Womersley, J.</td>
<td>37</td>
<td>Hartree</td>
</tr>
<tr>
<td>Wright, K.</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>Yackel, R. A.</td>
<td>73</td>
<td>(et al)</td>
</tr>
<tr>
<td>Yasinsky, J. B.</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>Zahar, R. V. M.</td>
<td>71(A)</td>
<td>(Barton)</td>
</tr>
<tr>
<td></td>
<td>71(B)</td>
<td>P (Barton)</td>
</tr>
<tr>
<td>Zadunaisky, P. E.</td>
<td>70</td>
<td>P</td>
</tr>
<tr>
<td>Zimmerman, J. M.</td>
<td>66</td>
<td>(et al)</td>
</tr>
</tbody>
</table>
SUBJECT INDEX* FOR PAPERS AND BIBLIOGRAPHY

A

Adjoint Differential System 268-269
Aerospace Applications 49-50 [65 M2]
Algebraic Systems (Linear and Nonlinear) 10-11, 15, 55-57, 85-86, 92, 100-104, 135-150, 193
[73 S5; 72 B3, B5, S3; 71 B3, B6, B7, C1, C3, C4, G3, H1, L2, W1; 70 G6, L2, O3; 69 B3, 68 B8, D2, D4, S1; 67 P2; 66 L2, L3; 64 K1, S1; 63 F2; 59 G1; 56 B1]
Algebraic/Functional Analysis 25-26, 40, 119, 218-219
[73 G2, O1, S4; 72 B4, B8, C3, G3, H5, K2, R3; 70 O3, R1; 69 H3, P1; 67 G3, L3, S2; 66 C3, S2; 62 S2; 59 D1; 52 C1]
Applications: Output Statistics/ Curves 60, 62, 90, 108, 132, 143, 144, 147, 183, 184, 195, 196, 198, 199
Astronomy 253-258 [70 Z1]
Asymptotic Expansions 3, 67-80, 165-170, 245-258, 259-270
[73 M1, S6; 71 J1, S5; 68 C3, D3, M1; 66 B5; 65 G1, S1; 63 B1; 59 D1]

B

Backward Differentiation Formulas 7
Backward Error Analysis (See Also: Error Analysis) 11, 158, 245-258
Banach Spaces 261, 263
Bilinear (Complex) Transformations 25, 28, 218
Bistability/Separatrix Curve 8
Boundary Layer Length/Transition 3-5, 165, 193, 197, 200
Boundary Value Problems and Methods 49-50, 51-65, 165-170, 190, 229-243, 258
[73 P1; 69 B1; 68 B2, K1; 67 B1; 62 H2; 57 F1]

C

Characteristic Equations 13, 24,
38-39, 44, 119
Chemical Kinetics 2, 9, 12, 51-65, 81-94, 187-200
[73 E1, J1, L3, O2, S1; 72 B1, C4, C6, C8, G2, H1, H2, H3, S1; 71 P1; 70 A2, C4, G1, G2, O1, S2; 69 B2, C2, M1, R1, V3, W1; 68 A1, D5, F2, G2, L2, R1; 67 D3, D4, H2, P1, T3; 66 B2, P2, R1, R2, S1, Z1; 65 A1, H1, T2; 63 B3, H2; 62 E1, I2, M1, W1]

*The bibliographic citation [65 M2] refers to the second paper under M in 1965 (i.e., the paper by C. H. Murphy). If the letter is underlined, the citation is a book or proceedings.
Collocation  
[70 W2; 69 V3; 68 F2; 67 V1]

Combining Approximate Solutions  
(See Also: Interpolation) 14, 75-80, 166, 169, 201-215, 217-227, 245-258
[73 S2, S7; 72 L3, L4; 71 J1, L2; 70 Z1; 69 S2; 68 D2, D3, M1, W1; 67 P2; 66 B4, B5, L4, P1; 65 G1, T1; 63 B1, 59 R1; 27 R1; 25 R1]

Compartmental Analysis  
[65 B1]

Complex Arithmetic 15, 102

Composite Multistep Methods 14, 21-36
[73 B1, S4; 70 U1]

Conservative Systems  
[69 R2]

Convergence 5, 16-17, 73, 101, 113, 118-121, 136, 259-270
[73 G1; 72 C3; 70 A1; 69 P1; 66 S2; 56 D1; 29 P1]

Decomposition Error 259-270

Defect Function (See: Backward Error Analysis)

Difference Corrections (See: Combining Approximate Solutions)

Difference-Differential Systems 15-16, 37-48
[73 F1, H1; 68 B6; 67 B2; 63 B2]

Differential Corrections 11, 245-258
[70 Z1]

Differential Inequalities  
[64 W1]

Differential Equations Books  
[73 L1, S6, W3; 71 G4, L1; 70 A1, D1, M1; 69 M2; 68 C3; 67 R2; 66 B1, C2, G3; 65 C2, W1; 64 G1; 63 B2, C1, H1; 62 D1, F3, H1, H3, P1; 60 C2, R3; 55 C1]

Differential Stability 38  
[70 S1; 68 B6, F1; 65 C2; 63 C1; 55 C1]

Diffusion Equation 51-65, 137, 190  
[70 D2; 67 D1; 61 V1; 55 C2, C3; 54 D1; 49 L1; 47 C1; 37 H1]

Distillation Problems 190  
[68 D5; 62 M1]

Differential Stability 38  
[70 S1; 68 B6, F1; 65 C2; 63 C1; 55 C1]
INDEX FOR PAPERS AND BIBLIOGRAPHY

G

Gas Dynamics/Flames
[73 C1; 72 G2; 70 O1; 68 A1; 67 P1; 66 C1; 65 C1, D1; 64 M1; 53 H1]

H

Hybrid (Analogue) Methods
[68 B4, G1; 66 G1]

I

Implicit Differential Equations
[70 V2; 69 V2]
Implicit Midpoint Rule 5, 12, 75-80, 86, 201-215
Initial Conditions, Modification of 5, 69, 84
Instability 120
Integral Equation Approach 261 [68 F2]
Interpolation (See Also: Splines) 10-11, 115, 251-258
[72 S2; 71 M1; 68 C4]
Interval Arithmetic [66 M3]

J

Jacobian Evaluation 11, 91, 127

L

Large Systems 51-65 [69 P1; 68 D4; 66 L3]
Liapunov Function/Condition 75
Life Sciences Applications 9, 51-65, 189, 200
[73 J1; 72 H1; 70 G1, G2; 68 B5, G2; 67 G1, H2; 66 Z1; 61 G1]
Lines, Method of (See Also: Partial Differential Equations) 12, 54, 229-243
[67 H3, 65 L2; 64 W1]
Lipschitz Bound 4-5, 67-75, 114, 116, 203
Lossless Transmission lines 15, 43

M

Mathematical Modeling (See Also: Chemical Kinetics, Life Sciences Applications, Physics Applications) 9-10, 51-65, 187-200
Matricial/Nonscalar Schemes 171-186
[73 L2; 72 L1; 71 M2, S2; 70 B1; 68 E2; 67 D2]
Mesh Refinement 169-170

N

Newton Update 10, 15, 56, 86-87, 100, 102-103, 126, 130-131, 139-141, 205-206
Norms 4, 16, 67-80, 111-121, 166, 261-269
[73 G1; 72 B4, S3]
Nuclear Reactor Problems 6 [68 F2; 67 Y1; 66 B2; 65 H1]
Numerical Differentiation 11, 91, 128

O

One Step Methods (See Also: Kunge-Kutta, Implicit Midpoint Rule) 5, 6, 259-270
[73 N1; 72 H5, W1; 71 C2, H3; 70 C4, L2, U1; 69 H2, S1; 68 W1; 66 L4]
Open Questions 18
Oscillations/Periodic Solutions 19 [72 P1; 70 Z1]
Oscillatory Error Behavior 78-80, 247, 267

P

Padé Approximations 6 [72 G3; 68 El; 61 V1; 42 O1]
Parameter Estimation 9, 81, 94, 189 [72 H2; 69 B2]
Parameter Variation 12, 18-19, 135-150
[72 B5; 70 S1; 53 D1]
Parameterization 31, 217-227
<table>
<thead>
<tr>
<th>Partial Differential Equations</th>
<th>Richardson Extrapolation (See: Combining Approximate Solutions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12, 51-65, 229-243</td>
<td>Round-Off Errors</td>
</tr>
<tr>
<td>[73 A1, C1, W3; 72 C1, R1; 71</td>
<td>[62 H3]</td>
</tr>
<tr>
<td>F1, H2; 70 D2; 69 G2, V3, W1;</td>
<td>Routh-Hurwitz Criterion 26, 219</td>
</tr>
<tr>
<td>68 S3; 67 D1, H3, T2; 66 B2;</td>
<td>[69 D1; 59 G1]</td>
</tr>
<tr>
<td>65 W2; 64 S2; 63 K1; 62 K1;</td>
<td>Runge-Kutta</td>
</tr>
<tr>
<td>61 V1; 37 H1; 10 R1]</td>
<td>[73 C2, W2; 72 K1 L2; 71 C5;</td>
</tr>
<tr>
<td>Partitioning 71-75, 83-84, 86,</td>
<td>70 G3, O2, W2; 69 V1; 67 L2;</td>
</tr>
<tr>
<td>92-93, 165-170, 192, 259-270</td>
<td>66 G2; 64 B1; 63 R1]</td>
</tr>
<tr>
<td>Pascal Triangle Matrix 126</td>
<td></td>
</tr>
<tr>
<td>Passivity/Passive Interpolation</td>
<td></td>
</tr>
<tr>
<td>40, 46</td>
<td></td>
</tr>
<tr>
<td>Photochemical Processes 53</td>
<td>Scaling 67, 74</td>
</tr>
<tr>
<td>Physics Applications 7-8, 51-65, 187-200</td>
<td></td>
</tr>
<tr>
<td>[73 A1, C1; 71 F1; 62 F1; 60</td>
<td>Second Derivative Methods 6, 14-15, 95-109</td>
</tr>
<tr>
<td>N1]</td>
<td>[73 E2; 70 L2]</td>
</tr>
<tr>
<td>Principal Error Function 10, 248</td>
<td>Sensitivity Analysis</td>
</tr>
<tr>
<td>Problem Oriented Programming</td>
<td>[71 D3; 69 C1, V4; 68 B1]</td>
</tr>
<tr>
<td>Languages 81-94, 123-133, 229</td>
<td>Shooting Methods 49-50</td>
</tr>
<tr>
<td>[71 B4, B5; 68 B7, G2; 67 S1]</td>
<td>[68 K1]</td>
</tr>
<tr>
<td>[72 C4; 71 B1, B4, B5, G1, G2,</td>
<td>Singular Perturbations 3-4, 165-170, 191-194, 262</td>
</tr>
<tr>
<td>H2; 70 S3; 68 B7; 67 D3, D4,</td>
<td>[73 D2, H2, M1, Y1; 72 P1, R1;</td>
</tr>
<tr>
<td>S1; 66 S1]</td>
<td>71 D3, O2; 70 O1; 69 B1, L1;</td>
</tr>
<tr>
<td>Qualitative Theory for Ordinary Differential Equations 19</td>
<td>K1; 68 C3, O1, P2; 63 B3, V1;</td>
</tr>
<tr>
<td>Recurrence Relations (See: Variable Coefficient Difference Equations)</td>
<td>62 H2; 57 V1; 54 L1]</td>
</tr>
<tr>
<td>Riccati Systems</td>
<td>Smooth Particular Solutions (See Also: Steady State Approximation, Pseudo) 3, 67-80, 202-203, 259-270</td>
</tr>
<tr>
<td>[73 Y1]</td>
<td>[73 M1; 68 D1]</td>
</tr>
<tr>
<td>Rational Padé Approximations (See Also Padé Approximations) 176-180</td>
<td></td>
</tr>
<tr>
<td>[72 G2; 66 B4, B5; 65 G1, L1]</td>
<td>Smoothing 5, 79, 201-215</td>
</tr>
<tr>
<td>Real Time Simulation/Control 5, 18, 49-50</td>
<td></td>
</tr>
<tr>
<td>[72 J2; 70 C3; 69 N1; 68 B7,</td>
<td>Software (See Also: Validation and Comparison; Program, Digital Computer)</td>
</tr>
<tr>
<td>P1; 67 E1, L1; 63 F1; 61 A1;</td>
<td>[73 H3, K2; 72 C4; 71 H1, R1, R2]</td>
</tr>
<tr>
<td>55 R1; 54 G1]</td>
<td>SOR Iteration 56-57</td>
</tr>
<tr>
<td>Sparse Matrices 11, 55-57, 127-131, 145-149</td>
<td></td>
</tr>
<tr>
<td>[72 B3; 71 B7, H1, R2; 70 G6, O3, R1; 69 W1; 68 S3]</td>
<td>Special Methods–Special Systems 18</td>
</tr>
<tr>
<td>Special Methods—Special Systems 18</td>
<td></td>
</tr>
<tr>
<td>[73 L2; 72 C2, C6, G3, H5, K1; 71 H3, M3, R3; 68 F2, G1; 67 C1, R3, T3; 66 D3, M1; 63 P1; 62 L1; 60 C1, R2; 57 F1; 52 B1, C1; 49 F1; 42 O1]</td>
<td></td>
</tr>
</tbody>
</table>
INDEX FOR PAPERS AND BIBLIOGRAPHY

Spectral Radius/Abscissa 4-5, 73
Splines 11

Stability Charts 28, 30, 33, 34, 35, 98, 106, 226
Stability Test 22, 26, 29
Statistical Analysis

Steady State Approximation,
Pseudo (See Also: Singular Perturbations, Chemical Kinetics) 3, 83, 187-200
Steady State, Nonuniqueness of 197
Step Size Change 18, 57-59, 76-80, 99-100, 111-121, 136, 139, 207, 213-215
Stiff System Examples 2, 3, 6-9, 82-89, 107-109, 182-184, 187-200, 267
Stiffness Conditions (See Also: Lipschitz Bound, Eigenvalues, Time Constant Spread) 3, 4
Stiffness Ratio 229-242
Stochastic Differential Equations [68 P1]
Storage Considerations 56
Structured Programming 99, 151-164
Successive Subtractions 166-167
Survey Papers, Initial Value Problem
Symbolic Processing Methods

Tables of Formula Coefficients 26, 32, 33, 97, 105, 238
Time Constant Spreads 54
Transformation of Systems 4, 72, 174, 264-266
Trigonometric Methods 19
Turning Points 188
Unsmooth Differential/Difference Equations 5, 18
Validation and Comparison 2, 151-164
Variable Coefficient Difference Equations (See Also: Variable Order-Variable Step Methods)
Variable Order-Variable Step Methods 14, 16-17, 111-121, 123-133, 135-150
Variation of Parameters 113, 135-150
Variational Equations 10, 75, 247
z-Transform Methods 66 D2, L5; 65 F1; 63 F1; 56 B1