Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations

edited by
A.K. Aziz

Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations

Academic Press Rapid Manuscript Reproduction

Proceedings of a Symposium
Held at the University of Maryland
Baltimore County Campus
June 10-12, 1974

Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations

edited by
A.K. Aziz
Division of Mathematics
University of Maryland
Baltimore, Maryland
and
Institute for Fluid Dynamics
and Applied Mathematics
University of Maryland
College Park, Maryland

Academic Press, Inc.
New York San Francisco London 1975
A Subsidiary of Harcourt Brace Jovanovich, Publishers

ACADEMIC PRESS, INC.
111 Fifth Avenue, New York, New York 10003

United Kingdom Edition published by
ACADEMIC PRESS, INC. (LONDON) LTD.
24/28 Oval Road, London NW1

Library of Congress Cataloging in Publication Data

Symposium on Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations, University of Maryland, Baltimore County, 1974.
Numerical solutions of boundary value problems for ordinary differential equations.

Sponsored by the Division of Mathematics, University of Maryland, Baltimore County Campus and the National Science Foundation.
Includes bibliographies and index.
1. Differential equations–Numerical solutions–Congresses. 2. Boundary value problems–Numerical solutions –Congresses. I. Aziz, Abdul Kadir, ed. II. University of Maryland, Baltimore County–Division of Mathematics. III. United States. National Science Foundation. IV. Title.
QA372.S94 1974 515'.35 75-1141
ISBN 0–12–068660–0

PRINTED IN THE UNITED STATES OF AMERICA

CONTENTS

** Indicates author who presented paper.*

CONTRIBUTORS

Ivo Babuška, Institute for Fluid Dynamics and Applied Mathematics, University of Maryland, College Park, Maryland 20742

***R. Bulirsch**, Mathematisches Institut, Technische Universität, 2 Arcisstrasse 21, (D-8000), München, West Germany

Carl de Boor, Mathematics Research Center, University of Wisconsin, 610 Walnut Street, Madison, Wisconsin 53706

S. M. Davenport, Sandia Laboratories, Albuquerque, New Mexico 87115

T. E. Hull, Department of Computer Science, University of Toronto, Toronto, Canada M5S1A7

H. B. Keller, Applied Mathematics, California Institute of Technology, Firestone Laboratory, Pasadena, California 91109

Heinz-Otto Kreiss, Mathematics Research Center, University of Wisconsin, 610 Walnut Street, Madison, Wisconsin 53706

Leon Lapidus, Department of Chemical Engineering, Princeton University, Princeton, New Jersey 08540

M. Lentini, Apartado 59002, Los Chaguaramos, Caracas 140, Venezuela

Gunter H. Meyer, School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332

**This manuscript was not received in time to be included.*

Seymour V. Parter, Department of Computer Science, University of Wisconsin, Van Vleck Hall, Madison, Wisconsin 53706

V. Pereyra, Apartado 59002, Los Chaguaramos, Caracas 140, Venezuela

Melvin R. Scott, Applied Mathematics Division—2642, Sandia Laboratories, Albuquerque, New Mexico 87115

T. I. Seidman, University of Maryland, Baltimore, Maryland 21228

L. F. Shampine, Sandia Laboratories, Albuquerque, New Mexico 87115

John N. Shoosmith, Computer Mathematics and Programing Branch, NASA, Langley Research Center, Hampton, Virginia 23665

Jiři Taufer, Mathematical Institute of Czechoslovak Academy of Sciences, Prague 1, Zitna 25, Czechoslovakia

J. M. Varah, Computer Science Department, University of British Columbia, Vancouver 8, British Columbia, Canada

H. A. Watts, Sandia Laboratories, Albuquerque, New Mexico 87115

PREFACE

The Symposium on Numerical Solutions of Boundary Value Problems for Ordinary Differential Equations was held June 10-12, 1974, at the University of Maryland, Baltimore County Campus. It was the second conference of its kind to be held at this campus. The prior conference took place in 1972 and proceedings of it have been published by Academic Press.* The main objective of the symposium was to bring together a number of numerical analysts currently involved in research in both theoretical and practical aspects of this field.

These proceedings consist of three parts. Part I contains the content of all but one of the four 90-minute survey lectures on initial and boundary value problems; Part II contains all of the 45-minute lectures given by the invited speakers. These papers cover a large number of important results of both theoretical and practical nature. Part III contains the abstracts of three 15-minute contributed talks.

The Department of Mathematics of the University Maryland, Baltimore County Campus and the National Science Foundation were the joint sponsors of the symposium. The financial assistance of NSF and the combined hard work of many members of the University of Maryland faculty and staff, contributed immensely to the success of this meeting.

The editor wishes to express his sincere thanks to all of the people associated with the meeting and these proceedings.

**A. K. Aziz (ed.)*–The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations (1972).

This page intentionally left blank

PART I
Survey Lectures

This page intentionally left blank

NUMERICAL SOLUTIONS OF INITIAL VALUE PROBLEMS FOR ORDINARY DIFFERENTIAL EQUATIONS

T. E. Hull
Department of Computer Science
University of Toronto

ABSTRACT

This paper is intended to be a survey of the current situation regarding programs for solving initial value problems associated with ordinary differential equations. Factors to be considered in choosing a program are discussed briefly. The main methods for non-stiff problems are outlined, and a basis for comparing their relative merits is described. A summary of the results of some recent tests is presented, and this leads to conclusions and recommendations about what methods to use on non-stiff problems. The calling sequence and a program structure are also discussed. Finally, five methods for solving stiff systems are discussed briefly, and conclusions based on the results of some preliminary comparisons are presented.

1. Introduction.

Initial value techniques play an important role in the numerical solution of boundary value problems, as is evidenced, for example, by the use of shooting methods, or invariant imbedding. It therefore seems appropriate at this conference to attempt a survey of the current situation regarding initial value techniques. It also seems appropriate to emphasize the properties of programs, and especially the relative merits of different programs, rather than discussing in isolation the special properties of any particular formulas.

In section 2 we discuss a number of factors to be considered in the choice, or the design, of programs for solving initial value problems. Then in section 3 we describe very briefly the main classes of methods for handling non-stiff problems, namely the Runge-Kutta, multistep and extrapolation methods. In section 4 we consider some aspects of comparisons between different methods, and present summaries for some statistics for 15 methods. On the basis of these, and more detailed results that appear elsewhere, we draw some conclusions about what methods would be most useful to have in a program library.

In section 5 we outline a proposed calling sequence, which manages at the same time to be both simple and capable of providing a very wide choice of options. Then in section 6 we outline a program structure that we have found convenient, and that can be used to help organize programs based on quite diverse approaches.

Finally, in section 7 we mention very briefly five methods for solving stiff systems and give some preliminary results of tests on them, from which we draw tentative conclusions about which should be considered for program libraries.

2. Factors to be considered.

Many different factors must be taken into consideration when choosing, or designing, a program for handling initial value problems. The following is a list of some of the more important factors.

(a) Stiffness If the system to be solved is stiff, it is essential to use a method that has been especially designed to cope with stiffness. Non-stiff methods can solve stiff problems, and they can do so without any special difficulty, except that they can be prohibitively expensive. Typically, a standard fourth order Runge-Kutta method can cost several orders of magnitude more than a method that has been especially designed to handle stiff systems.

(b) Sparseness Large systems or ordinary differential equations are usually sparse, and the numerical technique must take advantage of the sparseness if it is to be efficient.

(c) Cost of function evaluations We shall see in section 4 that some methods are able to get along with relatively few function evaluations, but only at the expense of a rather large amount of overhead. On the other hand, some methods require more functions evaluations for the same tasks, but manage to have low overhead costs. In choosing a program one can therefore be guided by the relative cost of function evaluations for the particular problem being considered.

(d) Ease of obtaining higher derivatives Finite difference formulas involving higher derivatives can have extremely small truncation errors. If higher derivatives are conveniently obtained, and not too expensive, this suggests trying to take advantage of such formulas. This idea does not seem to have been exploited particularly well, at least not in programs that are widely available. (However, all methods for stiff systems require the second derivative, i.e., the Jacobian, or at least an approximation to it.)

(e) Discontinuities Most programs do not make any special provision for handling discontinuities. More needs to be done in this area, so that programs are tested for their ability to cope with discontinuities, and so that their documentation provides sufficient guidance about what to expect.

(f) Linearity It seems reasonable to expect a program library to have programs that are especially designed to take advantage of linearity, so that the user can choose an appropriate program when he knows that his system is linear. This possibility is not generally available at the present time.

(g) Need for intermediate information There are times when a user requires information about the solution at points other than those produced naturally in the step by step solution of a problem. For example, he may need intermediate values for tabular output; he may even need a large number in graphical output, and it would clearly be too expensive to have the numerical method use the correspondingly very small steps. As another example, he may also want intermediate information in order to determine where some function of the components has a solution. The relevant point is that some methods (such as multistep methods) are more convenient for interpolation at intermediate points than are others (such as Runge-Kutta methods).

There are of course other factors to be considered, but the above is an indication of some of the major ones. (Is applicability to boundary value problems a factor that should be considered separately?) It is to be hoped that future program collections will include special provision for such factors, and that the user will be provided with documentation that guides him easily towards making good choices.

In what follows we will concentrate primarily on indicating what programs are available for both non-stiff and stiff systems, and giving assessments of their relative merits based on recent measurements of their cost and reliability.

3. Methods for non-stiff problems.

We will consider three main classes of methods for solving non-stiff problems and then make brief mention of a few others.

(a) Runge-Kutta methods Traditional Runge-Kutta methods are based on single formulas. For example, the very well known fourth order formula

$$y_{n+1} = y_n + (k_0+2k_1+2k_2+k_3)/6 \quad ,$$

where

$$k_0 = hf(x_n,y_n)$$

$$k_1 = hf(x_n+h/2,y_n+k_0/2)$$

etc.

is often used. If a program based on such a formula attempts to choose its own stepsizes, it will estimate its local error by comparing the result using two steps with the result using one double step. Many formulas are available, including formulas of all orders up to at least 8. (In section 4 a report on two such Runge-Kutta programs is given, one being of order 4 and the other of order 6.)

During the last few years a somewhat different approach has been developed. It involves finding pairs of Runge-Kutta formulas that are of different orders but that require the same function values. One formula is used for calculating the approximation to the solution, while the difference between the two formulas provides an estimate of the error. Thus, a formula of the form

$$y_{n+1} = y_n + \sum c_i k_i$$

is used as before to calculate the approximation, while the difference between it and another of different order yields $\sum c_i' k_i$ as an estimate of the error.

Fehlberg (1968, 1969, 1970) has developed formulas of this type for all orders up to 8. Experience with some of these has been very encouraging, although we have to qualify this statement in special circumstances when the order exceeds 4, as explained later in section 4, where we report on results obtained with programs based on his formulas of orders 4 and 6.

(b) Multistep methods These methods are based on pairs of formulas of the form

$$y_{n+1} = \sum_{0}^{k} a_i^* y_{n-1} + h \sum_{0}^{k} b_i^* y'_{n-i}$$

$$y_{n+1} = \sum_{0}^{k} a_i y_{n-i} + h \sum_{-1}^{k} b_i y'_{n-i}$$

where the first formula is explicit and is used for predicting, while the second is implicit and is used for correcting.

Much of the early work on multistep methods was concerned with such questions as stability, and its relationship to the zeros of the characteristic polynomial. In practice, most of the evidence seems to indicate that the Adams formulas are generally best for general purposes (assuming the problems are non-stiff). They have good stability properties, and they are convenient to program. They are not quite as accurate as other formulas of the same order, but it is probably better to change the order than to use more complicated and less stable formulas of the same order. (For further details see, for example, Hull and Creemer (1963).)

We are thus led to using an Adams-Bashforth formula,

$$y_{n+1} = y_n + \sum_{0}^{k} b_i^* y_{n-i} ,$$

as the predictor, and an Adams-Moulton formula

$$y_{n+1} = y_n + \sum_{-1}^{k-1} b_i y'_{n-i}$$