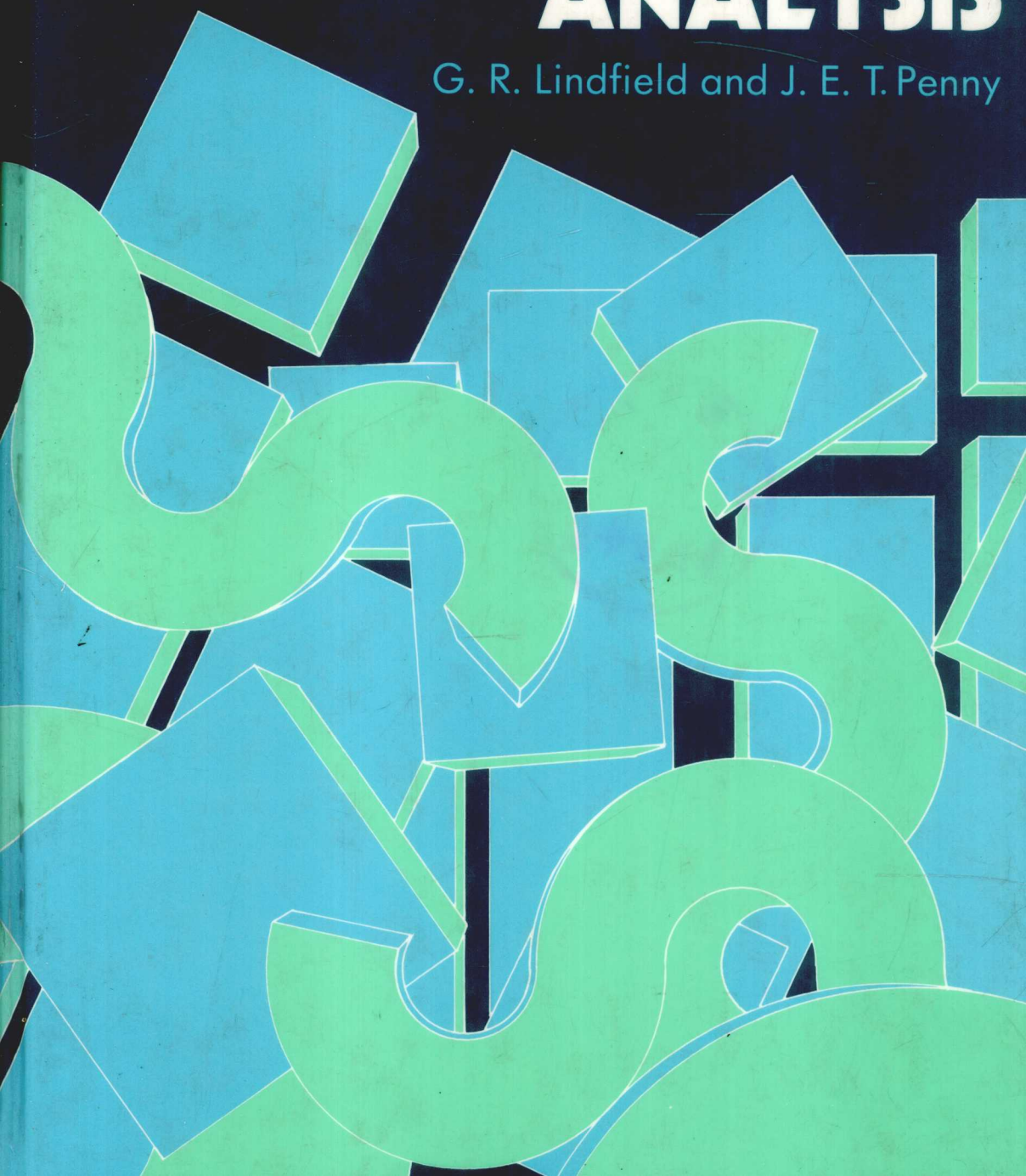
Ellis Horwood Series in **MATHEMATICS AND ITS APPLICATIONS**

MICROCOMPUTERS IN NUMERICAL ANALYSIS

G. R. Lindfield and J. E. T. Penny



MICROCOMPUTERS IN NUMERICAL ANALYSIS

G. R. LINDFIELD
Department of Computer Science and Applied Mathematics
Aston University, Birmingham

and

J. E. T. PENNY
Department of Mechanical and Production Engineering
Aston University, Birmingham

)RWOOD LIMITED
Publishers · Chichester

Halsted Press: a division of
JOHN WILEY & SONS
New York · Chichester · Brisbane · Toronto

First published in 1989 by
ELLIS HORWOOD LIMITED
Market Cross House, Cooper Street,
Chichester, West Sussex, PO19 1EB, England
The publisher's colophon is reproduced from James Gillison's drawing of the ancient Market Cross, Chichester.

Distributors:
Australia and New Zealand:
JACARANDA WILEY LIMITED
GPO Box 859, Brisbane, Queensland 4001, Australia
Canada:
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada
Europe and Africa:
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England
North and South America and the rest of the world:
Halsted Press: a division of
JOHN WILEY & SONS
605 Third Avenue, New York, NY 10158, USA
South-East Asia
JOHN WILEY & SONS (SEA) PTE LIMITED
37 Jalan Pemimpin # 05–04
Block B, Union Industrial Building, Singapore 2057
Indian Subcontinent
WILEY EASTERN LIMITED
4835/24 Ansari Road
Daryaganj, New Delhi 110002, India

British Library Cataloguing in Publication Data
Microcomputers in numerical analysis.
1. Numerical analysis. Applications of microcomputer systems
I. Title II. Penny, John E. T.
III. Series
591.4'028'5416

Library of Congress CIP available

ISBN 0–85312–781–6 (Ellis Horwood Limited)
ISBN 0–470–21415–5 (Halsted Press)

Printed in Great Britain by Hartnolls, Bodmin

MATHEMATICS AND ITS APPLICATIONS

Series Editor: G. M. BELL, Professor of Mathematics,
King's College London (KQC), University of London

NUMERICAL ANALYSIS, STATISTICS AND OPERATIONAL RESEARCH

Editor: B. W. CONOLLY, Emeritus Professor of Mathematics (Operational Research),
Queen Mary College, University of London

Mathematics and its applications are now awe-inspiring in their scope, variety and depth. Not only is there rapid growth in pure mathematics and its applications to the traditional fields of the physical sciences, engineering and statistics, but new fields of application are emerging in biology, ecology and social organization. The user of mathematics must assimilate subtle new techniques and also learn to handle the great power of the computer efficiently and economically.

The need for clear, concise and authoritative texts is thus greater than ever and our series will endeavour to supply this need. It aims to be comprehensive and yet flexible. Works surveying recent research will introduce new areas and up-to-date mathematical methods. Undergraduate texts on established topics will stimulate student interest by including applications relevant at the present day. The series will also include selected volumes of lecture notes which will enable certain important topics to be presented earlier than would otherwise be possible.

In all these ways it is hoped to render a valuable service to those who learn, teach, develop and use mathematics.

Mathematics and its Applications

Series Editor: G. M. BELL, Professor of Mathematics, King's College London (KQC), University of London

Anderson, I. — **Combinatorial Designs**
Artmann, B. — **Concept of Number: From Quaternions to Monads and Topological Fields**
Arczewski, K. & Pietrucha, J. — **Mathematical Modelling in Discrete Mechanical Systems**
Arczewski, K. and Pietrucha, J. — **Mathematical Modelling in Continuous Mechanical Systems**
Bainov, D.D. & Konstantinov, M. — **The Averaging Method and its Applications**
Baker, A.C. & Porteous, H.L. — **Linear Algebra and Differential Equations**
Balcerzyk, S. & Jösefiak, T. — **Commutative Rings**
Balcerzyk, S. & Jösefiak, T. — **Commutative Noetherian and Krull Rings**
Baldock, G.R. & Bridgeman, T. — **Mathematical Theory of Wave Motion**
Ball, M.A. — **Mathematics in the Social and Life Sciences: Theories, Models and Methods**
de Barra, G. — **Measure Theory and Integration**
Bartak, J., Herrmann, L., Lovicar, V. & Vejvoda, D. — **Partial Differential Equations of Evolution**
Bell, G.M. and Lavis, D.A. — **Co-operative Phenomena in Lattice Models, Vols. I & II**
Berkshire, F.H. — **Mountain and Lee Waves**
Berry, J.S., Burghes, D.N., Huntley, I.D., James, D.J.G. & Moscardini, A.O. — **Mathematical Modelling Courses**
Berry, J.S., Burghes, D.N., Huntley, I.D., James, D.J.G. & Moscardini, A.O. — **Mathematical Modelling Methodology, Models and Micros**
Berry, J.S., Burghes, D.N., Huntley, I.D., James, D.J.G. & Moscardini, A.O. — **Teaching and Applying Mathematical Modelling**
Blum, W. — **Applications and Modelling in Learning and Teaching Mathematics**
Brown, R. — **Topology: A Geometric Account of General Topology, Homotopy Types and the Fundamental Groupoid**
Burghes, D.N. & Borrie, M. — **Modelling with Differential Equations**
Burghes, D.N. & Downs, A.M. — **Modern Introduction to Classical Mechanics and Control**
Burghes, D.N. & Graham, A. — **Introduction to Control Theory, including Optimal Control**
Burghes, D.N., Huntley, I. & McDonald, J. — **Applying Mathematics**
Burghes, D.N. & Wood, A.D. — **Mathematical Models in the Social, Management and Life Sciences**
Butkovskiy, A.G. — **Green's Functions and Transfer Functions Handbook**
Cartwright, M. — **Fourier Methods: Applications in Mathematics, Engineering and Science**
Cerny, I. — **Complex Domain Analysis**
Chorlton, F. — **Textbook of Dynamics, 2nd Edition**
Chorlton, F. — **Vector and Tensor Methods**
Cohen, D.E. — **Computability and Logic**
Cordier, J.-M. & Porter, T. — **Shape Theory: Categorical Methods of Approximation**
Crapper, G.D. — **Introduction to Water Waves**
Cross, M. & Moscardini, A.O. — **Learning the Art of Mathematical Modelling**
Cullen, M.R. — **Linear Models in Biology**
Dunning-Davies, J. — **Mathematical Methods for Mathematicians, Physical Scientists and Engineers**
Eason, G., Coles, C.W. & Gettinby, G. — **Mathematics and Statistics for the Biosciences**
El Jai, A. & Pritchard, A.J. — **Sensors and Controls in the Analysis of Distributed Systems**
Exton, H. — **Multiple Hypergeometric Functions and Applications**
Exton, H. — **Handbook of Hypergeometric Integrals**
Exton, H. — ***q*-Hypergeometric Functions and Applications**
Faux, I.D. & Pratt, M.J. — **Computational Geometry for Design and Manufacture**
Firby, P.A. & Gardiner, C.F. — **Surface Topology**
Gardiner, C.F. — **Modern Algebra**

Series continued at back of book

Preface

This book has been written to assist non-specialists in numerical analysis and seeks to show how relatively advanced and sophisticated numerical analysis can be carried out using low-cost microcomputers. By non-specialist we mean those who use numerical methods but do not necessarily require a detailed theoretical knowledge of the subject or need to solve very large or complex problems. However, even the non-specialist must have a clear understanding of numerical methods and their range of application if procedures and the programs that implement them are to be used successfully. The non-specialist will include many undergraduate and postgraduate students, practising engineers and scientists and even sixth formers undertaking project work. They will have easy access to microcomputers and in this they are fortunate compared with their predecessors who, before the advent of the microcomputer, were forced to gain experience of numerical procedures by undertaking lengthy computation by hand.

Learning numerical methods with a microcomputer gives the user the ability to try many problems and to explore the range of applicability of algorithms. This book includes more than sixty-five BASIC programs suitable for use on microcomputers and the clarity of the description of the numerical procedures is enhanced by illustrating them with the numerical output from these programs. In addition the programs allow the user to learn from experience by experimentation and to this end a selection of problems is included in each chapter.

The overall structure of the book is as follows. Chapter 1 provides an elementary introduction to relevant aspects of computers and computing and to some of the more widely applied concepts of numerical analysis. Chapters 2 to 10 are concerned with the development of procedures and programs to solve a wide range of numerical problems. We rarely consider procedures that were developed for hand calculation, nor those which can only be implemented realistically on a main-frame computer. Emphasis is placed on the practical use of procedures and when they are not developed with full mathematical rigour, references are given. Finally in Chapter 11 we compare the performance, in terms of speed and accuracy, of several popular microcomputers and we use them to solve a selection of problems taken from Chapters 2 to 10. The mathematical prerequisites for a full understanding of the text are limited to a knowledge of simple algebra and trigonometry, basic differential and integral calculus including Taylor's series and the mean-value theorem, an understanding of the notation and rules for the manipulation of vectors and matrices and the algebra of complex numbers.

Choosing a programming language for this text was not easy. Pascal and BASIC were the most obvious candidates, FORTRAN was rejected because it is not widely used on microcomputers. Pascal has the clear advantage of being a well structured language. In contrast, BASIC is not standardised and some versions lack important features such as procedures and the *if-then-else* construct, leading to an over use of *goto* statements. However, a feature of many numerical algorithms that should not be overlooked is that their relative simplicity allows them to be programmed without loss of structural clarity using a restricted set of programming constructs. The advantages of BASIC are that it is still the most widely available programming language on microcomputers and is almost certainly the most widely known, used and understood. Consequently, after careful consideration, we have chosen to present programs in BASIC.

The programs given are written so that they will work satisfactorily, with very little modification, in most versions of BASIC and on most microcomputers. The programs could be developed by refining their structure and also their output presentation but these improvements would be at the expense of portability between machines. The robustness of the programs could be enhanced and data validation included but this would make them longer and more intricate and such refinements may mask the essential features of the numerical procedure. It should be relatively easy for readers to employ the facilities of their microcomputer to modify the programs to meet their particular requirements. It should be noted that whilst every effort has been made to check the correctness of programs we cannot guarantee that all the programs are error free. However, many of the programs have been used by the authors and their students for several years and are well and truly tested. It is also important to emphasise that even error free programs must be used with care and the reader must decide whether the application of a program to a particular problem is appropriate. All the programs have been run on the Apple Macintosh and the BBC microcomputers and the vast majority have also been run on the IBM PC-AT. In addition, many programs have been tested on other machines.

Finally the authors wish to acknowledge the help of their colleagues Gordon Betteley, Dr Barry Martin and Dr David Wilson, and also the help and patience of the staff of Ellis Horwood.

George Lindfield
John Penny
Aston University
September 1988.

Table of contents

11 Testing microcomputers

1

Using microcomputers to solve numerical problems

1.1 THE IMPACT OF THE MICROCOMPUTER

In almost every field of computation, data processing and information technology the microcomputer, as a consequence of its flexibility, user-friendliness, low cost and wide availability is becoming increasingly competitive with the large, multi-user, mainframe computer. Whilst the microcomputer cannot equal the performance of a modern mainframe computer in computing power, the modern microcomputer far exceeds the performance of many of the early large mainframe computers. An interesting illustration of this was described by Sinnott (1984). He reported that a series of lengthy and complex astronomical calculations (to determine the ephemerides of the five outer planets of the Solar system for a period of 400 years) was carried out in 1948 using the large IBM SSEC and took 120 hours. In 1984 an amateur astronomer repeated the calculations using his TRS80 microcomputer and the process took 10 hours 25 minutes. It was also estimated that it would take an unaided human calculator, working 40 hours a week, approximately 80 years to perform the same calculations. Such anecdotes are interesting in giving a feel for the rapid improvement in computer technology. The real point of this illustration is that in 1948 (or, indeed, in 1975) only the professional scientist and engineer had access to significant computing power. Today any school, college or university, any student, amateur scientist or computer hobbyist can have access to significant computing power for the expenditure of a few hundred pounds.

The microcomputer plays an increasingly important role in education and industry and much of the work that was once done on mainframe computers has now been transferred to the microcomputer. One area where both microcomputers and

mainframe computers have been used for many years is for the numerical solution and analysis of a wide range of mathematical problems. The large mainframe computer remains an essential tool for solving extremely large and complex problems despite the astonishing developments in microcomputing. However, in general users will prefer to apply a microcomputer to a numerical problem if the microcomputer memory capacity allows and if accuracy and speed of calculation are not critical factors. The microcomputer user is more in control of the computing environment than when competing with many other users for the resources of a mainframe computer. Furthermore the fast speed of operation of the mainframe computer may not be utilised during periods of high demand because the user may have to wait a considerable time before being able to access the required part of the system. Despite these comments, care must be taken to avoid the indiscriminate and uncritical use of the microcomputer and a judgement must be made based on the requirements of the particular problem. The estimation of memory requirement is normally fairly simple but the determination of the accuracy required from the system is more difficult. For some classes of numerical problem the accuracy with which the solution may be obtained is not related in any simple way to the accuracy with which the computer stores and operates on numerical values. This problem receives special consideration in later sections of this book.

The performance of a specific microcomputer is related to the hardware design and the software used by the machine. The speed of execution of programs depends on a number of factors, the most important of which are the maximum speed rating of the microprocessor chip and the form of the implementation of the computer language being used. The accuracy that can be obtained in performing computations is primarily controlled by the software implementation and in some implementations it is possible to trade off computational speed and effective memory size for increased accuracy.

The effectiveness of the microcomputer has been considerably enhanced by the dramatic increase in the memory that can be economically be incorporated into their design. At the beginning of the 1980s most microcomputers had memory capacities between 16k and 48k. By the mid-1980s memories of 512k were common and by 1987 memory capacities of 1 megabyte (1024k) were available. However, the memory accessible to programs and data may be more limited than at first appears because some memory may be used by the operating system and other software or because of poor software design. A simple one line program can be written to find the memory available and such a program has been used to obtain the results for a range of microcomputers as shown in Table 1.1.1.

The hardware, the software and the design philosophy of microcomputers continues to develop rapidly. One exciting design development which is having, and will continue to have, an impact on the implementation of numerical algorithms is the introduction of parallel processors, that is computers that can execute several elementary arithmetic operations simultaneously. By redesigning numerical

Table 1.1.1. Maximum number of elements that can be used in an integer (%), real or double precision (#) arrays in the computer indicated. The memory accessed is often dependent on the version of BASIC used.

	Maximum number of elements n		
Microcomputer	DIM A%(n)	DIM A(n)	DIM A#(n)
BBC (32k)	6329	5062	—
Amstrad CP464	21753	8700	—
Mac (512k)	159000	82350	37850
Apricot (256k)	31117	15558	7778
CBM 64	19442	7776	—
Apple	18161	7264	—
Amstrad (PC1512)	30986	15492	7745

algorithms so that they can make use of parallel processing machines, major improvements in the speed of execution of the numerical process can be obtained.

1.2 COMPUTER LANGUAGES ON THE MICROCOMPUTER

A wide range of computer languages have now been implemented on microcomputers and the number of implementations continues to grow. The purpose of these languages is to allow the user to interact as easily as possible with the computer and languages have been developed to serve the particular needs of different users. The language which is most widely used on microcomputers is BASIC. Other languages which are also available on microcomputers are FORTRAN, COBOL, Pascal, C, PILOT, LOGO and specialised languages such as APL, LISP and Prolog. These languages each have their special features which make them suitable for particular tasks. FORTRAN, BASIC, Pascal, C and APL may be used for solving scientific problems but full implementations of FORTRAN, Pascal, C and APL are available only on the more powerful microcomputers. FORTRAN was developed specifically to solve scientific problems, APL was designed to solve problems which involved extensive manipulation of matrices and Pascal and C are more general-purpose languages. COBOL is a business-orientated language which is suitable for handling large file structures containing numeric and non-numeric data. PILOT and LOGO are specialised languages, PILOT allows the microcomputer to be used efficiently for the preparation of educational material and LOGO is a graphics language. LISP and Prolog are used for research into artificial intelligence and symbolic manipulation.

All these languages are called high-level languages. High-level languages allow the user to solve a problem by writing computer executable statements which are

close to natural language statements. Clearly this simplifies the task of translating the problem-solving algorithm into a form that can be executed by the computer. Within the computer the instructions and data are stored and a procedure is required to translate from the high-level language to a machine-executable form, called machine code. This translation is achieved by using a special program known as a compiler or an interpreter. These compilers or interpreters are produced by software companies for a wide range of microcomputers. They may be supplied either on floppy discs or in ROM permanently resident within the computer. The main difference between the compiler and the interpreter is that while the compiler provides a complete and, after correction, executable machine code program the interpreter translates each statement of the high-level program as it is encountered. Consequently, since the compiled version of the program can be executed directly without need for any further translation, it should run significantly faster than an interpreted program where each statement is translated each time it is accessed.

The language used for the programs in this book is BASIC, the letters of which are an acronym for Beginner's All-purpose Symbolic Instruction Code. This high-level language is the most widely available on microcomputers and, because of its relatively simple structure, provides an easy introduction to programming for the microcomputer user. The language does have significant disadvantages because it is not well structured. This means that, because of its lack of some important programming features, programs written in BASIC may not have a clear logical structure and consequently may be difficult to understand and correct. This presents major problems for large programs, but programs to solve problems in numerical methods are often surprisingly short and consequently BASIC is usually a very adequate language for implementing these techniques.

The statement that BASIC is not well structured must be qualified because recently many new dialects of BASIC have become available that have features which allow structured programming. More recent versions of Microsoft BASIC, Hewlett Packard BASIC and BBC BASIC are examples of dialects that allow structured programming to some extent. The multiplicity of different dialects of BASIC leads to problems if a program is required to work on a wide range of machines, i.e. if it is to be portable. If we require portability, then the program must be written using only those features that are available in virtually all versions of BASIC so that it can be easily adapted to work on any microcomputer. The programs in this book are written, wherever possible, to satisfy this requirement.

1.3 NUMERICAL PROCESSES AND ERROR ANALYSIS

In general terms numerical analysis is the application of the fundamental arithmetic operations to numerical data to provide numerical solutions to problems posed in mathematical form. The microcomputer provides an excellent tool for carrying out numerical work. Even those microcomputers with small memory can be used to

solve a wide range of numerical problems since, as will be seen from later chapters of this book, the majority of programs require only a small amount of memory. However, although memory size may not be a problem in terms of program storage, it does present problems in the way numerical data is stored, since this will often lead to numerical errors.

Errors arise in computations performed on any computer because the finite storage capacity will place a restriction on the storage space available for each number. Numbers are automatically rounded to the number of significant places specified for the particular machine. The type of errors which arise from this process are known as rounding errors. For example the value given for 1/3 on many microcomputers is .333333333, clearly this number has been rounded to nine significant digits.

Although these rounding errors in themselves may seem insignificant, when expressions which involve many calculations are performed using the standard arithmetic operators with rounded numbers the errors will tend to accumulate.

As an example of how errors are magnified, consider the problem of solving the quadratic equation

$$ax^2 + bx + c = 0 \tag{1.3.1}$$

The solution of this equation is given by the well known formula

$$x = [-b \pm \sqrt{(b^2 - 4ac)}]/(2a) \tag{1.3.2}$$

Consider the calculation of the root

$$x = [-b + \sqrt{(b^2 - 4ac)}]/(2a) \tag{1.3.3}$$

This formula may give inaccurate results when $4ac$ is small relative to b^2. The inaccuracy is caused by the cancellation of many significant digits due to the subtraction of nearly equal values. To avoid this problem we can multiply the numerator and denominator of (1.3.3) by $b + \sqrt{(b^2 - 4ac)}$ to give

$$x = -2c/[b + \sqrt{(b^2 - 4ac)}] \tag{1.3.4}$$

This equation has removed the cancellation errors introduced by the subtraction of nearly equal quantities. Table 1.3.1 shows a comparison between the use of (1.3.3) and (1.3.4) to determine one root of a quadratic equation.

Having established that small errors can lead to significantly greater errors during calculations, we must now examine how computers store numbers and how arithmetic operations may magnify errors.