numerical algorithms: origins and applications

bruce w. arden
professor of computer and communication sciences,
and associate director, computing center, university of michigan

kenneth n. astill
professor of mechanical engineering, tufts university

addison-wesley publishing company
reading, massachusetts • menlo park, california
london • amsterdam • don mills, ontario • sydney

This book is published
under the editorship of
Michael A. Harrison

Printed in the United States of America. Published simultaneously in Canada. Library of Congress Catalog Card No. 76-100853.

ISBN 0-201-00336-8
CDEFGHIJKL-MA-7987

numerical algorithms: origins and applications

To our wives Patty and Pat
and our children,
for their patience and understanding

preface

This book has its origins in a previous text written by one of the authors, entitled *An Introduction to Digital Computing*. The earlier book was published by Addison-Wesley in 1963 and, as the name implies, it was intended to be a general introduction to digital computing. In the intervening years there has been an enormous increase in the subject matter of computing with the result that, increasingly, the four general topics of computers, programming, numerical algorithms, and nonnumerical algorithms are introduced separately. As always, such specialization has both benefits and detriments. On the positive side, it permits the production of books in which it can reasonably be assumed that the reader has already been introduced to machines and programming. On the negative side, the specialization may result in deemphasizing the need for a detailed consideration of algorithms, by setting it apart from the highly motivating material associated with programming.

This book, while it assumes a minimal computing background, attempts to retain a connection between the development and formulation of numerical algorithms and their programmed implementation. Also, for additional motivation many of the methods are introduced by the statement of a physical problem. In effect, the book is intermediate between a "FORTRAN Computing Course" and a formal book on numerical analysis. As such, it should be helpful for upperclass undergraduates and first-year graduate students in science, engineering, and computing. The authors strongly believe that anyone, regardless of discipline, who wishes to be regarded as educated in computing should recognize the essentiality of the necessary-number and function approximation (indeed, finitude in general) in the development and use of algorithms for numerical computation. Hopefully, this book will be useful in the acquisition of such understanding, as well as in providing factual knowledge about a carefully selected variety of useful numerical algorithms. It would be gratifying if, in addition, some readers achieve sufficient motivation to study in even more analytic detail the relationships of the concept of infinitude to numerical algorithms.

The prerequisites for a course using this book would include integral calculus, some differential equations, and the ability to program for a digital computer. While the emphasis is on numerical methods, it should be recognized that normally the ultimate objective in formulating a numerical solution is to carry it out on a digital computer. Consequently emphasis is given to program development.

In the typical pursuit of this course, one would expect that a student would program and execute four to eight problems on the computer in the course of a semester. With these objectives in mind, the text contains a liberal sprinkling of examples requiring the development of programs to solve problems with the methods being treated. Each program is written in FORTRAN IV language, and each is normally accompanied by a flow chart. In the actual formulation, the program was developed directly from the flow chart. Flow charts are a convenient mechanism for translating the mathematical problems into programs, and they represent a useful way for scientists and engineers to communicate their problems to professional programmers. Symbols used in the flow charts are defined in Appendix A. The manner in which the information flows is evident from the inspection of one or two flow charts, and students quickly develop skill in interpreting and developing flow charts if they do not already have this facility.

All of the FORTRAN programs included are complete in the sense that they have been compiled and run with sample data. Where an independent test program is required, it is included with the program. The input and output lines shown with the programs are those actually used and produced by the programs. The various FORTRAN languages do differ in small details. These programs were run on an IBM System/360, Model 67, at The University of Michigan under the MTS operating system. To the best of the authors' knowledge the programs are "standard" in form, with the single exception that a free-format input was often used. In such cases input values could be arbitrarily spaced, separated by commas, as long as each number occupied a field the same size as (or smaller than) the field description in the associated format statement.

The authors would like to express thanks to Mrs. Arthur Wallace and Mrs. Thomas Steel for their help in typing the manuscript, and to Mr. Joseph Paster for his help in preparing some of the programs.

October, 1969 B. W. A.
K. N. A.

In the realm of acknowledgments it is probably unusual for an author to comment on the efforts of a co-author. Considering the current pace of academic life, a joint effort of this type might easily be frustrated by the difficulties of distance and demanding schedules. That such frustrations did not arise is due largely to my colleague in this work who completely undertook the pressing and often tedious task of final editing and proofreading. Without his willingness to overcome delays by additional personal effort, this book would not be a reality at this time.

Ann Arbor, Michigan B. W. A.

contents

chapter one

computational error

The solution of problems through numerical computation often entails many repeated arithmetic operations. Small errors arising from input numbers or approximations in computer operations can propagate in the process to magnitudes which are unacceptable in the result. In this section we shall examine some of the sources of errors and how they accumulate.

Real numbers, with their infinite string of digits, cannot be represented in the finite number of storage locations in a digital computer; they must be approximated by rational numbers, for example,

$$\pi \approx \frac{31415926}{10000000} = 3.1415926.$$

This inability to represent real numbers is not the only source of error in a computational problem. The four categories listed below are in the realm of conscious error as opposed to mistakes. Since these errors are known to exist, the assumption is that they cannot be eliminated but that, hopefully, their magnitudes may be estimated.

1. The equations and expressions which are used to describe physical processes are, in general, approximations or idealizations which at the outset introduce a disparity between the physical problem and its computational analog. Such errors could be called *formulation* errors.

2. In actual fact, a digital computer is limited to performing the arithmetic operations on a limited set of rational numbers. Or, stated differently, only rational functions can be evaluated, where *rational function* is defined to mean any function that can be evaluated by operating on numbers using only addition, subtraction, multiplication, and division. The function

$$g(x) = \frac{4x^2 + 3}{3x^3 - 2x^2 + x - 8}$$

is an example. Functions which are defined by limiting processes, such as

$$\ln x = \int_1^x \frac{dt}{t},$$

must somehow be represented by a rational function, and thus an error is introduced. This error is called a *truncation* error since the rational function is often obtained by truncating (i.e., terminating) an infinite series after a specified number of terms.

3. The already mentioned fact that an infinite number of digits cannot be used to represent a number gives rise to what is called *round-off* errors. Remembering that a number is a polynomial, one sees that this type of error arises from the truncation of terms in this polynomial. However, it should be kept in mind that the term "truncation error" refers to the error arising from functional approximation, not number approximation.

4. Physical quantities can be measured only to a limited accuracy. When such values are used in computations, their indefinite value, or error, is frequently more restrictive than the limited number of digits available to express these measures. Such errors could be called *measurement* errors.

There is no way to estimate formulation error other than to check by observation how well the mathematical expression predicts the physical case. When the formulation error is admittedly large, then, occasionally, a great deal of effort is expended to reduce the other types of error. It would seem that the following corruption of an old adage would apply in such cases: "A thing not worth doing at all is not worth doing well."

Estimation of truncation errors is one of the main tasks of numerical analysis. Examples of such estimation are given in later sections.

Round-off and measurement errors have a similar effect even though the causes differ. The precision with which a number can be expressed is limited. The magnitude of the error caused by this imprecision is relatively simply determined in detail for individual operations and is the principal concern of this chapter. However, the determination of the error in the result of a complicated calculation (even though the error in the original quantities is known) is a vexing problem which cannot always be solved. As the last statement implies, error is propagated; that is, if one or both operands in an operation are approximate, then the result of the operation is approximate or in error. If this result is then used as an operand, the error is propagated to the value of another expression, etc. In addition to the error propagated at each step of a calculation, an error may also be *generated* because the operations themselves (not the operands) are only approximations. The most obvious example of an inexact operation is division, where, unless the divisor divides evenly into the dividend, the quotient will have an infinite number of digits. Restricting the number of quotient digits makes the operation an approximation. One also uses approximate multiplication, i.e., one retains not all the product digits but only a specified number of the most significant digits (generally, as many digits as the word size permits). In floating-point arithmetic, even addition and subtraction are often approximated. When numbers with different scale factors (or possibly even with the same scale factor) are added, not all the sum digits can be accommodated in the basic number size of the machine, and hence the least significant sum digits are dropped. In the following discussion such *generated* errors are not considered; the discussion is limited to the propagation of errors due to inaccurate operands.

Significant numbers

The most common way of indicating the degree of precision of a number is to write only those digits which are known accurately or, in other words, are *significant*. The rightmost digit that is written can be in error by at most half a unit because a greater error would mean that rounding would produce a different final digit. Writing the significant number 1.2932 implies that the "true" value represented by this number is less than or equal to 1.293249999 . . . and greater than or equal to 1.29315. If these extreme values are rounded or "half-adjusted" to four decimal places by adding .00005 and then dropping the digits from the fifth decimal place on, the result is 1.2932. Alternatively, this range can be indicated by writing 1.2932 ± 0.00005, which in turn can be written 1.2932 (.5), where the number enclosed in parentheses is understood to be in units of the rightmost place. Another alternative expression which allows the error amount to be written as an integer is 1.29320 (5). The shortcomings of significant number notation become apparent if one now supposes that the example number is not known quite so accurately and the range of indefiniteness is, say, 7 units in the first place dropped, i.e. 1.29320 (7). But this representation is no longer a significant number, and the next larger significant number which includes this range (1.29313 to 1.29327) is 1.2930 (5) = 1.29300 (50), whose range is 1.29250 to 1.29350. This notation forces one to designate more variation than is known to exist in the given approximation. An alternative approach is to deal directly with the range of the numbers, as was done above, rather than use the significant number notation. Before the discussion turns to *range numbers*, it should be noted that when an integer with trailing zeros, say 92600, is written, it is not clear how many of the digits are significant. To clear up this ambiguity one must either explicitly state the significance or adopt some writing convention, such as 9260×10^1, which indicates, in this example, that the rightmost zero is not significant.

Range numbers

In range-number form, a number N is replaced by a pair, the largest and smallest possible values in its range. These high and low values are bracketed and displayed one above the other as shown below.

$$\begin{bmatrix} N_H \\ N_L \end{bmatrix}$$

The significant number 1.2932 becomes

$$\begin{bmatrix} 1.29325 \\ 1.29315 \end{bmatrix}.$$

If a number is known exactly, the extremes of the range are the same:

$$6\tfrac{3}{8} = \begin{bmatrix} 6.375 \\ 6.375 \end{bmatrix}.$$

The arithmetic operations expressed in terms of range numbers and some numerical examples are shown below.

Addition

$$x + y = \begin{bmatrix} x_H \\ x_L \end{bmatrix} + \begin{bmatrix} y_H \\ y_L \end{bmatrix} = \begin{bmatrix} x_H + y_H \\ x_L + y_L \end{bmatrix},$$

$$1.29(6) + 7.81(4) = \begin{bmatrix} 1.35 \\ 1.23 \end{bmatrix} + \begin{bmatrix} 7.85 \\ 7.77 \end{bmatrix} = \begin{bmatrix} 9.20 \\ 9.00 \end{bmatrix}.$$

Subtraction

If the signs are considered to be a part of the numbers, the problem becomes one of addition.

$$1.29(6) - 7.81(4) = \begin{bmatrix} 1.35 \\ 1.23 \end{bmatrix} + \begin{bmatrix} -7.77 \\ -7.85 \end{bmatrix} = \begin{bmatrix} -6.42 \\ -6.62 \end{bmatrix}$$

Multiplication

$$x \cdot y = \begin{bmatrix} x_H \\ x_L \end{bmatrix} \times \begin{bmatrix} y_H \\ y_L \end{bmatrix} = \begin{bmatrix} x_H \cdot y_H \\ x_L \cdot y_L \end{bmatrix},$$

$$3.46(7) \times 2.120(5) = \begin{bmatrix} 3.53 \\ 3.39 \end{bmatrix} \times \begin{bmatrix} 2.125 \\ 2.115 \end{bmatrix} = \begin{bmatrix} 7.50125 \\ 7.16985 \end{bmatrix}.$$

In this instance negative signs, if any, should precede the brackets since the objective is to produce the products that are largest and smallest in absolute value. As an example consider

$$\begin{bmatrix} -3.39 \\ -3.53 \end{bmatrix} \quad \text{to be} \quad -\begin{bmatrix} 3.53 \\ 3.39 \end{bmatrix}.$$

Division

$$x \div y = \begin{bmatrix} x_H \\ x_L \end{bmatrix} \div \begin{bmatrix} y_H \\ y_L \end{bmatrix} = \begin{bmatrix} x_H \div y_L \\ x_L \div y_H \end{bmatrix}.$$

Here again signs should be placed outside the brackets so that the values within are highest and lowest in absolute value.

$$4.246(8) \div 0.120(5) = \begin{bmatrix} 4.254 \\ \\ 4.238 \end{bmatrix} \div \begin{bmatrix} 0.125 \\ \\ 0.115 \end{bmatrix} = \begin{bmatrix} 36.992 \\ \\ 33.904 \end{bmatrix}.$$

The quotients should be truncated so that no possible quotients are excluded from the range. Thus the quotient 4.254 ÷ 0.115 = 36.9913 . . . was adjusted to 36.992 to be included in the range 36.9913 . . . (Note that 36.991 would have excluded this value.)

If the extremes of the range are of different sign, zero is included in the range. This inclusion is not permissible if the range number is a divisor, and will require adjustments in the rules above for the other cases. However, all the rules given can be subsumed under one general rule prescribing the procedure of obtaining the range number of the result of an operation: Of the four possible combinations of the range number pairs which are operands, select the two giving the largest range to designate the resulting range number.

Range numbers are useful to demonstrate the propagation of errors. The following evaluation employing significant numbers illustrates this point.

$$y = 0.12 \times 236.4 - (63.8 \times 2.01) \div 25$$

$$= \begin{bmatrix} 0.125 \\ \\ 0.115 \end{bmatrix} \times \begin{bmatrix} 236.45 \\ \\ 236.35 \end{bmatrix} - \begin{bmatrix} 63.85 \\ \\ 63.75 \end{bmatrix} \times \begin{bmatrix} 2.015 \\ \\ 2.005 \end{bmatrix} \div \begin{bmatrix} 25.5 \\ \\ 24.5 \end{bmatrix}$$

$$= \begin{bmatrix} 29.55625 \\ \\ 27.18025 \end{bmatrix} - \begin{bmatrix} 128.65775 \\ \\ 127.81875 \end{bmatrix} \div \begin{bmatrix} 25.5 \\ \\ 24.5 \end{bmatrix}$$

$$= \begin{bmatrix} 29.55625 \\ \\ 27.18025 \end{bmatrix} - \begin{bmatrix} 5.251336 \\ \\ 5.012500 \end{bmatrix}$$

$$= \begin{bmatrix} 24.543750 \\ \\ 21.928914 \end{bmatrix}.$$

If this result is adjusted to two decimal places, it becomes

$$\begin{bmatrix} 24.55 \\ \\ 21.92 \end{bmatrix}.$$

As a significant number this range is represented by 2×10^1, i.e., by only one significant figure. Except for limited hand calculation, the utility of range numbers is restricted to such demonstrations. For the determination of error bounds, it is desirable to express the amount of the error explicitly. Moreover, the error from the approximation of infinite decimal numbers does not appear in range numbers, i.e.,

$$\tfrac{1}{3} \approx 0.3333 = \begin{bmatrix} 0.3333 \\ \\ 0.3333 \end{bmatrix}.$$

Since the error is known, there is no range; the *approximation-error* form corrects these deficiencies.

Approximation-error numbers

As the name implies, numbers in approximation-error form consist of two parts, the approximation and the error. For instance, $\frac{1}{3} = 0.3333 + \frac{1}{3} \times 10^{-5}$, and, in general, $x = \bar{x} + \epsilon$. More often than not the actual error is not known; only the range is known. Hence, the 1.2932, the significant number previously used as an example, is in this form:

$$1.2932 + \epsilon, \qquad -0.00005 \leq \epsilon < 0.00005.$$

Thus a complete statement is $x = \bar{x} + \epsilon$, where $-\eta \leq \epsilon < \eta$. The basic arithmetic operations are expressed in this form as follows.

Addition

$$x + y = (\bar{x} + \epsilon_1) + (\bar{y} + \epsilon_2) = (\bar{x} + \bar{y}) + (\epsilon_1 + \epsilon_2).$$

The errors are additive. If n numbers are added,

$$x_1 + x_2 + x_3 + \cdots + x_n,$$

the error of the sum will be

$$\epsilon_1 + \epsilon_2 + \epsilon_3 + \cdots + \epsilon_n.$$

If, in addition, the individual errors have a common range, that is, $-\eta \leq \epsilon_i < \eta$ for $i = 1, 2, \ldots, n$, then

$$\epsilon_1 + \epsilon_2 + \cdots + \epsilon_n = \sum_{i=1}^{n} \epsilon_i < n\eta.$$

As a simple example, suppose that the significant numbers 11, 12, 13, . . . , 20 are added. The error for each number is less than 0.5, and the total error for the sum of the ten numbers is less than ten times that upper bound:

$$\epsilon_{\text{total}} < 10 \cdot 0.5 = 5.$$

Subtraction

$$x - y = (\bar{x} + \epsilon_1) - (\bar{y} + \epsilon_2) = (\bar{x} - \bar{y}) + (\epsilon_1 - \epsilon_2).$$

Since the values of ϵ_1 and ϵ_2 may be positive or negative, the determination of the maximum error is the same as in addition.

Multiplication

$$x \cdot y = (\bar{x} + \epsilon_1)(\bar{y} + \epsilon_2) = (\bar{x} \cdot \bar{y}) + (\epsilon_2\bar{x} + \epsilon_1\bar{y} + \epsilon_1\epsilon_2)$$

Since the errors are usually small compared to the approximation numbers, the $\epsilon_1\epsilon_2$-term is very often neglected. As before, to determine the maximum error that can be made in a multiplication, the positive upper bounds for the errors are used, together with the absolute values of the approximating numbers. Assuming that $-\eta_1 \leq \epsilon_1 < \eta_1$ and $-\eta_2 \leq \epsilon_2 < \eta_2$, one has

$$\epsilon_2\bar{x} + \epsilon_1\bar{y} < \eta_2|\bar{x}| + \eta_1|\bar{y}|.$$

Division

This operation is a little more complicated.

$$\frac{x}{y} = \frac{\bar{x} + \epsilon_1}{\bar{y} + \epsilon_2} = \frac{\bar{x}(1 + \epsilon_1/\bar{x})}{\bar{y}(1 + \epsilon_2/\bar{y})} = \frac{\bar{x}}{\bar{y}}\left(1 + \frac{\epsilon_1}{\bar{x}}\right)\left(1 + \frac{\epsilon_2}{\bar{y}}\right)^{-1}$$

The rightmost term can be expanded by means of the binomial theorem to yield the common series

$$\frac{1}{1 + z} = 1 - z + z^2 - z^3 + \cdots$$

Then

$$\frac{x}{y} = \frac{\bar{x}}{\bar{y}}\left(1 + \frac{\epsilon_1}{\bar{x}}\right)\left(1 - \frac{\epsilon_2}{\bar{y}} + \frac{\epsilon_2^2}{\bar{y}^2} - \cdots\right).$$

If all second- and higher-order terms are neglected, i.e., those involving products of the ϵ's, then

$$\frac{x}{y} \cong \frac{\bar{x}}{\bar{y}}\left(1 + \frac{\epsilon_1}{\bar{x}} - \frac{\epsilon_2}{\bar{y}}\right) = \left(\frac{\bar{x}}{\bar{y}}\right) + \left(\frac{\epsilon_1\bar{y} - \epsilon_2\bar{x}}{\bar{y}^2}\right).$$

The maximum error can be obtained by replacing the errors with their upper bounds and using the absolute values of the approximations:

$$\frac{\epsilon_1\bar{y} - \epsilon_2\bar{x}}{\bar{y}^2} < \frac{\eta_1|\bar{y}| + \eta_2|\bar{x}|}{\bar{y}^2},$$

where

$$-\eta_1 \leq \epsilon_1 < \eta_1 \qquad \text{and} \qquad -\eta_2 \leq \epsilon_2 < \eta_2.$$

To illustrate this kind of analysis, the error in evaluating a second-degree polynomial, $a_2x^2 + a_1x + a_0$, will be computed. This error estimation will be carried