Topics in Interval Analysis

Edited by E. R. Hansen

Topics in Interval Analysis Hansen Oxford

This book is based on lectures given at a symposium on interval analysis, sponsored by the Oxford University Computing Laboratory, at which developments in research and the application of interval analysis were discussed by leading authorities. The book is divided into two parts, each with an introductory chapter. Part 1 discusses methods for bounding errors in the computed solutions, to algebraic problems, especially root finding and matrix computations. A discussion of triplex-arithmetic and some aspects of its programming is included.
Part 2, on continuous problems, discusses error bounding in the solution of ordinary and partial differential equations and integral equations. Among other topics statistical distributions of errors applied to linear programming are considered.

£2·50p *net*
50*s*. *net*

Topics in Interval Analysis

EDITED BY

E. HANSEN

OXFORD
AT THE CLARENDON PRESS
1969

Oxford University Press, Ely House, London W. 1

GLASGOW NEW YORK TORONTO MELBOURNE WELLINGTON
CAPE TOWN SALISBURY IBADAN NAIROBI LUSAKA ADDIS ABABA
BOMBAY CALCUTTA MADRAS KARACHI LAHORE DACCA
KUALA LUMPUR SINGAPORE HONG KONG TOKYO

PRINTED IN GREAT BRITAIN

Preface

In late 1967, Professor Leslie Fox, Director of the Oxford University Computing Laboratory, invited several persons to speak at a Symposium on Interval Analysis. This meeting was sponsored by the Oxford University Computing Laboratory and took place on 24 and 25 January 1968 at the Culham Laboratory, Culham, England.

The speakers were Dr. Michael Dempster of Oxford, Dr. Eldon Hansen of Lockheed Missiles and Space Co. (who at that time was a Visiting Research Fellow at Oxford), Professor Frederick Krückeberg of the University of Bonn, Professor Jean Meinguet of the University of Louvain, Professor Ramon E. Moore of the University of Wisconsin, Professor Karl Nickel of the Technische Hochschule, Karlsruhe, and Dr. James H. Wilkinson of the National Physical Laboratory, Teddington.

Drs. Dempster and Meinguet each gave one lecture; the other speakers gave two. The lectures were subsequently submitted in written form and appear as separate chapters in this book. Titles and authors are identified in the table of contents. Unfortunately, Dr. Wilkinson found it impracticable to submit a written contribution.

The symposium was divided into two parts; one on algebraic problems and one on continuous problems. Professor Moore was asked to introduce these topics in his two lectures. As so often and so naturally happens, the lectures did not necessarily fit entirely into one category. Partly for this reason, but especially to provide greater continuity, the order of presentation of lectures has been changed.

During discussion periods, several topics were discussed which were not a part of the formal lectures. The speakers were encouraged to include such topics when preparing their lectures in written form. Some have done so. In this spirit, Chapter 10 has been added because of repeated comments in lectures and discussions indicating interest in Moore's 'centred form'.

The notations used are not consistent from one chapter to the next. Different authors denoted quantities and ideas in different ways which

often were particularly suitable in different contexts. Hence it seemed desirable to leave notations unchanged. However, relevant comments to alternative notation have been added where appropriate, and literary styles have been altered for uniformity.

E. H.

Contents

PART 1

ALGEBRAIC PROBLEMS

1 · Introduction to Algebraic Problems

MATHEMATICS is considered by many as an 'exact science'. Mathematicians themselves continually talk of such precise-sounding terms as 'proofs' and 'solutions'. Numerical computation, particularly by computing machines, is regarded by most people as completely straightforward and flawless and reliable. Of course, the numerical analyst knows otherwise. It cannot be much of an exaggeration to say that nearly all the numbers that have been computed so far are of unknown (although 'probably' adequate) accuracy.

Even though mathematical techniques are known (and continually improved) for the analysis of error in most types of computation, they are rarely used in practice. It can be difficult, time-consuming, and expensive to carry out by hand a complete error-analysis for a complicated practical problem.

A variety of alternative procedures are commonly used. In many—perhaps most—cases they are quite reasonable and sensible procedures. Comparison of computed results with experimental measurements in some test cases, application of an algorithm to a simple test case with known solution, statistical estimation of error, 'asymptotic' estimates based on repeating a computation several times with changes in 'program parameters' such as step size, number of iterations, word length, and so on; all these are useful and often easy to apply in order to gain confidence in the validity and accuracy of the results of numerical computation. Chapter 6, by Meinguet, and Chapter 12, by Dempster, discuss some techniques for estimating errors in this spirit of gaining reasonable confidence in numerical results.

It seems a pity, nevertheless, that mathematical rigour should have to be abandoned precisely at the point when a problem is reduced to arithmetic.

Interval analysis is concerned with techniques that can be programmed for computing machines and contain both a computation and a rigorous and complete error-analysis of the results of the computation. Chapters

1, 2, 3, 5, and 7–11 by Hansen, Krückeberg, Moore, and Nickel deal with this approach.

There are three sources of error in numerical computation. The first and most serious, because it cannot be made arbitrarily small by additional computation, is the *propagation of error in initial data.* I include here uncertainties in the mathematical equations that are supposed to describe some physical process. For example, we might have a differential equation such as

$$y'' = ay^{-b}$$

with boundary conditions $y(0) = y_0$, $y(1) = y_1$, and perhaps all the quantities y_0, y_1, a, and b are only known approximately. The question then arises: 'How much error is there in the solution as a result of errors of given magnitude in the quantities y_0, y_1, a, and b?'

The second and third kinds of error in computation, *round-off error*, caused by computing with numbers rounded-off to a finite number of digits, and *truncation error*, caused by truncating infinite sequences of arithmetic operations after a finite number of steps, can always, in principle, be made arbitrarily small by doing enough computation.

In practice, of course, the question of efficiency arises and it is of importance to devise computational schemes so that just enough computing is done to make the second and third kinds of errors as small as warranted.

If computation is still 'an art', as some have suggested, rather than a science, then our work is not finished.

The propagation of error in initial data and the accumulation of round-off error *in any finite sequence of arithmetic operations* can both be rigorously bounded by the computing machine during the course of the arithmetic operations simply by performing them in *rounded-interval arithmetic* instead of ordinary machine arithmetic.

Arithmetic operations with intervals are defined as follows:

$$[a,b]+[c,d] = [a+c, b+d],$$

$$[a,b]-[c,d] = [a-d, b-c],$$

$$[a,b][c,d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)],$$

$$[a,b]/[c,d] = [a,b][1/d, 1/c) \quad (\text{if } 0 \notin [c,d]).$$

Rounded-interval arithmetic consists of adding a low-order bit to right-end points and subtracting a low-order bit from left-end points, *when necessary*, in order to compute intervals that are as narrow as possible and contain the exact interval results.

Both Fortran and Algol compilers can be extended in order to be able to do this kind of arithmetic. In this way ordinary algebraic expressions in Fortran or Algol symbols can be compiled and executed either in rounded-interval arithmetic or in ordinary machine arithmetic at the option of the user. In case a variable is declared to be of interval type, then an interval of initial values may be given. Chapter 2, by Nickel, contains further discussion concerning these things.

We now give some very simple illustrations of the kind of numerical results possible with rounded-interval arithmetic.

Consider the computation of the quantities $1/n!$ for $n = 1, 2, 3, \ldots, N$, using rounded-interval arithmetic. We suppose, for simplicity, that three-decimal-digit, rounded, normalized floating-point arithmetic is available on the computer. Define the quantities $f_n = 1/n!$ recursively as

$$f_1 = 1, \qquad f_{n+1} = f_n/(n+1) \quad (n = 1, 2, \ldots, N).$$

We suppose that N is small enough for the numbers $n+1$ $(n = 1, 2, \ldots, N)$ to be computed (or stored from input) exactly. The computation would proceed as follows for intervals F_n that contain f_n $(n = 1, 2, \ldots, N)$:

$$\begin{aligned}
F_1 &= [0{\cdot}100\times 10^{1}, 0{\cdot}100\times 10^{1}],\\
F_2 &= [0{\cdot}100\times 10^{1}, 0{\cdot}100\times 10^{1}]/[0{\cdot}200\times 10^{1}, 0{\cdot}200\times 10^{1}]\\
&= [0{\cdot}500\times 10^{0}, 0{\cdot}500\times 10^{0}],\\
F_3 &= [0{\cdot}500\times 10^{0}, 0{\cdot}500\times 10^{0}]/[0{\cdot}300\times 10^{1}, 0{\cdot}300\times 10^{1}]\\
&= [0{\cdot}166\times 10^{0}, 0{\cdot}167\times 10^{0}],\\
F_4 &= [0{\cdot}166\times 10^{0}, 0{\cdot}167\times 10^{0}]/[0{\cdot}400\times 10^{1}, 0{\cdot}400\times 10^{1}]\\
&= [0{\cdot}415\times 10^{-1}, 0{\cdot}418\times 10^{-1}],\\
F_5 &= [0{\cdot}415\times 10^{-1}, 0{\cdot}418\times 10^{-1}]/[0{\cdot}500\times 10^{1}, 0{\cdot}500\times 10^{1}]\\
&= [0{\cdot}830\times 10^{-2}, 0{\cdot}836\times 10^{-2}],\\
F_6 &= [0{\cdot}830\times 10^{-2}, 0{\cdot}836\times 10^{-2}]/[0{\cdot}600\times 10^{1}, 0{\cdot}600\times 10^{1}]\\
&= [0{\cdot}138\times 10^{-2}, 0{\cdot}140\times 10^{-2}],
\end{aligned}$$

and so on.

In the next example we illustrate the simultaneous bounding of propagation of error in initial data and round-off error accumulation.

Consider the problem of computing an approximate value and bounding its error for the quantity

$$y = \frac{a_1+a_2x}{a_3+a_4x^2}$$

when it is known that $x = 0{\cdot}452 \pm 0{\cdot}001$ and

$$a_1 = 0{\cdot}200 \pm 0{\cdot}001, \qquad a_2 = 0{\cdot}300 \pm 0{\cdot}005,$$
$$a_3 = 6{\cdot}17 \pm 0{\cdot}02, \qquad a_4 = -2{\cdot}0 \pm 0{\cdot}1.$$

The computation is *rounded*-interval arithmetic (again *assuming three-decimal-digit floating-point machine arithmetic*) would proceed as follows. (We represent the numbers here in an equivalent fixed-point notation.)

Put

$$X = [0{\cdot}451, 0{\cdot}453],$$
$$A_1 = [0{\cdot}199, 0{\cdot}201],$$
$$A_2 = [0{\cdot}295, 0{\cdot}305],$$
$$A_3 = [6{\cdot}15, 6{\cdot}19],$$
$$A_4 = [-2{\cdot}10, -1{\cdot}90];$$

then

$$X^2 = [0{\cdot}203, 0{\cdot}206],$$
$$A_4 X^2 = [-0{\cdot}433, -0{\cdot}385],$$
$$A_3 + A_4 X^2 = [5{\cdot}71, 5{\cdot}81],$$
$$A_2 X = [0{\cdot}133, 0{\cdot}139],$$
$$A_1 + A_2 X = [0{\cdot}332, 0{\cdot}340],$$
$$Y = \frac{A_1 + A_2 X}{A_3 + A_4 X^2} = [0{\cdot}0574, 0{\cdot}0599].$$

We know that Y must contain y, thus

$$0{\cdot}0574 \leqslant y \leqslant 0{\cdot}0599,$$

or we could write (averaging end points)

$$y = 0{\cdot}0586 \pm 0{\cdot}0013.$$

Computational problems in algebra—even as simple as that of calculating $\sqrt{2}$—often require *infinite* sequences of arithmetic operations for their *exact* solution. We cannot, and the computing machine cannot, execute infinitely many arithmetic operations during a finite interval of time; so we must approximate the limiting result by truncating the infinite sequence after some finite number of steps. This is the source of the third kind of error in computation.

In problems in linear algebra, such as matrix inversion, there are finite procedures, so-called 'direct methods' such as Gaussian elimination and indirect methods such as iterative procedures. It is sometimes better to use indirect methods, even at the cost of introducing truncation error, in order to sharpen the computed bounds on the other kinds of error.

The evaluation of any finite sequence of interval-arithmetic operations produces upper and lower bounds on the range of values of the same computation in real arithmetic for any choice of real numbers in the initial intervals. Rational expressions for truncation errors can be bounded in this way. Irrational expressions for truncation errors can often be bounded, making use of special information about the functions concerned.

One can often obtain rational interval functions, for example interval polynomials, which contain a given irrational function.

For example, consider the Taylor series with remainder in mean-value form for the exponential function with *negative* real argument:

$$e^x = 1+x+\frac{x^2}{2!}+\ldots+\frac{x^{k-1}}{(k-1)!}+\frac{e^t x^k}{k!}$$

for some $t \in [x, 0]$ where $x < 0$.

We have $e^t \in [e^x, e^0] \subset [0, 1]$ for $t \in [x, 0]$ and $x < 0$.

Therefore, for every positive integer k, and all $x \leqslant 0$, the exponential function is contained in the interval polynomial

$$Q_k(x) = 1+x+\frac{x^2}{2!}+\ldots+\frac{x^{k-1}}{(k-1)!}+[0, 1]\frac{x^k}{k!}.$$

We have $e^x \in Q_k(x)$ for $k = 1, 2, \ldots$, and $x \leqslant 0$.

If we evaluate $Q_k(x)$ in rounded-interval arithmetic on the computer we will obtain an interval containing the exact value of e^x, for negative x. The width of this interval can be made arbitrarily small by taking k large enough and by carrying enough digits on the machine. Of course, for large negative x there are more efficient ways to do the same thing.

In order to bound the range of values of e^x when $x \in [x_1, x_2]$ with $x_1 < x_2 \leqslant 0$ we *could* simply compute $Q_k([x_1, x_2])$ since

$$e^x \in Q_k([x_1, x_2]) \quad \text{for all } x \in [x_1, x_2].$$

The width of the bounding interval $Q_k([x_1, x_2])$ obtained in this way will be slightly greater (carrying enough digits) than $e^{(x_2-x_1)}-1$. If x_2-x_1 is small, then this will be a narrow interval. On the other hand for wide intervals $[x_1, x_2]$ we can make use of the monotonicity of the exponential function and notice that $x_1 < x_2 \leqslant 0$ implies

$$e^{x_1} < e^{x_2} \leqslant 1.$$

Then we can compute $Q_k(x_1)$ and $Q_k(x_2)$ and we shall obtain

$$e^{x_1} \in Q_k(x_1) = [a_1(x_1), b_1(x_1)]$$

and

$$e^{x_2} \in Q_k(x_2) = [a_2(x_2), b_2(x_2)]$$

and we will have for all $x \in [x_1, x_2]$ with $x_2 \leqslant 0$,

$$e^x \in [e^{x_1}, e^{x_2}] \subset [a_1(x_1), b_2(x_2)].$$

For large enough k, and carrying enough digits, this bounding interval, $[a_1(x_1), b_2(x_2)]$, will have a width greater than the width, $e^{x_2}-e^{x_1}$, of the actual range of values by an arbitrarily small amount.

We conclude this introduction with an application of these remarks to the problem of finding a zero of the function $f(x) = e^x+x$ with rigorous error-bounding.

The given function surely has a root in $[-1, 0]$ since it is continuous and $f(-1) = e^{-1}-1 < 0$ whereas $f(0) = e^0+0 = 1 > 0$. We shall not attempt to decide whether this observation should be considered as art or science. In any case let us pass on to the next step.

From the mean value theorem, we have (since f is continuously differentiable)

$$f(x) = f(y)+f'(\xi)(x-y)$$

for some ξ between x and y. Suppose $f(x) = 0$ and suppose that both x and y are in some interval $[a, b]$; then so also is ξ in $[a, b]$ and we can write

$$x = y-\frac{f(y)}{f'(\xi)} \subset y-\frac{f(y)}{F'([a,b])},$$

where $F'([a, b])$ is an interval function containing the range of values of $f'(\xi)$ when $\xi \in [a, b]$. For the problem at hand, $f'(\xi) = e^\xi+1$. We can get improved bounds on the root using even the crude bounds on the range of values of f' given by

$$f'(\xi) \in Q_2([a,b])+1$$

for $\xi \in [a, b]$ and $a < b \leqslant 0$ with

$$Q_2([a,b]) = 1+[a,b]+\tfrac{1}{2}[0,1][a,b]^2.$$

If x and y are in $[a, b] \subset [-1, 0]$, then so is ξ in $[a, b]$ and we have

$$x = y-\frac{f(y)}{f'(\xi)} \in y-\frac{f(y)}{Q_2([a,b])+1} = y-\frac{Q_2(y)+y}{Q_2([a,b])+1}.$$

Now take $[a, b] = [-1, 0]$ and $y = \frac{1}{2}(a+b) = -\frac{1}{2}$, then x and y are in $[-1, 0]$ and

$$Q_2(y) = Q_2(-\tfrac{1}{2}) = 1-\tfrac{1}{2}+\tfrac{1}{2}[0,1](-\tfrac{1}{2})^2 = [\tfrac{1}{2}, \tfrac{5}{8}],$$

$$Q_2([a,b]) = Q_2([-1,0]) = 1+[-1,0]+\tfrac{1}{2}[0,1][-1,0]^2 = [0, \tfrac{3}{2}].$$

Therefore

$$x \in y-\frac{Q_2(y)+y}{Q_2([a,b])+1} = -\frac{1}{2}-\frac{[\frac{1}{2}, \frac{5}{8}]+(-\frac{1}{2})}{[0, \frac{3}{2}]+1}.$$

Carrying out the interval arithmetic, we obtain

$$x \in [-\tfrac{5}{8}, -\tfrac{1}{2}].$$

We now know that a zero of e^x+x is in the narrower interval $[-\frac{5}{8}, -\frac{1}{2}]$. The procedure we used amounts to an interval version of Newton's root-finding method. We could iterate the process and obtain a sequence of intervals each containing the root. The sequence would not converge to an interval of zero width, of course, unless we allow k, the number of terms in our interval approximation to e^x, to increase as the iteration continues.

Chapters 2 and 3 by Nickel contain further work on interval methods for such problems. For some related work, see [1].

Some work has been done also on an n-dimensional interval version of Newton's method for systems of non-linear algebraic equations. See [2], [3], and [4].

REFERENCES

1. Dargel, R. H., Loscalzo, F. R., and Witt, T. H., Automatic error bounds on real zeros of functions. *Communs Ass. comput. Mach.* **9**, 806–9 (1966).

2. Hansen, Eldon, On solving systems of equations using interval arithmetic. *Math. Comput.* **22**, 374–84 (1968).

3. Moore, R. E. *Interval analysis.* Prentice-Hall, New Jersey (1966).

4. —— *Practical aspects of interval computation.* Aplikace matematiky, Prague **13**, 52–92 (1968).

2 · Triplex-Algol and Applications

1. Introduction

In this chapter we present a survey of triplex-Algol 60 and some examples showing how to work with triplex-Algol. As one example, a Newton-algorithm in triplex notation is given which always converges.

In 1966 a team of mathematicians at the University (*Technische Hochschule*) of Karlsruhe, Germany, started the triplex-project.† The triplex group consists of members of the *Lehrstuhl für Numerische Mathematik und Grossrechenanlagen* and of the *Rechenzentrum*. Since then the following three subjects have been treated:

(1) development of the new algorithmic language 'triplex' and the exact definition in Backus-Naur-form as 'triplex-Algol 60',
(2) realization of a triplex-Algol compiler,
(3) description of a number of triplex-Algol algorithms for the numerical solution of mathematical problems.

Some of the results are given in references [1] and [5]–[14].

2. What is triplex?

The triplex language is an extension of one of the common algorithmic languages such as Fortran, PL/1, or Algol 60. The extension consists of the addition of a new kind of variable of the type *'triplex'* to the *'integer'*, *'real'*, *'Boolean'*,... variables. The triplex numbers are an extension of Moore's interval numbers (see [3]) and the triplex arithmetic contains Moore's interval arithmetic (see [3]).

Until now, only the triplex-Algol 60 language has been defined (see [1]). A triplex-Algol 60 compiler has been working since May 1967 for the computer ZUSE Z 23 (see [9]–[14]). A compiler for the computer Electrologica X 8 is under construction. Sub-routines for triplex arithmetic in PL/1 have been written [2]. In what follows only the completely defined triplex-Algol 60 language will be considered.

† Previously called '*Fehlerschranken-Zahlen-Projekt*'.

3. Why triplex-Algol?

Triplex-Algol is a formalized language containing Moore's interval-analysis. Therefore it is possible to define each 'interval-algorithm' in that general language *universally*, i.e. independently of code procedures, a machine language, or special computer characteristics.

4. Definition of triplex-Algol 60

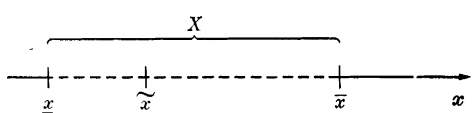
A detailed description of triplex-Algol is given in the literature as indicated in section 2 above. For our present purposes, we give the following definition in extract form:

Types

integer	also the combinations
triplex	*integer array, triplex array,...*
real	*integer procedure, triplex procedure,...*
Boolean	are permitted.

Triplex numbers

$X := [\underline{x}, \tilde{x}, \bar{x}]$† (see figure).



A triplex number X is an entity. The values of $\underline{x}$, $\tilde{x}$, $\bar{x}$ are of type *real*. Their meaning is: $\underline{x}$, $\bar{x}$ = lower, upper bound; $\tilde{x}$ = main value. The relations $\underline{x} \leqslant \tilde{x} \leqslant \bar{x}$ must hold.

Arithmetic

If $*$ denotes one of the operators $(+, -, \times, /)$ then for triplex numbers X and Y, by definition:

$$Z := X * Y := \{x * y : x \in X, y \in Y\}, \quad \text{where } 0 \notin Y \text{ for } * = /.$$

The main value is defined by $\tilde{z} := \tilde{x} * \tilde{y}$, using the ordinary *real* (i.e. floating-point) arithmetic (permanence principle, see [8]). The round-off errors are included.

Transfer functions

$$\underline{x} := \inf(X), \qquad X := \text{compose}(\underline{x}, \tilde{x}, \bar{x}),$$
$$\tilde{x} := \text{main}(X),$$
$$\bar{x} := \sup(X).$$

† Here and in what follows we denote *triplex* variables by capital letters, *integer* and *real* variables by small letters.

Relational operators $<, =, >$.

Relational operators are defined in terms of the bounds (not the main value) of a triplex number:

$$X < Y: \bar{x} < \underline{y},$$
$$X = Y: \underline{x} = \underline{y} \wedge \bar{x} = \bar{y},$$
$$X > Y: \underline{x} > \bar{y}.$$

Also the operators $\leqslant$, $\geqslant$, $\neq$ have been defined.

Input/output

Input/output includes rounding if a decimal–binary conversion is necessary.

Standard sub-routines

Standard sub-routines for triplex arguments are defined, too. For example,

signum:
$$\operatorname{sign}(X): = \begin{cases} 1 & \text{if } 0 < X, \\ 0 & \text{if } 0 \in X, \\ -1 & \text{if } 0 > X. \end{cases}$$

absolute value:
$$\operatorname{abs}(X): = \begin{cases} X \text{ if } \operatorname{sign}(X) = 1, \\ -X \text{ if } \operatorname{sign}(X) = -1, \\ [0, \operatorname{abs}(\tilde{x}), \max(-\underline{x}, \bar{x})] \text{ if } \operatorname{sign}(X) = 0. \end{cases}$$

intersection:
$$\operatorname{intsct}(X, Y): = \begin{cases} [\max(\underline{x}, \underline{y}), \zeta, \min(\bar{x}, \bar{y})] \text{ if } \operatorname{sign}(X-Y) = 0, \dagger \\ \text{not defined if } \operatorname{sign}(X-Y) \neq 0. \end{cases}$$

5. Reasons for the use of interval numbers

In section 6 we shall explain why the triplex-Algol system was developed. To this end, we now consider reasons why interval numbers are used. The first reason is that mathematical problems cannot always be solved in floating-point arithmetic because of round-off errors. For example, the equation

$$3x = 1$$

has the unique solution

$$\hat{x} = 1/3 = 0{\cdot}333...,$$

which cannot be written as a floating-point number of finite length in decimal or binary notation.

In interval numbers the solution, to five decimal digits, is

$$X = [0{\cdot}33333, 0{\cdot}33334];$$

i.e. the solution X can be written by means of floating-point numbers.

† ζ is arbitrary but lies between the bounds of intsct(X, Y).

The second reason is that intervals (more generally, sets) often arise naturally in mathematics. For example, assume $\hat{x} < \tilde{x}$. The mean value theorem,

$$f(\hat{x}) = f(\tilde{x}) + (\hat{x} - \tilde{x}) f'(\xi) \quad (\hat{x} \leqslant \xi \leqslant \tilde{x}),$$

cannot be treated numerically in real numbers, because ξ is unknown. But the relation

$$f(\hat{x}) \subseteq f(\tilde{x}) + (\hat{x} - \tilde{x}) f'(X)$$

with $X := [\hat{x}, \tilde{x}]$ can be used *numerically* with the aid of interval analysis.

6. Reasons for the use of triplex numbers

We now consider reasons why the use of triplex numbers is preferred to the use of interval numbers. The first reason is that storing the numbers is simpler. For example, the number $\pi := 3{\cdot}14159265\ldots$ written in interval form, using nine significant decimal digits, is

$$\pi := [3{\cdot}14159265,\ 3{\cdot}14159266].$$

To store this interval number (without sign) requires eighteen decimal digits. Similar information can be expressed in the form

$$\pi := 3{\cdot}14159265 \pm 1 \times 10^{-8}$$

which (without sign) requires eleven digits. In the triplex case, we can use the latter form. That is, we need not store the numbers $\underline{x}$, $\tilde{x}$, and $\bar{x}$ to represent $X = [\underline{x}, \tilde{x}, \bar{x}]$. Instead, we can use the equivalent form $(\tilde{x}, \tilde{x} - \underline{x}, \bar{x} - \tilde{x})$ or even $(\tilde{x}, \max(\tilde{x} - \underline{x}, \bar{x} - \tilde{x}))$.

The second reason is that the arithmetic is faster. For example, to add the intervals

$$e := [2{\cdot}71828182, 2{\cdot}71828183] \text{ and } \pi := [3{\cdot}14159265, 3{\cdot}14159266]$$

requires eighteen digit-to-digit additions (ignoring carry) to get

$$e + \pi := [5{\cdot}85987447, 5{\cdot}85987449].$$

If a triplex number X is written in the form $X := \tilde{x} - (\tilde{x} - \underline{x}) + (\bar{x} - \tilde{x})$, we have

$$e := 2{\cdot}81728182 - 0 + 1 \times 10^{-8} \quad \text{and} \quad \pi := 3{\cdot}14159265 - 0 + 1 \times 10^{-8}$$

so that $e + \pi := 5{\cdot}85987447 - 0 + 2 \times 10^{-8}$. Here only ten digit-to-digit additions (ignoring carry) are required. This is important for 'long numbers'.

The third reason is that there is no loss of information if the bounds are pessimistic. After a great number of arithmetic operations it may be that, in floating-point arithmetic, the result is $\tilde{x} = 7{\cdot}12 \times 10^4$ while an interval analysis gives the result $X = [-3{\cdot}74 \times 10^3, +7{\cdot}51 \times 10^4]$. In

this case the interval notation is practically meaningless. But it may be that the approximation $\tilde{x}$ is not too bad. Printing only the interval X means throwing away the previous information $\tilde{x}$. In triplex that information is conserved and printed out too.

The fourth reason is that many algorithms in numerical mathematics need both the main value $\tilde{x}$ plus a corresponding interval $[\underline{x}, \bar{x}]$ containing $\tilde{x}$. Therefore in triplex the combination $X: = [\underline{x}, \tilde{x}, \bar{x}]$ is used. In what follows the Newton procedure is an example of such an algorithm.

7. Examples, working in triplex-Algol

In the following examples the problem of finding a root of

$$f(x): = x - \frac{1-x^2}{3+x^2} = 0$$

is solved in several ways.

7.1. *Iterative solution*

Let $$\varphi(x): = \frac{1-x^2}{3+x^2}.$$

Because $$|\varphi'(x)| = \left|\frac{8x}{(3+x^2)^2}\right| \leqslant \tfrac{1}{2} < 1,$$

the iteration

$$x_0 \text{ arbitrary}, \; x_{n+1}: = \varphi(x_n) \text{ for } n = 0, 1, 2,...$$

leads to an always converging sequence $\{x_n\}$. Replacing the *real* numbers x_n by *triplex* numbers X_n gives the iteration formula

$$X_0 \text{ arbitrary}, \; X_{n+1}: = \varphi(X_n) \text{ for } n = 0, 1,....$$

The following is a triplex-Algol-program† for this iterative algorithm. It was stopped manually.

```
'BEGIN'
        'INTEGER' N;
        'TRIPLEX' X;
        N: = 0;
        X: = [-1, 0, +1];
LABEL:  PRINT (N, X);
        N: = N+1;
        X: = (1-X 'POWER' 2)/(3+X 'POWER' 2);
        'GOTO' LABEL
'END'
```

† The program is in the Alcor notation. For details and programming examples see [4].

The results† are

N	X		
	$\underline{x}$	$\tilde{x}$	$\bar{x}$
0	$-0{\cdot}100000000\times10^{1}$	0·0	$0{\cdot}100000000\times10^{1}$
1	·0	·333333334	·333333334
2	·285714284	·285714286	·333333334
3	·285714284	·298013245	·298013246
4	·294996307	·294996309	·298013246
5	·294996307	·295746818	·295746820
6	·295560749	·295560750	·295746820
7	·295560749	·295606920	·295606921
8	·295595465	·295595466	·295606921
9	·295595465	·295598307	·295598309
10	·295597601	·295597602	·295598309
11	·295597601	·295597777	·295597779
12	·295597733	·295597734	·295597779
13	·295597733	·295597745	·295597746
14	·295597741	·295597742	·295597746
15	·295597741	·295597743	·295597744
16	·295597741	·295597743	·295597744

The convergence is linear. The main value and the bounds both have approximately the same speed of convergence.

7.2. *Naïve Newton method*

In order to get a better rate of convergence (quadratic instead of linear) the well-known Newton method is often used. In real analysis the algorithm is

$$x_0 \text{ arbitrary}, \quad x_{n+1} := x_n - f(x_n)/f'(x_n) \text{ for } n = 0, 1, \ldots,$$

where

$$f'(x) := 1 + \frac{8x}{(3+x^2)^2}.$$

Replacing the *real* numbers x_n by *triplex* numbers X_n gives the (naïve) algorithm:

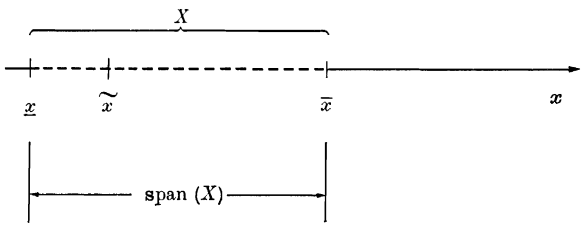
$$X_0 \text{ arbitrary}, \quad X_{n+1} := X_n - f(X_n)/f'(X_n) \text{ for } n = 0, 1, \ldots.$$

Programming this algorithm gives the astonishing result that the resulting sequence X_n is always divergent for an *arbitrary* initial value $X_0 \neq \hat{x}$!

† Computed with the computer ZUSE Z 23 with the triplex-Algol compiler of Dr. Wippermann (see [10]–[14]).

What is the reason for this behaviour? Before answering that question, let us introduce the real-valued function†

$$\text{span}(X) = \text{span}([\underline{x}, \tilde{x}, \bar{x}]) := \bar{x} - \underline{x} \quad \text{(see figure)}.$$



The span has the following properties:

$$\text{span}(X) \geqslant 0,$$
$$\text{span}(\alpha) = 0 \quad \text{if } \alpha \text{ is } \textit{real},$$
$$\text{span}(\alpha X) = |\alpha| \,\text{span}(X) \quad (\alpha \textit{ real}),$$
$$\text{span}(X \pm Y) = \text{span}(X) + \text{span}(Y).$$

With this definition, one sees that for $X_0 \neq \hat{x}$ and $f \not\equiv 0$,

$$\text{span}(X_{n+1}) = \text{span}(X_n) + \text{span}(f(X_n)/f'(X_n)) > \text{span}(X_n).$$

Therefore the convergence of the bounds of X_n is impossible.

7.3. *Pseudo-Newton method*

Rewriting

$$x - \frac{f}{f'} = x - \frac{x-\varphi}{1-\varphi'} = \frac{x - x\varphi' - x + \varphi}{1-\varphi'} = \frac{\varphi - x\varphi'}{1-\varphi'},$$

one may hope to get a convergent Newton method. The following two examples show two possibilities:

(*a*) $$x - \frac{f}{f'} = \frac{3 + 6x^2 - x^4}{4 + 4(1+x)^2 + (1+x^2)^2}.$$

This formulation leads to the triplex-Algol program.‡

```
'BEGIN'
        'INTEGER' N;
        'TRIPLEX' X, Y;
        N := 0;
        X := [-1, 0, 1];
```

† Moore and Hansen (see Chapters 1, 5, and 11) call this value 'width' and denote it by $w(X)$.

‡ The program is in the Alcor notation. For details and programming examples see [4].

```
LABEL: PRINT (N, X);
       N := N+1;
       Y := X 'POWER' 2;
       X := (3+6×Y−Y 'POWER' 2)/
            (4+4×(1+X) 'POWER' 2+(1+Y) 'POWER' 2);
       'GOTO' LABEL;
'END'
```

This program yields the following results:†

N	X		
	$\underline{x}$	$\tilde{x}$	$\bar{x}$
0	$-0{\cdot}100000000\times10^{1}$	0·0	$0{\cdot}100000000\times10^{1}$
1	$\cdot833333332\times10^{-1}$	·333333334	$\cdot180000001\times10^{1}$
2	$-\cdot767989327$	·296000000	$\cdot231139988\times10^{1}$
3	$-\cdot489769912\times10^{1}$	·295597791	$\cdot672162869\times10^{1}$
4	$-\cdot407651761\times10^{3}$	·295597743	$\cdot548163506\times10^{2}$
5	$-\cdot552315567\times10^{10}$	·295597743	$\cdot199416550\times10^{6}$

The program stopped because of overflow. The main value is—as expected—quadratically convergent; the bounds are still divergent.

(*b*)
$$x-\frac{f}{f'}=\frac{4+4x^2-(1-x^2)^2}{4+4(1+x)^2+(1+x^2)^2}.$$

This formulation gives the following triplex-Algol program‡ (which was stopped manually):

```
'BEGIN'
       'INTEGER' N;
       'TRIPLEX' X, Y;
       N := 0;
       X := [−1, 0, 1];
LABEL: PRINT (N, X);
       N := N+1;
       Y := X 'POWER' 2;
       X := (4+4×Y−(1−Y) 'POWER' 2)/
            (4+4×(1+X) 'POWER' 2+(1+Y) 'POWER' 2);
       'GOTO' LABEL
'END'
```

† Computed with the computer ZUSE Z 23 with the triplex-Algol compiler of Dr. Wippermann ([10]–[14]).

‡ The program is in the Alcor notation. For details and programming examples see [4].

The numerical results† are:

N	X		
	$\underline{x}$	$\tilde{x}$	$\bar{x}$
0	$-0{\cdot}100000000\times10^{1}$	$0{\cdot}0$	$0{\cdot}100000000\times10^{1}$
1	$\cdot125000000$	$\cdot333333334$	$\cdot160000001\times10^{1}$
2	$\cdot372630021\times10^{-1}$	$\cdot296000000$	$\cdot141073988\times10^{1}$
3	$\cdot831308385\times10^{-1}$	$\cdot295597791$	$\cdot128521236\times10^{1}$
4	$\cdot952804747\times10^{-1}$	$\cdot295597743$	$\cdot109277484\times10^{1}$
5	$\cdot115990216$	$\cdot295597743$	$\cdot894041905$
6	$\cdot142702889$	$\cdot295597743$	$\cdot715065710$
7	$\cdot172952331$	$\cdot295597743$	$\cdot565699620$
8	$\cdot204435240$	$\cdot295597743$	$\cdot456048028$
9	$\cdot233079488$	$\cdot295597743$	$\cdot386170705$
10	$\cdot255490542$	$\cdot295597743$	$\cdot345960847$
11	$\cdot270988171$	$\cdot295597743$	$\cdot323764036$
12	$\cdot280883393$	$\cdot295597743$	$\cdot311518896$
13	$\cdot286924555$	$\cdot295597743$	$\cdot304677140$
14	$\cdot290525348$	$\cdot295597743$	$\cdot300806489$
15	$\cdot292644115$	$\cdot295597743$	$\cdot298597097$
16	$\cdot293882076$	$\cdot295597743$	$\cdot297328739$
17	$\cdot294602557$	$\cdot295597743$	$\cdot296598067$
18	$\cdot295020938$	$\cdot295597743$	$\cdot296176268$
19	$\cdot295263584$	$\cdot295597743$	$\cdot295932479$
20	$\cdot295404206$	$\cdot295597743$	$\cdot295791474$
21	$\cdot295485667$	$\cdot295597743$	$\cdot295709883$
22	$\cdot295532845$	$\cdot295597743$	$\cdot295662661$
23	$\cdot295560165$	$\cdot295597743$	$\cdot295635327$
24	$\cdot295575984$	$\cdot295597743$	$\cdot295619504$
25	$\cdot295585143$	$\cdot295597743$	$\cdot295610343$
26	$\cdot295590446$	$\cdot295597743$	$\cdot295605039$
27	$\cdot295593516$	$\cdot295597743$	$\cdot295601968$
28	$\cdot295595294$	$\cdot295597743$	$\cdot295600191$
29	$\cdot295596323$	$\cdot295597743$	$\cdot295599162$
30	$\cdot295596919$	$\cdot295597743$	$\cdot295598566$
31	$\cdot295597264$	$\cdot295597743$	$\cdot295598221$
32	$\cdot295597463$	$\cdot295597743$	$\cdot295598020$
33	$\cdot295597580$	$\cdot295597743$	$\cdot295597905$
34	$\cdot295597647$	$\cdot295597743$	$\cdot295597838$
35	$\cdot295597685$	$\cdot295597743$	$\cdot295597800$
36	$\cdot295597708$	$\cdot295597743$	$\cdot295597777$
37	$\cdot295597721$	$\cdot295597743$	$\cdot295597765$
38	$\cdot295597728$	$\cdot295597743$	$\cdot295597757$
39	$\cdot295597732$	$\cdot295597743$	$\cdot295597753$
40	$\cdot295597736$	$\cdot295597743$	$\cdot295597750$
41	$\cdot295597737$	$\cdot295597743$	$\cdot295597748$
42	$\cdot295597737$	$\cdot295597743$	$\cdot295597747$
43	$\cdot295597738$	$\cdot295597743$	$\cdot295597747$
44	$\cdot295597738$	$\cdot295597743$	$\cdot295597747$

† Computed with the computer ZUSE Z 23 with the triplex-Algol compiler of Dr. Wippermann ([10]–[14]).

The main value is again quadratically convergent, the bounds are now convergent—but only linearly. In fact, the number of iterations necessary to obtain the final result is much higher (43 compared with 15) than in the case of the simple iteration method (see section 7.1). Also the computing time is greater because of the higher number of arithmetic operations needed. Therefore this pseudo-Newton method is too poor to be used.

7.4. *Newton method*

Let $\hat{x}$ be the solution of $f(x) = 0$ and $\tilde{x}$ an arbitrary real number. Then by the mean value theorem there exists a number $\xi = \xi(\hat{x}, \tilde{x})$ such that $0 = f(\hat{x}) = f(\tilde{x}) + (\hat{x} - \tilde{x}) f'(\xi)$. Let $X := [\underline{x}, \tilde{x}, \bar{x}]$ be a triplex number containing $\hat{x}$. Then $\xi \in X$ and therefore

$$\hat{x} = \tilde{x} + f(\tilde{x})/f'(\xi) \in \tilde{x} + f(\tilde{x})/f'(X)$$

if $0 \notin f'(X)$. Therefore we now define the following triplex Newton algorithm:

$$X_0 \text{ arbitrary, but containing } \hat{x},$$

$$X_{n+1} := \tilde{x}_n + f(\tilde{x}_n)/f'(X_n) \quad \text{for } n = 0, 1, \ldots.$$

If $\tilde{x}_n = \text{main}(X_n)$ is quadratically convergent and $f'(X_n)$ is bounded, then X_n is obviously also (quadratically) convergent, because

$$\begin{aligned} \text{span}(X_{n+1} - \hat{x}) &= \text{span}(\tilde{x}_n - \hat{x} + f(\tilde{x}_n)/f'(X_n)) \\ &= |f(\tilde{x}_n)| \, \text{span}(1/f'(X_n)) \to 0. \end{aligned}$$

This algorithm was first given by Moore (see [3], p. 64) with $(\bar{x}_n + \underline{x}_n)/2$ instead of $\tilde{x}_n$.

In this example it can be clearly seen that the use of the real 'main value' $\tilde{x}_n$ *together* with the interval X_n is very often quite 'natural' for numerical algorithms. This was one of the main reasons for introducing the triplex numbers replacing interval numbers.

A triplex-Algol program for this algorithm is:†

```
'BEGIN'    'INTEGER' N;
           'TRIPLEX' X, Y;
'TRIPLEX'  'PROCEDURE' F(Y);
           'TRIPLEX' Y;
           'BEGIN' 'TRIPLEX' Z;
                   Z := Y;
                   Y := Y 'POWER' 2;
                   F := Z-(1-Y)/(3+Y)
           'END';
```

† The program is in the Alcor notation. For details and programming examples see [4].

```
'TRIPLEX' 'PROCEDURE' F PRIME (Y);
          'TRIPLEX' Y;
          F PRIME := 1+8×Y/((3+Y 'POWER' 2) 'POWER' 2);
          N := 0;
          X := [−1, 0, +1];
LABEL:    PRINT (N, X);
          N := N+1;
          Y := COMPOSE (MAIN (X), MAIN (X), MAIN (X));
          X := Y−F(Y)/F PRIME (X);
          'GOTO' LABEL
'END'
```

The program was stopped manually. The numerical results† were:

	X		
N	$\underline{x}$	$\tilde{x}$	$\bar{x}$
0	$-0{\cdot}100000000\times10^{1}$	0·0	$0{\cdot}100000000\times10^{1}$
1	·176470587	·333333334	$\cdot300000001\times10^{1}$
2	·286176605	·296000000	·320150327
3	·295594364	·295597791	·295604528
4	·295597741	·295597743	·295597744
5	·295597741	·295597743	·295597743
6	·295597741	·295597743	·295597743

These results are very satisfactory. The convergence is now obviously quadratic for both main value and bounds. It is interesting to note that after one iteration the upper bound $\bar{x}_1$ is outside of X_0.

7.5. *Generalized Newton method*

THEOREM. *Let* $-\infty < a < b < \infty$, $I := [a,b]$, $f(x) \in C_1(I)$ *or* $C_2(I)$, *and let* $F'(X)$ *be a continuous triplex extension of* $f'(x)$‡ *with the property* $0 \notin F'(X)$ *for* $X \subset I$. *Then the following generalized Newton algorithm is always super-linear or quadratically convergent*:

$$X_0 \in I \text{ arbitrary}$$
$$X_{n+1} := [\tilde{x}_n - f(\tilde{x}_n)/F'(X_n)] \cap X_n.$$

The *proof* is not difficult and will not be given here. The fact that this algorithm is quadratically convergent if span (X_0) is sufficiently small has been given by Moore ([3]).

A realization of this algorithm in triplex-Algol applied to the function $f(x) := x/(1+|x|), f'(x) := 1/(1+|x|)^2, \hat{x} = 0$ in the interval $[-10^6, 10^6]$ with the initial value $X_0 := [-7, 4711, 247921]$ gives the following program:

† Computed with the compiler ZUSE Z 23 with the triplex-Algol compiler of Dr. Wippermann ([10]–[14]).

‡ i.e. $F'(X) \in C_1(I)$, $f'(\tilde{x}) \in F'(X)$ if $\tilde{x} \in X$, and $f'(\tilde{x}) = F'([\tilde{x}, \tilde{x}, \tilde{x}])$.

```
'BEGIN'   'INTEGER' N;
          'TRIPLEX' X, Y, Z, ZX;
          'REAL' R, ZM, ZXI, ZXS;
'TRIPLEX' 'PROCEDURE' F(X);
          'TRIPLEX' X;
          'BEGIN'
                    'REAL' XI, XS;
                    XI: = ABS (INF (X));
                    XS: = ABS (SUP (X));
                    'IF' XI 'GREATER' XS 'THEN' XS: = XI;
                    'IF' SIGN (X) 'EQUAL' 0 'THEN' XI: = 0;
                    F: = X/(1+COMPOSE (XI, ABS (MAIN (X)), XS));
          'END';
'TRIPLEX' 'PROCEDURE' F PRIME (X);
          'TRIPLEX' X;
          'BEGIN' 'TRIPLEX' Z;
                    'REAL' ZI, ZM, ZS, M;
                    'REAL' XI, XS;
                    XI: = ABS (INF (X));
                    XS: = ABS (SUP (X));
                    'IF' XI 'GREATER' XS 'THEN' XS: = XI;
                    'IF' SIGN (X) 'EQUAL' 0 'THEN' XI = 0;
                    M: = 1/(1+10**6) 'POWER' 2;
                    Z: = 1/((1+COMPOSE (XI, ABS (MAIN (X)), XS))
                                                    'POWER' 2);
                    ZI: = INF (Z);
                    ZM: = MAIN (Z);
                    ZS: = SUP (Z);
                    'IF' ZS 'GREATER' 1 'THEN' ZS: = 1;
                    'IF' ZM 'GREATER' 1 'THEN' ZM: = 1;
                    'IF' ZI 'LESS' M 'THEN' ZI: = M;
                    'IF' ZM 'LESS' M 'THEN' ZM: = M;
                    F PRIME: = COMPOSE (ZI, ZM, ZS);
          'END';
          X: = [-7, 4711, 247921];
          N: = 0;
L1:       PRINT (N, X);
          N: = N+1;
          'FOR' R: = MAIN (X), INF (X), SUP (X) 'DO'
```

```
'BEGIN' Y := COMPOSE (R, R, R);
        Z := Y−F(Y)/F PRIME (X);
        ZX := INTSCT (X, Z);
        ZM := MAIN (Z);
        ZXI := INF (ZX);
        ZXS := SUP (ZX);
        'IF' ZX 'EQUAL' X 'THEN' 'GOTO' L2;
        'IF' ZM 'NOT LESS' ZXI 'AND' ZM 'NOT GREATER'
                                                ZXS
                'THEN' X := COMPOSE (ZXI, ZM, ZXS)
                'ELSE' X := COMPOSE (ZXI, (ZXI+
                                        +ZXS)/2, ZXS);
        'GOTO' L1;
L2:
'END';
WRITE ("
");
        PRINT (N, X)
'END'
```

The results† were:

	X		
N	$\underline{x}$	$\tilde{x}$	$\bar{x}$
0	$-0{\cdot}700000000\times10^{1}$	$0{\cdot}471100000\times10^{4}$	$0{\cdot}247921000\times10^{6}$
1	$-{\cdot}700000000\times10^{1}$	${\cdot}235150011\times10^{4}$	${\cdot}471000022\times10^{4}$
2	$-{\cdot}700000000\times10^{1}$	${\cdot}117175027\times10^{4}$	${\cdot}235050054\times10^{4}$
3	$-{\cdot}700000000\times10^{1}$	${\cdot}581875561\times10^{3}$	${\cdot}117075113\times10^{4}$
4	$-{\cdot}700000000\times10^{1}$	${\cdot}286938638\times10^{3}$	${\cdot}580877277\times10^{3}$
5	$-{\cdot}700000000\times10^{1}$	${\cdot}139471056\times10^{3}$	${\cdot}285942112\times10^{3}$
6	$-{\cdot}700000000\times10^{1}$	${\cdot}657390873\times10^{2}$	${\cdot}138478175\times10^{3}$
7	$-{\cdot}700000000\times10^{1}$	${\cdot}288770356\times10^{2}$	${\cdot}647540712\times10^{2}$
8	$-{\cdot}700000000\times10^{1}$	${\cdot}104552531\times10^{2}$	${\cdot}279105061\times10^{2}$
9	$-{\cdot}700000000\times10^{1}$	${\cdot}127127463\times10^{1}$	${\cdot}954254926\times10^{1}$
10	$-{\cdot}700000000\times10^{1}$	$-{\cdot}161613917\times10^{1}$	${\cdot}711556039$
11	$-{\cdot}998381836$	$-{\cdot}143412898$	${\cdot}711556039$
12	$-{\cdot}179876052\times10^{-1}$	${\cdot}205672593\times10^{-1}$	${\cdot}357476771$
13	$-{\cdot}165691241\times10^{-1}$	$-{\cdot}423012156\times10^{-3}$	${\cdot}414487324\times10^{-3}$
14	$-{\cdot}178863957\times10^{-6}$	${\cdot}178939445\times10^{-6}$	${\cdot}139491763\times10^{-4}$
15	$-{\cdot}496069853\times10^{-11}$	$-{\cdot}319744231\times10^{-13}$	${\cdot}324185124\times10^{-13}$
16	$-{\cdot}105879119\times10^{-21}$	0	${\cdot}211758238\times10^{-21}$
17	0	0	0
18	0	0	0

† Computed with the computer ZUSE Z 23 with the triplex-Algol compiler of Dr. Wippermann ([10]–[14]).

This example and the initial value were chosen such that the convergence is not very fast. The *real* Newton method for this function $f(x)$ is always *diverging* for $|x_0| \geqslant 1$. For $n = 0(1)11$ the convergence is only linear; then it becomes quadratic until $n = 15$ where the round-off errors slow down the rate of convergence. For $n = 16$ the main value becomes zero by underflow. Then the bounds for $n = 17$ are the exact value. The final result occurs for $n = 18$.

Additional note by E. Hansen

In order to apply the Newton method as described in section 7.4 or section 7.5, we require an initial bound on the root. It is frequently possible to obtain such a bound in the course of the computation. This is made possible by the following theorem:

THEOREM. *Assume* $f(x) \in C_1(X)$ *and* $0 \notin f'(X)$ *for some given interval* X. *Let* $\tilde{x}$ *be a point in* X *and define*

$$TX: = \tilde{x} - f(\tilde{x})/f'(X).$$

If $TX \subset X$, *then there exists a point* $\hat{x} \in TX$ *such that* $f(\hat{x}) = 0$.

Proof. Since $0 \notin f'(X)$, we can assume without loss of generality that $f'(X) > 0$. Denote $f'(X) = [c, d]$. If $f(\tilde{x}) = 0$, there is nothing to prove. If $f(\tilde{x}) > 0$, then since $f'(x) \in [c, d]$ for $x \in X$, the curve $y = f(x)$ lies below the line $y_1 = c(x - \tilde{x}) + f(\tilde{x})$ and above the line $y_2 = d(x - \tilde{x}) + f(\tilde{x})$ for $x < \tilde{x}$. Hence $f(x) = 0$ for some point in the interval

$$[\hat{x} - f(\tilde{x})/c, \quad \tilde{x} - f(\tilde{x})/d].$$

But this interval is contained in X since $TX \subset X$ by assumption. Hence $\hat{x} \in X$, which implies $\hat{x} \in TX$ (see Lemma 7.2 of [3]). A similar proof can be obtained when $f(\tilde{x}) < 0$.

In practice, rounding occurs and instead of TX we will obtain, say, $T_1X \supset TX$. If we find $T_1X \subset X$, this assures $\hat{x} \in T_1X$ since it implies $TX \subset X$.

It is of interest to note that $\hat{x}$ is not a fixed point of the mapping T (unless $\tilde{x} = \hat{x}$).

It has been shown by Professor W. Kahan (private communication) that the above theorem can be extended to the multidimensional case, where it is especially useful.

REFERENCES

1. APOSTOLATOS, N., KULISCH, U., KRAWCZYK, R., LORTZ, B., NICKEL, K., and WIPPERMANN, H.-W. The algorithmic language triplex-Algol 60. *Num. Math.* **11**, 175–80 (1968).

2. CSAJKA, I., MÜNZNER, W., and NICKEL, K. *Subroutines ADD, NEG, SUB, DIV, MUL for use in an 'error-bound arithmetic'*. IBM Research Laboratory, Säumerstrasse 4, 8803 Rüschlikon, Switzerland.

3. MOORE, R. E., *Interval Analysis*. Prentice Hall, New Jersey (1966).

4. NICKEL, K. *Algol Praktikum*. G. Braun-Verlag, Karlsruhe (1964).

5. —— Über die Notwendigkeit einer Fehlerschranken-Arithmetik für Rechenautomaten. *Num. Math.* **9**, 69–79 (1966).

6. —— Die vollautomatische Berechnung einer einfachen Nullstelle von $F(t) = 0$ einschließlich einer Fehlerabschätzung. *Computing*, **2**, 232–45 (1967).

7. —— Quadraturverfahren mit Fehlerschranken. *Computing*, **3**, 47–64 (1968).

8. —— Anwendungen einer Fehlerschranken-Arithmetik. To appear in *Numerische Mathematik, Differentialgleichungen, Approximationstheorie*. Internationale Schriftenreihe zur numerischen Mathematik, Birkhäuser, Basel (1968).

9. —— Zwei neue Rechenmaschinen-Systeme an der Universität (TH) Karlsruhe. HYDRA-X8 Triplex-Algol-Z 23. *Umschau* **67**, 525–6 (1967).

10. WIPPERMANN, H.-W. Realisierung einer Interval-Arithmetik in einem Algol-60-System. *Elektron. Rechenanl.* **9**, 224–33 (1967).

11. —— Definition von Schrankenzahlen in Triplex-Algol. *Computing*, **3**, 99–109 (1968).

12. —— Implementierung eines Algol-Systems mit Schrankenzahlen. *Elektron. Datenverarb.* **10**, 189–94 (1968).

13. —— Manual für das System Triplex-Algol Karlsruhe. *Ber. Inst. Angew. Math. Rechenzent. Univ. (TH) Karlsruhe*, April (1967).

14. —— Ein Algol-60 Compiler mit Triplex-Zahlen. *Z. angew. Math. Mech.* **47**, T76–T79 (1967).

3 · Zeros of Polynomials and Other Topics

1. Dependent intervals

LET $$f(x) := \sum_{\mu=0}^{m} a_\mu x^\mu \Big/ \sum_{\nu=0}^{n} b_\nu x^\nu$$

be a rational function with real rational coefficients a_μ and b_ν. Let $X := [\underline{x}, \bar{x}]$ be an interval and

$$F(X) := \{f(x) : x \in X\}.$$

In interval arithmetic or triplex arithmetic, the function $f(x)$ can be evaluated for x replaced by the interval X yielding

$$f(X) := \sum_{\mu=0}^{m} a_\mu X^\mu \Big/ \sum_{\nu=0}^{n} b_\nu X^\nu,$$

where the arithmetic operations are performed as machine-interval operations. It is well known that in general

$$f(X) \neq F(X),$$

but (see Moore [7]) the following is always true:

$$f(X) \supseteq F(X).$$

As an example, let

$$f(x) := 1 - x + x^2 - x^3 + x^4 - x^5 \quad \text{and} \quad X = [\underline{x}, \bar{x}] := [2, 3].$$

Now $f'(x) < 0$ for $x > -1$ and therefore

$$F(X) = [f(\bar{x}), f(\underline{x})] = [-182, -21],$$

while $$f(X) = [-252, +49] \supset F(X).$$

Very often the value of $f(X)$ obtained in this way depends on the kind of algorithm used, so that it is possible to get 'better' results just by rearranging the form of $f(x)$. To illustrate this, note that in this example,

the interval $X := [2, 3]$ gives the following different values for $f(X)$ if f is written in the forms shown:

$f(x)$	$f([2,3])$
$1-x+x^2-x^3+x^4-x^5$	$[-252, +49]$
$(1-x^6)/(1+x)$	$[-242\frac{2}{3}, -15\frac{3}{4}]$
$(1-x)(1+x^2+x^4)$	$[-182, -21]$

The last value is the correct one for $f(X)$.

Under certain conditions the equality

$$(*) \qquad f(X) = F(X)$$

holds. This is true, for example (even in the case in which the coefficients are interval numbers), if $X \geqslant 0$ and

(a) $n = 0$, i.e. $f(x) := \sum_{\mu=0}^{m} a_\mu x^\mu / b_0$, and $a_\mu \geqslant 0$ for $\mu = 0(1)m$;

(b) $a_0 > 0$, $a_\mu \leqslant 0$ for $\mu = 1(1)m$, and $\operatorname{sign} f(X) = \operatorname{sign} b_\nu$ for $\nu = 0(1)n$.

Unfortunately the relation (*) is in general false. This means that in general the result of an algorithm has bounds worse than is desired. This is revealed by the following very simple example. Let $f(x) = x-x$. Then $f(x) \equiv 0$ and therefore $f(X) = 0$ for each $X = [\underline{x}, \bar{x}]$. But (see section 7.2 of Chapter 2)

$$\operatorname{span} f(X) = \operatorname{span}(X-X) = 2 \operatorname{span} X \geqslant 0$$

where equality holds if and only if X is a real number.

It is very important to learn why (*) is not true in general, i.e. why and when such a loss of exactitude occurs. The reason for this behaviour was first pointed out in full detail by Apostolatos and Kulisch ([1]–[3]). In order to discuss this reason, we make the following definition. Two intervals $X = [\underline{x}, \bar{x}]$ and $Y = [\underline{y}, \bar{y}]$ are called *dependent*, if there exists a point-wise relation $X \leftrightarrow Y$, i.e. if each point $x \in X$ belongs to a point $y \in Y$ and vice versa. In other words, there are two functions $\xi(t)$, $\eta(t)$ defined on $0 \leqslant t \leqslant 1$ such that

$$X := \{\xi(t) : 0 \leqslant t \leqslant 1\}, \qquad Y := \{\eta(t) : 0 \leqslant t \leqslant 1\}.$$

Some examples of dependent intervals are:

(a) $X = X$, i.e. each interval X depends on itself,
(b) $Y := X^2$, $Y := X(1-X)$, etc.,
(c) $X := \{t(1-t^2) : 0 \leqslant t \leqslant 1\}$, $Y := \{t^2(1-t) : 0 \leqslant t \leqslant 1\}$.

Let $*$ denote one of the arithmetic operations $+$, $-$, $\times$, or $/$. For dependent intervals, the 'natural' definition of $X * Y$ becomes

$$X \overset{d}{*} Y := \{\xi(t) * \eta(t) : 0 \leqslant t \leqslant 1\},$$

where, in the case of the division, $\eta(t) \neq 0$ for any $t \in [0, 1]$. Apostolatos and Kulisch call the arithmetic based on the operations $\overset{d}{*}$ and $*$ 'extended' (*erweiterte*) and 'simple' (*einfache*) interval arithmetic, respectively. Within the computer only the 'simple' interval arithmetic can be realized. Obviously the following inclusion always holds:

$$X \overset{d}{*} Y \subseteq X * Y.$$

For other properties and results, see the papers of Apostolatos and Kulisch ([1]–[3]).

In algorithms concerning dependent intervals the resulting bounds are in general not 'best', if the 'simple' interval arithmetic is used. For example, this difficulty arises if a polynomial with a triplex number as its argument is evaluated or if a polynomial is divided by a linear factor containing a triplex number. Both these operations are commonly done during the evaluation of all roots of a polynomial with computed error bounds. See section 5.

2. Application of the theory of dependent intervals to matrix problems

Let A be an $n \times n$ matrix and b an n vector. We consider the two following problems.

(A) Solve the linear system $\quad Ax = b.$

(B) Compute A^{-1}, i.e. solve the matrix-system

$$AX = I,$$

where I denotes the unit matrix. Both problems can be solved in a finite number of steps by the Gaussian elimination method. The number of arithmetic operations required is proportional† to n^3.

If this method is straightforwardly programmed using interval- or triplex-arithmetic the results are interval- or triplex-numbers containing the exact results, as must be expected. But a very interesting phenomenon is experimentally observed. If n is not too small (say $n > 10$) then the bounds for the results are in general *very* pessimistic. For the reason for this, see Moore [7], Hansen [4], or Hansen and Smith [5]. If $T := [\underline{t}, \tilde{t}, \bar{t}]$ is a typical triplex-component—say $T := x_i$ in problem (A)

† A multiplicative constant factor being suppressed.

or $T := X_{ik}$ in problem (B)—then it is not unusual to find $(\bar{t}-\underline{t})/|\tilde{t}| > 10^5$ or even 10^8, even in the case of well-conditioned matrices. What is the reason for this behaviour? It is interesting to see that it can be explained very simply with the aid of the idea of dependent intervals.

Let us first assume the coefficients a_{ik} of A are interval (triplex) numbers with non-zero span. There are n^2 of them. The computation of the solution of problem (A) or (B) requires about n^3 operations.† That means that, on the average, each element a_{ik} of A is used about n times.† Because each interval is dependent on itself this implies the use of a formula that is highly dependent upon the input data a_{ik}. Such dependency in general causes great loss of accuracy, as has been shown in section 1. Therefore in this case the occurrence of unrealistic error-bounds is explained.

Now the remaining case of a real-valued matrix A will be treated. In order to change A into a triangular matrix in the Gauss method the following algorithm is used: multiply the first row of A by $\mu_1 := -a_{21}/a_{11}$, $\mu_2 := -a_{31}/a_{11}$,... and add the result to the second, third,... row; then perform the same operations with the remaining $(n-1)\times(n-1)$ matrix, and so on. There are obviously $(n-1)+(n-2)+\ldots+1 = n(n-1)/2 \cong n^2$ such μ-factors,† giving a total of n^3 arithmetic operations,† each containing at least one μ-factor. So necessarily each μ-factor—being an interval because of round-off errors—has to be used several times, namely† n times. For the same reason as above the results of so many operations with dependent intervals will yield unrealistic error bounds.

3. Solving systems of linear equations by the Newton method

We will now restrict ourselves (without loss of generality) to problem (A) of section 2 with a real-valued matrix A and vector b, i.e. to solving the system

$$f(x) := Ax-b = 0. \tag{1}$$

Because $\partial f/\partial x = A$ the application of the Newton method to (1) gives the algorithm

$$x_0 \text{ arbitrary},$$
$$x_{k+1} := x_k - A^{-1}(Ax_k-b) \quad (k = 0, 1,\ldots).$$

As in Chapter 2 we transform this real algorithm to the triplex algorithm

$$X_0 = [\underline{x}_0, \tilde{x}_0, \bar{x}_0] \text{ arbitrary},$$
$$\lfloor X_{k+1} \rfloor = \lfloor [\underline{x}_{k+1}, \tilde{x}_{k+1}, \bar{x}_{k+1}] \rfloor := \tilde{x}_k - \lfloor C \rfloor (A\tilde{x}_k - b) \quad (k = 0, 1,\ldots),$$
$$\lfloor C \rfloor \supseteq A^{-1}.$$

† A multiplicative constant factor being suppressed.

Here the arithmetic has to be performed as triplex-arithmetic. The dot under a variable (e.g. $\dot{A}$) means this variable (matrix, vector,...) is real-valued. The notation $\lfloor\ \rfloor$ indicates that a variable is triplex-valued (e.g. $\lfloor C \rfloor$).

The matrix $\lfloor C \rfloor$ may have very unrealistic error bounds; in general, one will use the 'inverse' computed with the triplex Gauss algorithm. Theoretically the main values $\tilde{c}_{ik}$ of the components $[\underline{c}_{ik}, \tilde{c}_{ik}, \bar{c}_{ik}]$ of $\lfloor C \rfloor$ are the exact values of the components of A^{-1}. Therefore, for an arbitrary starting vector $X_0 = [\underline{x}_0, \tilde{x}_0, \bar{x}_0]$, the main value $\tilde{x}_1$ of

$$X_1 = [\underline{x}_1, \tilde{x}_1, \bar{x}_1]$$

has the correct value, i.e. $A\tilde{x}_1 = b$. Thus, for exact real arithmetic, $X_2 = X_3 = \ldots = [\tilde{x}_1, \tilde{x}_1, \tilde{x}_1]$.

Practically there will be a small difference between $\tilde{x}_1$ and the correct solution x, due to the round-off errors. But using (for the component $\tilde{c}_{ik}$) the values obtained with ordinary floating-point arithmetic is obviously the best that can be done with a given computer. Therefore the practical convergence will always be very fast. In some thousand examples, even for condition numbers higher than 10^{10}(!), never more than 5 iteration steps had to be performed.

Hansen ([4], [5]) uses exactly the above-given algorithm with the only difference that he—having in interval arithmetic no 'main' value—replaces the $\tilde{c}_{ik}$ by $(\underline{c}_{ik}+\bar{c}_{ik})/2$, which are in general worse values. This is a second example illustrating the fact that numerical algorithms very often quite naturally need not only the error bounds but also the 'main' value, which is available in triplex numbers and triplex arithmetic.

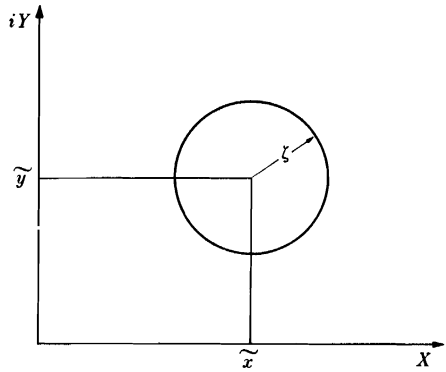
After the Oxford Symposium the author found the excellent paper of Hansen and Smith [5], which he did not know of before. Independently of each other Krückeberg [6] and the author have also considered the methods used therein. The results of the evaluations of A^{-1} and $A^{-1}b$ in some thousand experiments,† with well- and ill-conditioned matrices, were always very good. In these experiments, double-precision arithmetic entailed 16 decimal digits. A typical example for an extremely ill-conditioned matrix A had $n = 10$, $\|A\| = \|b\| = 10$, $x = (1, 1, \ldots, 1)^T$, $\|A^{-1}\| = 1{\cdot}96 \times 10^{11}$, and conditions number $\|A\| \cdot \|A^{-1}\| = 1{\cdot}96 \times 10^{12}$(!). Both A^{-1} and $A^{-1}b$ were iteratively computed. The number of iterations

† Computed with the IBM 360/67 computer in October 1967 while the author was at the IBM Thos. J. Watson Research Laboratory at Yorktown Heights, New York, U.S.A.

needed was 5; the max span (A^{-1})† to begin with was $7{\cdot}40\times10^{14}$, and max span (A^{-1}) after 5 iterations was $2{\cdot}07\times10^{6}$. Also, max span (X_0)† $= 3{\cdot}06\times10^{16}$, max span $(X_4) = 1{\cdot}47\times10^{-3}$. Therefore the value of $\theta := 10^{16}\times$ max span $(X_4)/\|A\|\cdot\|A^{-1}\| = 7{\cdot}46$. This value θ shows, as is well known (see Wilkinson [10]), how realistically the bounds could be computed using 16 decimal digits. In the experiments, θ was always(!) between 2 and 9. Also the maximum of the span of the bounds of the 'inverse' A^{-1} was 2×10^{6}, which is nearly 'best' for matrices A and A^{-1} with norms 10^{1} and 2×10^{11} if 16-decimal-digit arithmetic is involved.

4. Complex-valued triplex arithmetic

If complex numbers $z := x+iy$, $i^2 = -1$ are considered there are different possibilities for changing to triplex variables. The most complicated way is to store a complex triplex number $Z = X+iY$ as a pair of two triplex numbers $X = [\underline{x}, \tilde{x}, \bar{x}]$ and $Y = [\underline{y}, \tilde{y}, \bar{y}]$. A much more natural way is the following (see figure). A complex triplex number Z describes a circle $|Z-\tilde{z}| \leqslant \zeta$ and is stored as a triple $Z = [\tilde{x}, \tilde{y}, \zeta]$. Here $\tilde{x}$ and $\tilde{y}$ will generally be two floating-point numbers of full length (single or double precision), while ζ may be a number having fewer digits.



Until now there has not been defined such a generalization of one of the common algorithmic languages. There is no *complex* concept in Algol (except in Russia). It should, however, not be too difficult to introduce triplex-complex in Fortran IV and PL/I, where complex variables are known.

† For a vector B with the triplex elements $B_i = [\underline{b}_i, \tilde{b}_i, \bar{b}_i]$ and for a matrix C with $C_{ik} = [\underline{c}_{ik}, \tilde{c}_{ik}, \bar{c}_{ik}]$, by definition, max span(B): $= \max_{i=1(1)n}$ span$(B_i) = \max_{i=1(1)n} (\bar{b}_i - \underline{b}_i)$ and max span(C): $= \max_{i,k=1(1)n}$ span$(C_{ik}) = \max_{i,k=1(1)n} (\bar{c}_{ik} - \underline{c}_{ik})$.

5. Roots of polynomials

In a previous paper [8] an algorithm was given for the computation of all roots W_p of a given polynomial

$$P(Z) := \sum_{\nu=0}^{n} a_\nu Z^\nu = a_n \prod_{p=1}^{n} (Z-W_p)$$

for a_ν real, $a_n \neq 0$, $Z = X+iY$. This algorithm includes error bounds obtained by the aid of a triplex-like arithmetic. The algorithm was also published as an Algol program without the computation of the error bounds [9].

The evaluation of one root W_p is performed in four steps.

(A) Find an approximation Z^*.
(B) Improve Z^*.
(C) Go to step (D) if Z^* is good enough; otherwise go back to (B).
(D) Compute an error bound $\zeta \geqslant 0$ such that

$$|Z^*-W_p| \leqslant \zeta.$$

After finishing these four steps put $W_p := (Z^*, \zeta)$, where Z^* is an approximation and ζ an error bound. For the evaluation of W_{p+1} the polynomial $P(Z)$ is replaced by $P(Z)/(Z-W_p)$. This is done for $p = 1(1)n$.

It is very important to note that *no* triplex analysis is performed during the steps (A) and (B). In step (C) a triplex-based decision can be used if desired. In this case the phrase 'Z^* is good enough' is interpreted as 'the circle for $P(Z^*)$, computed with complex triplex arithmetic, contains the zero'. But it should be emphasized that step (C) can also be performed in floating-point complex arithmetic without any knowledge of the round-off errors (see [9]).

For the computation of ζ in step (D) the following formula due to Fekete will be used:

$$\zeta := \min_{k=1(1)n} \sqrt[k]{\left\{\binom{n}{k} \Big/ \left|\frac{C_0}{C_k}\right|\right\}}.$$

Here the coefficients C_ν are defined by the identity

$$P(Z) = \sum_{\nu=0}^{n} C_\nu (Z-Z^*)^\nu,$$

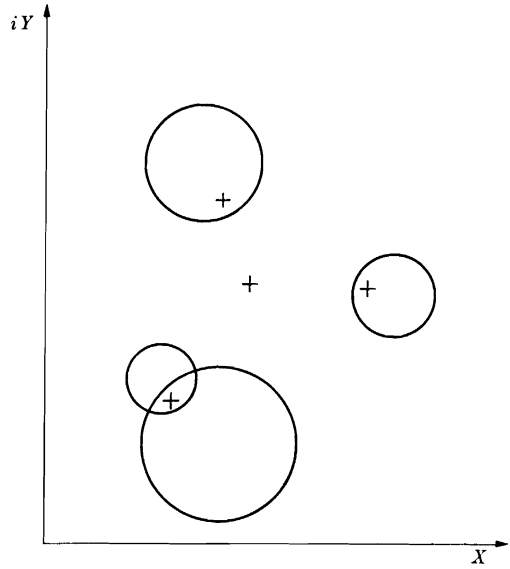
but they are computed using triplex arithmetic.

It is rather trivial to note that triplex arithmetic is necessary for this step. This can be seen from just the fact that in floating-point arithmetic it may be that $P(Z^*) = C_0$ is numerically zero by underflow, even in the case $P(Z^*) \neq 0$. This would give, however, an error bound $\zeta = 0$, which is obviously wrong.

There is, however, a still unsolved problem. In one approach to the problem the reduction $P(Z)/(Z-W_p)$ is performed in triplex arithmetic where $W_p = (Z^*, \zeta)$ as above. In this case the computation of the error bounds for the roots of the reduced polynomial gives exact values; that is, strict bounds. But, unfortunately, a great loss of accuracy of the bounds occurs, due to the occurrence of dependent intervals during the reduction. As can be seen in the results of [8], for just $n \geqslant 6$ the typical bounds have a relative error of more than 100 per cent

$$(\text{i.e. span}\,(|Z^*-W_p|) > |W_p|).$$

Alternatively, if the reduction is not performed in triplex arithmetic but in floating-point arithmetic, the bounds ζ for the approximation Z must be computed by using the (undisturbed) values of the initial coefficients a_ν for $\nu = 0(1)n$. In this case (see [8], Fig. 2) the following phenomenon may occur.



It may be that some of the circles obtained by step (D) are separated and contain therefore at least (in general, exactly) one root, while others overlap. For the latter case it can be concluded only that there is at least one root inside them. Therefore some of the roots may get 'lost'. See the figure for the case $n = 4$, where the exact roots are denoted by crosses. There are two roots 'trapped' in single circles, one root in the intersection of two circles, and one root is not contained in any of the circles.

If by *a posteriori* inspection it is found that all circles are separated, then all roots W_p have been 'trapped'. But if there are multiple roots or clusters of roots then the geometric situation is not so favourable. In this case, the reduction can be performed by means of triplex arithmetic. Then, however, favourable error bounds cannot be guaranteed for all roots.

It is possible, however, to obtain error bounds ζ for a given approximation Z^* such that there exist at least $k \leqslant n$ roots $W_1, W_2, \ldots, W_k$ for which $|W_p - Z^*| \leqslant \zeta$ for $p = 1(1)k$. With such bounds the above problem can obviously be completely solved. An example for $k = 2$ is the following formula:

$$\zeta := \frac{n-1}{2} |C_1/C_2| + \left[\frac{(n-1)^2}{4} |C_1/C_2|^2 + \frac{n(n-1)}{2} |C_0/C_2| \right]^{\frac{1}{2}}.$$

As a numerical example for $n = 10$, consider

$$P(Z) := 0{\cdot}1 + 0{\cdot}1Z + 10Z^2 + \ldots, \quad \text{with } Z^* := 0.$$

Fekete's formula reveals that there exists one root W_1 of $P(Z)$ such that

$$|W_1| \leqslant \min\left(n|C_0/C_1|, \left[\frac{n(n-1)}{2} |C_0/C_2| \right]^{\frac{1}{2}}, \ldots \right)$$
$$= \min\left(10, \sqrt{(0{\cdot}45)}, \ldots\right) < 0{\cdot}671.$$

Obviously there is one other root in the neighbourhood of the origin. The new formula gives $|W_1|, |W_2| < 0{\cdot}718$

with a bound ζ nearly as good as that from the Fekete formula but now with the certainty of bounding two zeros instead of one.

REFERENCES

1. Apostolatos, N., and Kulisch, U. Grundlagen einer Maschinenintervallarithmetik. *Computing*, **2**, 89–104 (1967).
2. —— —— Approximation der erweiterten Intervallarithmetik durch die einfache Maschinenintervallarithmetik. Ibid. **2**, 181–94 (1967).
3. —— —— Grundzüge einer Intervallrechnung für Matrizen und einige Anwendungen. *Elektron. Rechenanl.* **10**, 73–83 (1968).
4. Hansen, E. Interval arithmetic in matrix computations, Part I. *SIAM Jl numer. Anal.* **2**, 308–20 (1965).
5. —— and Smith, R. Interval arithmetic in matrix computations, Part II. Ibid. **4**, 1–9 (1967).
6. Krückeberg, F. Inversion von Matrizen mit Fehlererfassung. *Z. angew. Math. Mech.* **46**, T69–T71 (1966).

7. MOORE, R. E. *Interval analysis.* Prentice Hall, New Jersey (1966).

8. NICKEL, K. Die numerische Berechnung der Wurzeln eines Polynoms. *Num. Math.* **9,** 80–98 (1966).

9. —— Algorithmus 5. Die Nullstellen eines Polynoms. *Computing,* **2,** 284–8 (1967).

10. WILKINSON, J. H. *Rounding errors in algebraic processes.* Prentice Hall, New Jersey (1963).

4 · On Linear Algebraic Equations with Interval Coefficients

1. Introduction

CONSIDER the set of equations

$$Ax = b \tag{1.1}$$

where $b = (b_i)$ is a given real vector of order n and $A = (a_{ij})$ is a given, non-singular, real matrix of order n. The solution vector x is the unique vector $A^{-1}b$. Suppose, however, that A and b are subject to error. Suppose we know only that $a_{ij} \in a_{ij}^I = [a_{ij}^L, a_{ij}^R]$ and $b_i \in b_i^I = [b_i^L, b_i^R]$, for $i, j = 1,..., n$. Denote $A^I = (a_{ij}^I)$, and $b^I = (b_i^I)$ and assume no $A \in A^I$ is singular. We wish to know the set of solutions

$$X = \{x: Ax = b, A \in A^I, b \in b^I\} \tag{1.2}$$

to the equation

$$A^I x = b^I. \tag{1.3}$$

Hansen and Smith [3] discussed methods for computing an interval vector x^I containing X. A vector $x^I = [x^L, x^R]$ defines a region in an n-dimensional space bounded by the planes $x_i = x_i^L$ and $x_i = x_i^R$ $(i = 1,..., n)$. In general, the set X is not a region bounded by planes parallel to the coordinate axes and hence the narrowest interval vector x^I is not equal to X. (By the narrowest interval vector, we mean the vector whose elements are all narrowest possible.)

Nevertheless, the smallest x^I is of interest. In this chapter we show how to obtain both upper and lower bounds on the narrowest $x^I \supset X$.

We assume that at least one element of A^I or b^I is an interval of non-zero width. Otherwise, the method described below is no different from that in [3].

2. The solution set X

In this section we briefly consider the set X. Rewrite (1.3) as

$$\sum_{j=1}^{n} a_{ij}^I x_j = b_i^I \quad (i = 1,..., n). \tag{2.1}$$

Suppose some fixed point $x \in X$ lies in the positive orthant. For this x, (2.1) can be written (symbolically only) as

$$\sum_{j=1}^{n} [a_{ij}^L x_j, a_{ij}^R x_j] = b_i^I,$$

or

$$\Big[\sum_{j=1}^{n} a_{ij}^L x_j, \sum_{j=1}^{n} a_{ij}^R x_j\Big] = b_i^I \quad (i = 1,..., n). \tag{2.2}$$

There exists $A \in A^I$ and $b \in b^I$ such that $Ax = b$, for this fixed x, provided the intervals on the left and right of (2.2) intersect for each $i = 1,..., n$. That is, $x \in X$ (if $x_j \geqslant 0$ for $j = 1,..., n$) provided

$$\sum_{j=1}^{n} a_{ij}^L x_j \leqslant b_i^R, \qquad \sum_{j=1}^{n} a_{ij}^R x_j \geqslant b_i^L \quad (i = 1,..., n). \tag{2.3}$$

If $x_j \leqslant 0$ for some value of j, then $a_{ij}^I x_j = [a_{ij}^R x, a_{ij}^L x_j]$. Hence by noting the sign of each x_j for some given x, we can write an 'equation' corresponding to (2.2) and obtain conditions similar to (2.3). Thus it is a simple matter to check whether a given x lies in X.

However, in general it is not so simple to find X or to represent it once it is found. It is for this reason we choose to study x^I instead. The set X is discussed by Oettli, Prager, and Wilkinson [6]. See also Rigal and Gaches [7]. For our purpose we shall find X in a simple illustrative case by solving inequalities of the form (2.3).

Consider the equations:

$$\begin{aligned} [2, 3]x_1 + [0, 1]x_2 &= [0, 120], \\ [1, 2]x_1 + [2, 3]x_2 &= [60, 240]. \end{aligned} \tag{2.4}$$

In this first quadrant, where

$$x_1 \geqslant 0, \qquad x_2 \geqslant 0, \tag{2.5}$$

we can rewrite the left members of (2.4) as $[2x_1, 3x_1+x_2]$ and $[x_1+2x_2, 2x_1+3x_2]$, respectively. The intersections

$$[2x_1, 3x_1+x_2] \cap [0, 120]$$

and

$$[x_1+2x_2, 2x_1+3x_2] \cap [60, 240]$$

must not be empty and hence

$$\begin{aligned} 2x_1 &\leqslant 120, & 3x_1+x_2 &\geqslant 0, \\ x_1+2x_2 &\leqslant 240, & 2x_1+3x_2 &\geqslant 60. \end{aligned} \tag{2.6}$$

From (2.5) and (2.6) we easily find that (in the first quadrant) X is a polygon with vertices at the points $(30, 0)$, $(60, 0)$, $(60, 90)$, $(0, 120)$, and $(0, 20)$.

Repeating this process for the other three quadrants, we find that X is a polygon in the second and fourth quadrants but that no point of X lies in the third quadrant. Combining these results, we find that X is an eight-sided polygon. Proceeding counter-clockwise around the

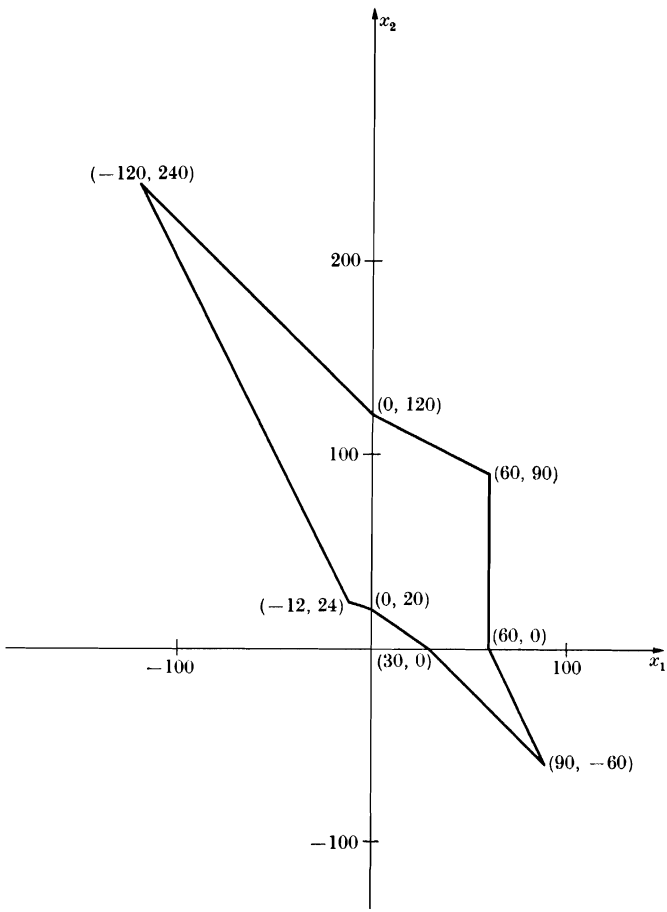


FIG. 4.1. The solution set for equations (2.4).

boundary of X, its vertices are at the points $(30, 0)$, $(90, -60)$, $(60, 0)$, $(60, 90)$, $(0, 120)$, $(-120, 240)$, $(-12, 24)$, and $(0, 20)$. See Fig. 4.1.

Even in this two-dimensional case, the set X is not particularly easy to represent. Obviously, in higher-dimensional cases, X is difficult to represent in general. (In special cases, of course, this need not be so.) We can, however, easily represent the smallest parallelepiped containing X having sides parallel to the coordinate axes. In our example, this

parallelepiped is bounded by the planes $x_1 = -120, x_1 = 90, x_2 = -60$, and $x_2 = 240$. We may thus represent it by

$$x^I = \begin{bmatrix} [-120, 90] \\ [-60, 240] \end{bmatrix}.$$

It is our purpose in this chapter to show how to obtain both upper and lower bounds on x^I for arbitrary A^I and b^I.

3. The basic method

We shall use a method that is essentially an interval analytic version of a procedure of Kuperman's [4] for bounding errors in the computed solution to a set of linear equations. Kuperman's method is applicable for only certain problems (described later). We shall show how to extend its applicability to all cases.

Consider the equation $Ax = b$ and suppose $\partial x_r/\partial a_{ij} \geqslant 0$, where x_r is the rth element of x. Then x_r is non-decreasing as a_{ij} increases. If a_{ij} can take any value in $a_{ij}^I = [a_{ij}^I, a_{ij}^R]$, then x_r will be smallest if $a_{ij} = a_{ij}^L$ and largest if $a_{ij} = a_{ij}^R$. If $\partial x_r/\partial a_{ij}$ is of one sign for each $i, j = 1,..., n$ for all $a_{ij} \in a_{ij}^I$ and all $x_r \in x_r^I$, then the largest (and smallest) value of x_r occurs in the solution of a problem in which each a_{ij} $(i, j = 1,..., n)$ has the appropriate extreme value a_{ij}^L or a_{ij}^R.

We now develop these ideas into an algorithm for computing bounds on x^I. In so doing we include the possibility that $\partial x_r/\partial a_{ij}$ may change sign. We also consider the effect of the vector b^I.

Consider the equation $A^I x = b^I$ and define the matrices P_r^I and Q_r^I and the vectors c_r^I and d_r^I whose elements are, respectively,

$$P_{rij}^I = \begin{cases} a_{ij}^L \text{ if } \partial x_r/\partial a_{ij} \geqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ a_{ij}^R \text{ if } \partial x_r/\partial a_{ij} \leqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ a_{ij}^I \text{ otherwise;} \end{cases} \tag{3.1}$$

$$Q_{rij}^I = \begin{cases} a_{ij}^R \text{ if } \partial x_r/\partial a_{ij} \geqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ a_{ij}^L \text{ if } \partial x_r/\partial a_{ij} \leqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ a_{ij}^I \text{ otherwise;} \end{cases} \tag{3.2}$$

$$c_{ri}^I = \begin{cases} b_i^L \text{ if } \partial x_r/\partial b_i \geqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ b_i^R \text{ if } \partial x_r/\partial b_i \leqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ b_i^I \text{ otherwise;} \end{cases} \tag{3.3}$$

$$d_{ri}^I = \begin{cases} b_i^R \text{ if } \partial x_r/\partial b_i \geqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ b_i^L \text{ if } \partial x_r/\partial b_i \leqslant 0 \text{ for all } A \in A^I \text{ and all } b \in b^I, \\ b_i^I \text{ otherwise.} \end{cases} \tag{3.4}$$

Let $$P_r^I y_r = c_r^I \tag{3.5}$$

and $$Q_r^I z_r = d_r^I. \tag{3.6}$$

The rth component y_{rr}^I of the solution y_r^I of (3.5) contains the left endpoint x_r^L of the rth component x_r^I of x^I. Similarly, z_{rr}^I contains the right endpoint x_r^R of x_r^I.

If we solve (3.5) and (3.6) for $r = 1,..., n$ using interval arithmetic, we obtain both upper and lower bounds on each of the components x_r^L and x_r^R. Such a solution might be obtained by the method recommended in [3] (a Fortran program called LSD using this method is given in [8] and is available from the SHARE organization). Note that only the rth component of y_r^I is required. Hence for most values of r, the solution of (3.5) need not be fully completed. If P_r^I and c_r^I are the same for two (or more) values of r, then (3.5) and (3.6) need be solved only once for both values of r. In the most fortuitous case P_r^I is the same for all $r = 1,..., n$. This occurs, for example, if every $A \in A^I$ is an M matrix and $b_i^I \geqslant 0$ $(i = 1,..., n)$. (For a definition of an M matrix see, for example, Varga [9].)

Having solved (3.5) for y_{rr}^I, we solve (3.6) for z_{rr}^I. The remarks in the preceding paragraph again apply.

4. Obtaining the derivatives

We now consider how to obtain the derivatives which determine how P_r, Q_r, c_r, and d_r will be formed. Differentiating (1.1) with respect to a_{ij}, we have

$$A \frac{\partial x}{\partial a_{ij}} + E_{ij} x = 0, \tag{4.1}$$

where E_{ij} denotes the matrix whose every element is 0 except that the element in position (i,j) is 1. Let $W = A^{-1}$, then

$$\frac{\partial x}{\partial a_{ij}} = -W E_{ij} x$$

and hence $$\frac{\partial x_r}{\partial a_{ij}} = -w_{ri} x_j. \tag{4.2}$$

Differentiating (1.1) with respect to b_i, we have

$$A \frac{\partial x}{\partial b_i} = e_i, \tag{4.3}$$

where e_i denotes the ith column of the identity matrix. Therefore

$$\frac{\partial x}{\partial b_i} = We_i$$

and

$$\frac{\partial x_r}{\partial b_i} = w_{ri}. \tag{4.4}$$

In (3.1)–(3.4) we need to know whether the derivatives $\partial x_r/\partial a_{ij}$ and/or $\partial x_r/\partial b_i$ are of one sign for all $A \in A^I$ and all $b \in b^I$. We can use (4.2) and (4.4) to attempt to determine this. Using (say) the LSD program we can obtain an interval vector $\bar{x}^I$ containing x^I and an interval matrix $\overline{W}^I$ containing $(A^I)^{-1}$.

If, for example, $\bar{w}^I_{ri}\bar{x}^I_j \geqslant 0$ for some values of r, i, and j, then, from (4.2) $\partial x_r/\partial a_{ij} \leqslant 0$ for all $A \in A^I$ and all $b \in b^I$. Similarly, if $\bar{w}_{ri} \geqslant 0$, say, then from (4.4), $\partial x_r/\partial b_i \geqslant 0$ for all $A \in A^I$ and all $b \in b^I$. That is, examining $\bar{w}^I$ and $\bar{x}^I$, we can determine P^I_r, Q^I_r, c^I_r, and d^I_r for $r = 1,\ldots,n$. Actually, since $\bar{w}^I$ and $\bar{x}^I$ will not be sharp, in general, we may obtain a matrix $\overline{P}^I_r$ containing but not equal to P^I_r. That is, it may be that $w^I_{ri}x^I_j \geqslant 0$ while 0 is an interior point of $\bar{w}^I_{ri}\bar{x}^I_j$. Similarly we may not obtain Q^I_r, c^I_r, and/or d^I_r sharply. This would not invalidate our results but would make them less sharp.

5. An example

We shall now consider an example of a type for which many elements of w^I and x^I change sign as A ranges over A^I and b ranges over b^I. Thus many elements of P^I_r, Q^I_r, c^I_r, and d^I_r will be intervals. As we shall see, however, the results are still rather good with respect to sharpness.

Let $A^I = A + \epsilon^I E$, where $\epsilon^I = a[-1, 1]$, $a = 10^{-4}$, E is the matrix whose every element is 1, and

$$A = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Let $b^I = (1+\epsilon^I)e$, where $e = (1, 1,\ldots, 1)^T$.

The equation $A^I x = b^I$ was solved and the 'inverse' $\overline{W}^I \supset (A^I)^{-1}$ was computed by the LSD program using floating-point arithmetic with a 27-binary-bit mantissa. It was found that

$$\begin{gathered} \bar{w}^I_{ii} > 0 \quad (i = 1,\ldots, 5), \\ \bar{w}^I_{i,i+1} < 0, \qquad \bar{w}^I_{i+1,i} < 0 \quad (i = 1,\ldots, 4), \\ 0 \in \bar{w}^I_{ij}, \quad \text{otherwise.} \end{gathered} \tag{5.1}$$

It can be shown that relations (5.1) hold for $\overline{W}^I$ replaced by $W^I = (A^I)^{-1}$. Using the same program, a vector $\bar{x}^I$ was computed which contains the solution x^I to $A^I x = b^I$. It was found that

$$\begin{gathered} 0 \in \bar{x}_i^I \quad (i = 1,..., 4), \\ \bar{x}_5^I > 0. \end{gathered} \tag{5.2}$$

It can be shown that (5.2) holds with $\bar{x}^I$ replaced by x^I.

From (4.2), (5.1), and (5.2), we see that $\partial x_r/\partial a_{ij}$ changes sign as A ranges over A^I and b ranges over b^I unless $j = 5$ and $r = i-1, i$, or $i+1$. Hence $P_r^I = Q_r^I = A^I$ except for either two or three elements in the last column. Also, from (4.4) and (5.1), $c_r^I = d_r^I = b^I$ unless $r = i-1$, i, or $i+1$. The vectors c_r^I are

$$\begin{bmatrix} 1-a \\ 1+a \\ 1+\epsilon^I \\ 1+\epsilon^I \\ 1+\epsilon^I \end{bmatrix}, \quad \begin{bmatrix} 1+a \\ 1-a \\ 1+a \\ 1+\epsilon^I \\ 1+\epsilon^I \end{bmatrix}, \quad \begin{bmatrix} 1+\epsilon^I \\ 1+a \\ 1-a \\ 1+a \\ 1+\epsilon^I \end{bmatrix}, \quad \begin{bmatrix} 1+\epsilon^I \\ 1+\epsilon^I \\ 1+a \\ 1-a \\ 1+a \end{bmatrix}, \quad \begin{bmatrix} 1+\epsilon^I \\ 1+\epsilon^I \\ 1+\epsilon^I \\ 1+a \\ 1-a \end{bmatrix}$$

for $r = 1,..., 5$, respectively. The vectors d_r^I ($r = 1,..., 5$) are obtained from c_r^I by replacing a by $-a$. The matrices P_r^I ($i = 1,..., 5$) are obtained by replacing the last column of A^I by d_r^I. The matrices Q_r^I ($i = 1,..., 5$) are obtained by replacing the last column of A^I by c_r^I.

We solve (3.5) for y_{rr}^I and (3.6) for z_{rr}^I using the LSD program. Let y^I denote the vector with elements y_{rr}^I ($r = 1,..., 5$) and z^I denote the vector with elements z_{rr}^I ($r = 1,..., 5$). We obtain

$$y^I = \begin{bmatrix} [-4{\cdot}006\ 812, & -3{\cdot}993\ 190]\times 10^{-4} \\ [-2{\cdot}013\ 624, & -7{\cdot}986\ 387]\times 10^{-4} \\ [-8{\cdot}012\ 018, & -7{\cdot}986\ 392]\times 10^{-4} \\ [-8{\cdot}013\ 624, & -7{\cdot}991\ 186]\times 10^{-4} \\ [0{\cdot}9993\ 993, & 0{\cdot}9994\ 011] \end{bmatrix},$$

$$z^I = \begin{bmatrix} [3{\cdot}993\ 190, & 4{\cdot}006\ 812]\times 10^{-4} \\ [7{\cdot}986\ 388, & 8{\cdot}013\ 623]\times 10^{-4} \\ [7{\cdot}987\ 986, & 8{\cdot}013\ 624]\times 10^{-4} \\ [7{\cdot}986\ 400, & 8{\cdot}008\ 811]\times 10^{-4} \\ [1{\cdot}000\ 599, & 1{\cdot}000\ 602] \end{bmatrix},$$

where we have recorded the results to only seven significant decimal digits. Denote $y^I = [y^L, y^R]$ and $z^I = [z^L, z^R]$. Then

$$[y^R, z^L] \subset x^I \subset [y^L, z^R].$$

If we approximate x^I by

$$[(y^L+y^R)/2, (z^L+z^R)/2] = \begin{bmatrix} [-4{\cdot}000\,001, & 4{\cdot}000\,001]\times 10^{-4} \\ [-8{\cdot}000\,005, & 8{\cdot}000\,005]\times 10^{-4} \\ [-7{\cdot}999\,205, & 8{\cdot}000\,805]\times 10^{-4} \\ [-8{\cdot}002\,405, & 7{\cdot}997\,605]\times 10^{-4} \\ [0{\cdot}9994\,002, & 1{\cdot}000\,600] \end{bmatrix},$$

we know that no endpoint of any element can be in error by as much as 2×10^{-6}.

The widths of the elements of y^I and z^I are quite small even though many elements P_r^I, Q_r^I, c_r^I, and d_r^I were intervals rather than real numbers. This can be explained as follows. The interval elements occurred because 0 was contained either in the set

$$S = \left\{\frac{\partial x_r}{\partial a_{ij}} : Ax = b,\ A \in A^I,\ b \in b^I\right\}$$

or

$$T = \left\{\frac{\partial x_r}{\partial b_i} : Ax = b,\ A \in A^I,\ b \in b^I\right\}.$$

But if $0 \in S$, then, in general, $\partial x_r/\partial a_{ij}$ is small for all $A \in A^I$ and $b \in b^I$. Therefore the value of x_r depends very little on the value of a_{ij}. Hence $w(x_r^I)$ is increased only slightly by using a_{ij}^I rather than some real number $a_{ij} \in a_{ij}^I$. Here $w(x_r^I)$ denotes the width of the interval x_r^I. A similar argument holds if $0 \in T$.

6. A special case

Suppose that no element of W^I or x^I contains zero. Then P_r^I, Q_r^I, c_r^I, and d_r^I are real. It was this special case for which Kuperman [4] observed that the analysis of the previous sections was valid. (However, he did not express his results in interval analytic form as we have done.) It is also in this special case that the linear form of Oettli's method [5] is valid. He obtains X as the solution of a linear programming problem.

Since in this case (3.5) and (3.6) are real, not interval, equations, we can solve them to arbitrary accuracy by using (interval, say) arithmetic of sufficiently high precision. This is not the case for the original equation $A^I x = b^I$ nor is it the case for (3.5) and (3.6) if they are interval, not real, equations. In the latter cases, the inherent lack of sharpness (see [2] or [3]) precludes obtaining arbitrarily high accuracy in the computed result without a prohibitively large amount of work.

7. A more special case

We have considered the case in which no element of x or W changes sign as A ranges over A^I and b ranges over b^I. If in addition, the width of a_{ij}^I is the same for each row of A^I or for each column of A^I, the vector x^I can be found more easily.

Denote
$$A^I = A^c + [-1, 1]M, \tag{7.1}$$
where A^c is the centre of A^I and M is the real matrix with $m_{ij} = \frac{1}{2}w(a_{ij}^I)$. If m_{ij} is independent of i, then $M = ev^T$ for some vector v, where v^T denotes the transpose of v. If m_{ij} is independent of j, then $M = ue^T$ for some u. In the example in section 5, we chose $M = 10^{-4}ee^T$.

Examples of this kind could be obtained in practice. For instance, we might replace m_{ij} by $\max_{1 \leqslant i \leqslant n} m_{ij}$ for each $j = 1,..., n$. We would thus sacrifice sharpness for ease of solution.

As a more general case, assume
$$M = uv^T. \tag{7.2}$$
Then
$$a_{ij}^L = a_{ij}^c - u_i v_j, \qquad a_{ij}^R = a_{ij}^c + u_i v_j. \tag{7.3}$$
Since by assumption no element of x or W changes sign as A ranges over A^I and b ranges over b^I, then, from (3.1) and (4.2),
$$P_{rij} = \begin{cases} a_{ij}^L & \text{if } w_{ri}^I x_j^I \leqslant 0, \\ a_{ij}^R & \text{if } w_{ri}^I x_j^I \geqslant 0. \end{cases} \tag{7.4}$$
Define
$$s_j = \begin{cases} v_j & \text{if } x_j^R > 0, \\ -v_j & \text{if } x_j^L < 0, \\ 0 & \text{if } x_j^L = x_j^R = 0, \end{cases}$$
$$k_i^{(r)} = \begin{cases} 1 & \text{if } w_{ri}^R > 0, \\ -1 & \text{if } w_{ri}^L < 0, \\ 0 & \text{if } w_{ri}^L = w_{ri}^R = 0, \end{cases}$$
and $t_i^{(r)} = k_i^{(r)} u_i$. Then
$$P_{rij} = a_{ij}^c + t_i^{(r)} s_j. \tag{7.5}$$
Denote
$$b^I = b^c + [-1, 1]f, \tag{7.6}$$
where b^c is the centre of b^I and f is the real vector with elements
$$f_i = \tfrac{1}{2}w(b_i^I).$$
From (3.3) and (4.4)
$$c_{ri} = \begin{cases} b_i^L & \text{if } w_{ri}^I \geqslant 0, \\ b_i^R & \text{if } w_{ri}^I \leqslant 0 \end{cases}$$
$$= b_i^c - k_i^{(r)} f_i. \tag{7.7}$$

Let $g^{(r)}$ denote the vector with elements

$$g_i^{(r)} = k_i^{(r)} f_i.$$

Then $$c_{ri} = b_i^c - g_i^{(r)}$$

and (3.5) becomes $$(A^c + t^{(r)} s^T) y_r = b^c - g^{(r)}. \tag{7.8}$$

Now $$(A^c + t^{(r)} s^T)^{-1} = W^c (I - \alpha_r t^{(r)} s^T W^c),$$

where $W^c = (A^c)^{-1}$ and

$$\alpha_r = \frac{1}{1 + s^T W^c t^{(r)}}.$$

Hence $$y_r = W^c (I - \alpha_r t^{(r)} s^T W^c)(b^c - g^{(r)}).$$

We require only the rth element of y_r. It is given by

$$y_{rr} = e_r^T y_r = e_r^T W^c (I - \alpha_r t^{(r)} s^T W^c)(b^c - g^{(r)}). \tag{7.9}$$

Similarly, $$z_{rr} = e_r^T W^c (I + \beta_r t^{(r)} s^T W^c)(b^c + g^{(r)}), \tag{7.10}$$

where $$\beta_r = \frac{1}{1 - s^T W^c t^{(r)}}.$$

The elements of x^I are given by

$$x_r^I = [y_{rr}, z_{rr}].$$

In practice we can obtain bounds on y_{rr} and z_{rr} using the following procedure. First invert A^I, obtaining $\overline{W}^I \supset (A^I)^{-1}$, and solve $A^I x = b^I$, obtaining $\bar{x}^I$. If some element of $\overline{W}^I$ or $\bar{x}^I$ contains 0 as an interior point, we cannot proceed with our simplified method. If no element of $\overline{W}^I$ or $\bar{x}^I$ contains 0 as an interior point, we form the vectors $k^{(r)}$, s, $t^{(r)}$, and $g^{(r)}$. Next invert A^c, obtaining an interval matrix $\overline{W}^{cI}$ containing W^c. We can now compute intervals y_{rr}^I and z_{rr}^I obtained by replacing W^c by $\overline{W}^{cI}$ in (7.9) and (7.10).

8. Example

To illustrate the analysis of the last section, we consider an example of order 4 studied by Albrecht [1] and Oettli [5]. Let

$$A^I = A^c + \epsilon^I E, \qquad b^I = b^c + \epsilon^I e,$$

where $\epsilon^I = 5 \times 10^{-3}[-1, 1]$, E is the matrix whose every element is 1, and

$$A^c = \begin{bmatrix} 4{\cdot}33 & -1{\cdot}12 & -1{\cdot}08 & 1{\cdot}14 \\ -1{\cdot}12 & 4{\cdot}33 & 0{\cdot}24 & -1{\cdot}22 \\ -1{\cdot}08 & 0{\cdot}24 & 7{\cdot}21 & -3{\cdot}22 \\ 1{\cdot}14 & -1{\cdot}22 & -3{\cdot}22 & 5{\cdot}43 \end{bmatrix}, \quad b^c = \begin{bmatrix} 3{\cdot}52 \\ 1{\cdot}57 \\ 0{\cdot}54 \\ -1{\cdot}09 \end{bmatrix}.$$

We wish to solve $A^I x = b^I$.

Following the prescription in section 7, we invert A^I, using the LSD program obtaining $\overline{W}^I \supset (A^I)^{-1}$. We find that $\bar{w}^I_{ij} > 0$ for all $i, j = 1, 2, 3, 4$ except that $\bar{w}^I_{14} < 0$ and $\bar{w}^I_{41} < 0$. Similarly, we find $\bar{x}^I_i > 0$ for $i = 1, 2$, and 3 and $\bar{x}^I_4 < 0$. We thus know that no element of $(A^I)^{-1}$ or X contains 0 and the method of section 7 applies.

We require W^c. Using the same program to invert A^c, we find

$$\overline{W}^{cI} = \begin{bmatrix} [0{\cdot}25922, 0{\cdot}25923] & [0{\cdot}058200, 0{\cdot}058201] & [0{\cdot}025065, 0{\cdot}025066] & [-0{\cdot}026483, -0{\cdot}026482] \\ [0{\cdot}058200, 0{\cdot}058201] & [0{\cdot}26233, 0{\cdot}26234] & [0{\cdot}028363, 0{\cdot}028364] & [0{\cdot}063540, 0{\cdot}063541] \\ [0{\cdot}025065, 0{\cdot}025066] & [0{\cdot}028363, 0{\cdot}028367] & [0{\cdot}19315, 0{\cdot}19316] & [0{\cdot}11565, 0{\cdot}11566] \\ [-0{\cdot}026483, -0{\cdot}026482] & [0{\cdot}063540, 0{\cdot}063541] & [0{\cdot}11565, 0{\cdot}11566] & [0{\cdot}27258, 0{\cdot}27259] \end{bmatrix},$$

where we have (outwardly) rounded the results to five significant decimal digits.

Now $M = 5 \times 10^{-3} E = 5 \times 10^{-3} e e^T$. Hence from (7.2) we may let $u = 5 \times 10^{-3} e$ and $v = e$. Noting $\bar{x}^I$, we see that $s = (1, 1, 1, -1)^T$ and noting $\overline{W}^I$, we have

$$k^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}, \qquad k^{(2)} = k^{(3)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad k^{(4)} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Also, $f = 5 \times 10^{-3} e$. We now have the necessary information to compute y_{rr} from (7.9) and z_{rr} from (7.10) for $r = 1$, 2, 3, and 4. We find

$$\begin{aligned} y_{11} &= [1{\cdot}0408, 1{\cdot}0409], & z_{11} &= [1{\cdot}0517, 1{\cdot}0518], \\ y_{22} &= [0{\cdot}55672, 0{\cdot}55673], & z_{22} &= [0{\cdot}56888, 0{\cdot}56889], \\ y_{33} &= [0{\cdot}10568, 0{\cdot}10569], & z_{33} &= [0{\cdot}11636, 0{\cdot}11637], \\ y_{44} &= [-0{\cdot}23518, -0{\cdot}23517], & z_{44} &= [-0{\cdot}22108, -0{\cdot}22107]. \end{aligned}$$

Again we have recorded results to only five significant digits.

The author is indebted to Mrs. Roberta Smith who procured the numerical results quoted in this chapter.

REFERENCES

1. ALBRECHT, J. Monotone Iterationsfolgen und ihre Verwendung zur Lösung linearer Gleichungssysteme. *Num. Math.* **3**, 345–58 (1961).

2. HANSEN, ELDON. Interval arithmetic in matrix computations, Part I. *SIAM Jl numer. Anal.* **2**, 308–20 (1965).

3. Hansen, Eldon, and Smith, Roberta. Interval arithmetic in matrix computations, Part II. Ibid. **4**, 1–9 (1967).

4. Kuperman, I. B. Approximate linear algebraic equations and rounding error estimation. Ph.D. thesis, Dept. of Applied Math., Univ. of Witwatersrand, Johannesburg (1967).

5. Oettli, H. On the solution set of a linear system with inaccurate coefficients. *SIAM Jl numer. Anal.* **2**, 115–18 (1965).

6. —— Prager, H., and Wilkinson, J. H. Admissible solutions of linear systems with not sharply defined coefficients. Ibid. **2**, 291–9 (1965).

7. Rigal, J. L. and Gaches, J. On the compatibility of a given solution with the data of a linear system. *J. Ass. comput. Math.* **14**, 543–8 (1967).

8. Smith, Roberta, and Hansen, Eldon. A computer program for solving a system of linear equations and matrix inversion with automatic error bounding using interval arithmetic. *Lockheed Missiles and Space Co. Report* LMSC4–22–66–3 (1968).

9. Varga, Richard S. *Matrix iterative analysis*. Prentice-Hall, New Jersey (1962).

5 · On the Estimation of Significance

1. Introduction

INTERVAL arithmetic is manifestly an important tool in digital computation and programming. When it is used in a naïve manner—as a simple technique for simulating a forward error analysis—interval arithmetic is, however, unable to give sharp bounds on the total computational error. As a matter of fact, to achieve that purpose, provision must be made for combining the results of local-error monitoring with estimates obtained by global analysis, which falls within the competence of the numerical expert. For a variety of important problems, algorithms have been found that answer the above purpose by using interval arithmetic together with other techniques; needless to say, such elaborate procedures usually require a much greater amount of computation time than the direct simulation in interval arithmetic of conventional numerical processes. Similar remarks obviously apply also to the other techniques for automatic error monitoring that have been studied and experimented with so far, namely: unnormalized arithmetic (see, for example, Ashenhurst and Metropolis [2]), normalized floating-point arithmetic with an index of significance (see, for example, Gray and Harrison [9]), and automatic controlled precision arithmetic (see, for example, Chartres [7]).

The purpose of this chapter is to contribute to the economical solution of the general problem of automatic-error estimation by emphasizing the most welcome fact that, in certain important situations, the pseudo-arithmetic effect of the accumulation of generated errors can be entirely disregarded with respect to the effect of propagation of inherent errors. It should be realized that any preliminary separation of those effects can be regarded as a major simplification of the original problem of error estimation provided that the number of guarding figures to be kept in the calculation to ensure such a separation can be readily estimated. This is typically the case for any numerical process that has been proved to be *'gutartig'* in the sense of Bauer [4]–[6]: indeed, from the equivalent condition that the natural instability dominate the numerical instability

of the process—both kinds of instability being measured by appropriate condition numbers—it follows that rounding phenomena can be entirely disregarded if the calculations are performed with (say) one more place than strictly needed to cope with the accuracy of the data. Section 3 will be concerned with an analysis of the concept of *gutartig* algorithm, which seems to be insufficiently known so far. To help the reader to realize the full implications of that important concept, we shall outline the very instructive proof, also due to Bauer [6], of the following remarkable property: the Gauss–Jordan scheme without any pivoting for size is *gutartig* if the matrix to be inverted is positive definite and Hermitian. It can be almost *gutartig* for more general matrices if a suitable pivoting strategy is followed.

The concept of a *gutartig* algorithm can also be defined in terms of compatibility requirements of computed results with prescribed tolerances. Since the general problem of the compatibility is important by itself, it will be discussed independently in section 2, mainly from a theoretical point of view. According to Bauer [5], the underlying criteria in forward and backward error analysis should be regarded only as extremely particular cases, the general criterion of compatibility being indeed defined by criterion (3). We shall show that, at least for matrix inversion, the set of admissible solutions in the Bauer sense can be much larger than the sets defined by the classical criteria. This property, which is supported here by simple convexity arguments, clearly has many interesting applications.

In section 4, we shall present a simple technique for the automatic estimation of the significance in matrix inversion (essentially in the positive definite case). It turns out that the concept of *gutartig* numerical processes and techniques like *a posteriori* estimation and unnormalized arithmetic are all relevant for the required procedure.

2. The problem of compatibility in digital computation

The following discussion of errors in digital computation is based essentially on a recent analysis (Bauer [4]–[6]) which seems to be insufficiently known considering its far-reaching implications. Since the original work is available only in the German language, it is hoped that the present survey will prove useful to a number of readers interested in error analysis and control. A certain familiarity with the topics discussed here will be assumed later.

Like Bauer [5], we shall take a *theoretical problem* (the inversion of a non-singular matrix, for example) to mean a single-valued mapping f

from the space $\mathscr{D}$ of data to a space $\mathscr{R}$ of results, it being understood for simplicity that a unique solution is assumed to exist. As a general rule, an *algorithm* devised (by the numerical analyst!) to perform the mapping f will actually evaluate a mapping f^* from $\mathscr{D}^* \subset \mathscr{D}$ to $\mathscr{R}^* \subset \mathscr{R}$. Any numerical process f^* is indeed essentially a machine-oriented program involving a finite number of finite precision-arithmetic operations and logical decisions. Accordingly it depends on parameters (for example, the stepsize of the discretization in continuous problems, the number of steps in the approximation of limit processes, etc.) and on pseudo-arithmetic details (for example, the word length of the machine, the rounding rules, etc.), which control the size of the accumulated errors resulting from finitude, namely, the *analytic* errors (due to discretization and truncation) and the *generated* errors (due to rounding). On the other hand, the numerical analyst is not responsible for the other two sources of error distinguished by von Neumann and Goldstine (see, for example, Householder [12]) although he should be concerned with the practical estimation of the range of uncertainty in $\mathscr{R}$ to which they give rise. Disregarding the errors due to possibly idealized formulations, we shall concentrate hereafter on a comparative study of the relative effects of *propagation of inherent errors* and *accumulation of generated errors*. It should indeed be realized that *inherent errors* are generally present, not only in the problems that are termed physical (Fox [8]) where they are due to experimental measurement, but also in most mathematical problems in view of the necessary restriction of $\mathscr{D}$ to the subspace $\mathscr{D}^* = g\mathscr{D}$ of machine numbers. Moreover, errors arising from the various sources result, irrespective of their complex interaction, in a single error on any intermediate result, which error can be interpreted as inherent for the subsequent computational steps.

An over-all assessment η of the size of errors arising in the practical evaluation of f is given by

$$\eta = \sup d(f^*A^*, fA) \quad \text{for } A^* = gA,\ A \in \mathscr{D}, \tag{1}$$

where d denotes an arbitrary distance function on $\mathscr{R}$. Needless to say, more refined measures could be used instead, specially by resorting to pseudo-distances in the sense of Schröder [21]. Any candidate f^*A^* for the approximation within a prescribed tolerance δ of the theoretical outcome fA can certainly be accepted if

$$fA \in \mathscr{S}_\delta(f^*A^*) = \{X \in \mathscr{R} : d(f^*A^*, X) \leqslant \delta\}. \tag{2}$$

This simple condition is too restrictive, however, when inherent errors

(of whatever source) are present. As readily verified, the criterion appropriate to that general case can be written in the form (given by Bauer [5])

$$\exists\, Y \in \mathscr{S}_{\delta'}(A^*) = \{Y \in \mathscr{D} : d'(A^*, Y) \leqslant \delta'\} : fY \in \mathscr{S}_{\delta}(f^*A^*), \tag{3}$$

where δ' denotes the tolerance within which the data are known, $\mathscr{D}$ being regarded here as a metric space with distance function d'. Criterion (3) reduces to (2) for $\delta' = 0$ (and $A^* = A$) and to

$$\exists\, Y \in \mathscr{S}_{\delta'}(A^*) : fY = f^*A^* \tag{4}$$

for $\delta = 0$, these specialized criteria being classically appropriate to *forward* and *backward error analysis*, respectively. As emphasized recently (see, for example, Oettli *et al.*, [17]–[19] and Rigal and Gaches [20]) in connection with important algebraic problems (linear equations, polynomial and eigenvalue problems), the combination of elementary techniques of backward error analysis and *a posteriori* estimation proves extremely useful. It leads indeed, among others, to simple criteria for deciding, on a realistic basis and independently of the numerical process itself, whether a given candidate for an approximate solution can be accepted under the given constraints of accuracy. On the other hand, the general criterion (3) has not received much attention so far and does not seem to have been sufficiently exploited in concrete situations in spite of its obvious theoretical importance.

To exemplify the actual gain in significance that can be achieved by using criterion (3) instead of (4) or (2), we shall outline a discussion (to be completed in a further paper elsewhere) of the compatibility problem in matrix inversion, answering thereby a suggestion made by Wilkinson (personal communication). Proceeding first in the spirit of (4), we must regard a computed (right-hand) inverse X of a given non-singular matrix A as the exact inverse of some perturbed matrix $Y = A + \Delta A$. The basic identities

$$R = I - A.X = \Delta A.X, \tag{5a}$$

$$X - A^{-1} = -A^{-1}.R \tag{5b}$$

then yield, among others, the well-known *a priori* estimates of the relative error in X,

$$\mathrm{cond}^{-1}(X)\|\Delta A\|/\|A\| \leqslant \|X - A^{-1}\|/\|X\| \leqslant \mathrm{cond}(A)\,\|\Delta A\|/\|A\|, \tag{6}$$

and the corresponding *a posteriori* estimates

$$(\|X\|\,\|A\|)^{-1}\|R\| \leqslant \|X - A^{-1}\|/\|X\| \leqslant (\|A^{-1}\|/\|X\|)\,\|R\|, \tag{7a}$$

$$\mathrm{cond}^{-1}(X)\,\|R\| \leqslant \|\Delta A\|/\|X^{-1}\| \leqslant \|R\|, \tag{7b}$$

which are clearly of much more value in practice (the main purpose of an *a priori* analysis is indeed to reveal the basic weaknesses of a numerical

process and to gain some idea of its fundamental limitations). The above appraisals take a much simpler form, lub norms being used throughout (so that $\|I\| = 1$), if

$$\|\Delta A\| \cdot \|A^{-1}\| \ll 1. \tag{8}$$

This may be regarded as a fairly natural assumption in view of the identity

$$(I + A^{-1}.\Delta A)X = A^{-1}. \tag{5c}$$

As a matter of fact, the approximations $\|A^{-1}\| \sim \|X\|$, $\|A\| \sim \|X^{-1}\|$, $\text{cond}\,(A) = \|A\| \cdot \|A^{-1}\| \sim \text{cond}\,(X) \sim \|A\| \cdot \|X\|$ are then all admissible within a very low relative error. According to (4), an alleged inverse X of A (computed by any technique) can be accepted if and only if the perturbation matrix ΔA satisfies

$$\|\Delta A\| \leqslant \delta' = \delta''\|A\|. \tag{9a}$$

In this case the right-hand inequality in (6) takes the form

$$\|X - A^{-1}\|/\|X\| \leqslant \delta'\|A^{-1}\| = \delta'' \,\text{cond}\,(A) = \delta/\|X\|. \tag{9b}$$

As emphasized by Wilkinson ([23], see p. 319), the well-known fact that the appraisal (9b) is sharp on the sphere (9a) does not at all imply that, conversely, any X satisfying (9b) is the exact inverse of some matrix $A + \Delta A$ for which (9a) holds. Roughly this disappointing result is supported by the left-hand inequality in (6) which is known to be sharp on the sphere (9b).

It should be realized that the shortcomings inherent in the elementary appraisals (6) and (7) cannot be remedied as long as $\mathscr{S}_{\delta'}(A)$ and $\mathscr{S}_{\delta}(X)$ are the only convex sets under consideration. This simple remark suggests an alternative but less convenient approach we shall now briefly explain. According to criterion (3), an alleged inverse X of A (computed by any method) can be accepted if and only if there exist perturbation matrices ΔX and ΔA related by $(A + \Delta A)(X + \Delta X) = I$ or equivalently by

$$R = I - AX = \Delta A.X + A.\Delta X + \Delta A.\Delta X \tag{10}$$

under the constraints

$$\|\Delta A\| \leqslant \delta', \tag{11a}$$

$$\|\Delta X\| \leqslant \delta, \tag{11b}$$

where δ' and δ are the prescribed tolerances. Assuming for simplicity that (8) holds true, which should be the case as a general rule, we are justified in disregarding the quadratic term in (10) and in replacing accordingly the unknown A^{-1} by the known X whenever it appears as a factor in the appraisals. Then, to exploit criterion (3), we have only

to verify whether the computed residual matrix R belongs to the equilibrated convex body defined by the so-called *Minkowski sum*

$$\mathscr{S} = \mathscr{S}_{\delta'}(0) . X + A . \mathscr{S}_{\delta}(0), \tag{12}$$

where $\mathscr{S}_{\delta'}(0) \subset \mathscr{D}$ and $\mathscr{S}_{\delta}(0) \subset \mathscr{R}$ are clearly the spheres defined by (11 a) and (11 b), respectively (for a survey of the theory of norms and convexity, see, for example, Householder [13]). In particular, according as $\delta = 0$ (backward analysis) or $\delta' = 0$ (forward analysis), the criterion of compatibility $R \in \mathscr{S}$ reduces to the respective conditions

$$\|R . A\| \leqslant \delta', \tag{13 a}$$

$$\|X . R\| \leqslant \delta, \tag{13 b}$$

which are classical to a certain extent in *a posteriori* analyses. In the general case, however, such simplifications are not allowed, so that *distance functions* and *support functions* cannot be dispensed with for describing completely the three convex sets appearing in (12). Anyway, the above geometric interpretation of $\mathscr{S}$ shows that criterion (3) can be regarded as a most significant extension of both criteria (2, 13 b) and (4, 13 a). Indeed, in current practice, the Minkowski sum turns out to be much larger than either component subset.

3. The concept of *gutartig* numerical process

For any theoretical problem that is *well-posed* in the Hadamard sense (see, for example, Isaacson and Keller [14]), the distance functions d and d' may be chosen arbitrarily, provided only that the mapping f is bounded (i.e. Lipschitz continuous on its domain), which means that

$$d(fA^*, fY) \leqslant l . d'(A^*, Y) \quad \text{for } Y \in \mathscr{S}_{\delta'}(A^*), \tag{14}$$

where l is a Lipschitz constant. Then, whenever condition (3) is fulfilled, the triangle inequality yields, for example, the appraisal

$$d(f^*A^*, fA^*) \leqslant d(f^*A^*, fY) + d(fA^*, fY) \leqslant \delta + l . \delta', \tag{15}$$

where the right-hand bound is seen to reflect the relative ill-posedness of f and may be large accordingly. As supported by (15), we can hardly expect that a computed f^*A^* will be more accurate than inherent errors allow. More specifically, a high accuracy of the outcome, say within δ, cannot be guaranteed from compatibility arguments alone.

The intersection of all spheres $\mathscr{S}_{\delta}(f^*A^*)$ containing at least one point (distinct from the centre f^*A^*) of the space $\mathscr{R}^*$ (i.e. $\mathscr{R}$ restricted by rounding) is clearly a sphere $\mathscr{S}_{\epsilon}(f^*A^*)$ whose radius ϵ is strictly positive and depends mainly on the word length of the machine. For example,

in p-place binary fixed-point computations, ϵ will denote $2^{-p}/2$ or $n \cdot 2^{-p}/2$ according as $\mathscr{R}$ is a metric space of numbers (with the absolute distance function) or a normed space of n-dimensional matrices (with the Euclidean norm). Of course, a similar quantity ϵ' must be defined in $\mathscr{D}$ too, whenever the data cannot be represented exactly by machine numbers. Then the most we have a right to demand of a numerical process f^* is that it be compatible, in the sense of criterion (3), with tolerances δ and δ' not greater than ϵ and ϵ', respectively. Any such algorithm is said to be *gutartig* (this term has been coined by Bauer [4], p. 64) and should be regarded as achieving the best possible fit of the theoretical requirements of accuracy to the practical limitations resulting from the finite precision of digital computation.

From the above definition it follows indeed that the global effect of the accumulation of generated errors for any *gutartig* algorithm is dominated by the effect of propagation of inherent errors of the size ϵ' however well-posed the theoretical problem may be. Because of this most welcome property, rounding phenomena can be entirely disregarded if the calculations are performed with, say, one more place than strictly needed to cope with the over-all accuracy of the data. The problem of error estimation then reduces, at least if analytic errors can be disregarded or controlled by other means, to the study of the propagation of inherent errors, which phenomenon is independent of the mode of computation. In other words (see Bauer ([4], [6])), it may be said that an algorithm is *gutartig* if the *natural instability* (i.e. the over-all susceptibility of the solution to perturbations in the data) dominates the *numerical instability* (i.e. the over-all sensitivity of the solution with respect to perturbations in all the intermediate results that may be subject to rounding errors). Both types of instability are to be measured by *condition numbers* appropriate to the pseudo-arithmetic used (in floating-point computation, for example, relative measures are generally preferred to absolute ones).

Since the calculation tree concerned with the propagation of any single generated error is necessarily obtained by extraction from the process graph representing the whole of a (*gutartig*) algorithm, it is hardly to be expected that some Y exists such that the conditions $d(f^*A^*, fY) \leqslant \epsilon$ and $d'(A^*, Y) \ll \epsilon'$ are both fulfilled. That is, however, the case for certain important applications where it can be proved that the total effect of all the rounding errors made in the process of solution is indeed far less than those which come from the inherent errors. By way of illustration, we may consider for example the inversion, working with eight decimals, of a symmetric segment of order five of the Hilbert matrix;

the detailed discussion that can be found in Wilkinson ([23], p. 319) shows that

$$\max|H^{-1}-(H^*)^{-1}|_{ij} \sim 6000, \tag{16a}$$

whereas

$$\max|(H^*)^{-1}-X|_{ij} \sim 1, \tag{16b}$$

where H denotes the exact Hilbert segment, H^* the same matrix with the elements truncated to eight decimals, and X the approximate inverse of H^* computed by an appropriate Gaussian technique.

Certain well-known algorithms are manifestly not *gutartig*. The most overworked examples (in floating-point arithmetic) are probably the calculation of the smallest (real) root of a quadratic equation from its classical definition in terms of the coefficients and the calculation of the eigenvalues of a Hermitian matrix from its characteristic polynomial. It can be proved, indeed, that, as a general rule, the instability with respect to generated errors affecting certain intermediate results (the coefficients of the characteristic polynomial, for example) dominates the natural instability. In other words, an algorithm is certainly not *gutartig* if an error-decreasing section is followed by an error-increasing section since then any error generated at the beginning of the latter section can only be magnified, whereas inherent errors (and generated errors of the former section) are damped before being all magnified in the same way. Specifically, all situations where an error-decreasing section is followed by cancellation effects should be systematically avoided in single-precision floating-point computation.

It is unnecessary to say that *gutartig* algorithms are still unknown for many theoretical problems. As a matter of fact, the very question whether a given algorithm is *gutartig* is a challenging matter. Any comparison on a common basis of the respective effects of accumulation of generated errors and of propagation of inherent errors indeed requires an *a priori* analysis where special care has to be taken, however, to obtain appraisals that are really sharp. It follows that the one-stage error estimates, whose concatenation is typical of elementary forward techniques and of automatic error monitoring, cannot be used blindly. In principle, more elaborate forward techniques (devised at the level of procedures, for example) and backward techniques cannot be dispensed with. In this respect, the theoretical and practical importance of Wilkinson's work in the field of matrix computations and related algebraic processes (see, for example, Wilkinson [24], [25]) can hardly be exaggerated. In current practice it may happen that actually *gutartig* algorithms are found to be only 'almost *gutartig*' in view of the obvious fact that *a priori* estimates are too liberal, which unduly magnifies rounding effects,

whereas the propagation of inherent errors is governed by the theoretical matrix of condition numbers. If an algorithm has been proved to be almost *gutartig*, then the effects of pure propagated errors and round-off can be automatically separated by keeping a small number of *guarding figures* in the calculation (for an interesting illustration of this type of argument—'lengthy' Gauss elimination process in fixed-point arithmetic with partial pivoting—see, for example, Fox [8], pp. 159–67).

Among other significant results, it must be emphasized here that Gaussian elimination is a *gutartig* numerical process when applied to positive definite Hermitian matrices and that the complete pivoting scheme is almost *gutartig* as a general rule. As the following outline is intended to show, the proof (due to Bauer [6]) of this most important property turns out to be surprisingly simple and particularly instructive. Let $A^{(k)}$ denote the intermediate theoretical matrix after the first k exchange-steps; in particular, $A^{(0)}$ coincides with the given non-singular n-square matrix A and $A^{(n)}$ coincides with the theoretical inverse $B = A^{-1}$. By proceeding, for example, as explained by Stiefel [22], it is easily seen that

$$A^{(k)} = \begin{pmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12} \\ A_{21}A_{11}^{-1} & A_{22}-A_{21}A_{11}^{-1}A_{12} \end{pmatrix}$$

$$= \begin{pmatrix} B_{11}-B_{12}B_{22}^{-1}B_{21} & B_{12}B_{22}^{-1} \\ -B_{22}^{-1}B_{21} & B_{22}^{-1} \end{pmatrix}, \tag{17}$$

where A_{ij} and B_{ij} $(i, j = 1, 2)$ denote the blocks into which A and $B = A^{-1}$ are partitioned conformally, A_{11} and B_{11} being k-square. Whatever form of pivoting is used, (17) is true (in infinite precision arithmetic) provided A is used to denote the original matrix with its rows and columns suitably permuted; it is known, however, that no pivoting for size is required in the positive definite case (see, for example, Wilkinson [23], p. 285). It follows directly from (17) that a Gauss–Jordan scheme can be *gutartig* only if the pivotal strategy is such that none of the bottom right-hand corner principal submatrices B_{22}^{-1} is more ill-conditioned than A itself with respect to the calculation of the inverse. Formulae (9) accordingly yield the following set of necessary conditions for the numerical process to be *gutartig*:

$$\text{cond}\,(B_{22}) \leqslant \text{cond}\,(A) \quad \text{for } 1 \leqslant k \leqslant n, \tag{18}$$

where suitable condition numbers must still be chosen.

If A is positive definite and Hermitian, then conditions (18), where spectral condition numbers are used, are automatically fulfilled. This is in fact a straightforward consequence of the well-known separation

theorem (see, for example, Householder [13], p. 76) which states that the eigenvalues of any $(n-1)$-dimensional section of a Hermitian n-matrix A separate those of A. To establish that the Gauss–Jordan scheme (without any pivoting for size) is actually *gutartig* in the present case, it remains essentially to prove that the theoretical inverse $B = A^{-1}$ is not more sensitive to perturbations $\Delta A_{ij}^{(k)}$ in the blocks appearing in (17) than to perturbations ΔA_{ij} in the original blocks A_{ij} themselves. Of course, this *a priori* comparison of the respective conditions of the conformally partitioned matrices $A^{(k)}$ and $A = A^{(0)}$ with respect to the calculation of the blocks B_{ij} can depend to a certain extent on pseudo-arithmetic details since the perturbations $\Delta A_{ij}^{(k)}$ must be interpreted as originating from pure round-off. More precisely, in floating-point computation, an elementary backward analysis of the generated errors for the kth exchange-step yields the componentwise appraisal

$$|\Delta A_{ij}^{(k)}| \leqslant |A_{ij}^{(k)}| 9\epsilon \quad \text{for } i, j = 1, 2 \tag{19}$$

with

$$\epsilon = (\beta/2)\beta^{-p}, \tag{20}$$

where β is the base (usually 2 or 10) and p is the precision of the digital representation (i.e. the number of digits in the mantissa). By using intermediately Euclidean norms $\|.\|_E$, it follows from (19) that the spectral lub norms of the perturbation blocks are restricted by

$$\operatorname{lub}(\Delta A_{ij}^{(k)}) \leqslant \|\Delta A_{ij}^{(k)}\|_E \leqslant \|A_{ij}^{(k)}\|_E 9\epsilon \leqslant \operatorname{lub}(A_{ij}^{(k)})\gamma(i,j)9\epsilon, \tag{21 a}$$

where

$$\gamma(i,1) = k^{\frac{1}{2}}, \qquad \gamma(i,2) = (n-k)^{\frac{1}{2}}. \tag{21 b}$$

The next step in Bauer's proof consists in determining sharp upper bounds (in terms of only the spectral lub norms of A and A^{-1}) for each $\operatorname{lub}(A_{ij}^{(k)})$. This is readily done for the diagonal blocks in view of the separation theorem recalled above. For the other two blocks, a remarkable extension (also due to Bauer) of the Kantorovich inequality must be exploited. In view of the basic identity

$$\Delta A^{-1} = \begin{pmatrix} I & B_{12}+\Delta B_{12} \\ 0 & B_{22}+\Delta B_{22} \end{pmatrix} \begin{pmatrix} \Delta A_{11}^{(k)} & \Delta A_{12}^{(k)} \\ -\Delta A_{21}^{(k)} & -\Delta A_{22}^{(k)} \end{pmatrix} \begin{pmatrix} I & 0 \\ B_{21} & B_{22} \end{pmatrix}, \tag{22}$$

which can be verified by an explicit algebraic calculation, a sharp estimate (of the prescribed form) for the relative error (in the spectral norm) in the inverse can be obtained from the above appraisals. The final result is of a fairly complicated form (see Bauer [6], p. 418) unless $\operatorname{cond}(A) \gg 1$. In that especially important case it reduces indeed to

$$\operatorname{lub}(\Delta A^{-1})/\operatorname{lub}(A^{-1}) \leqslant \operatorname{cond}(A).(n-k)^{\frac{1}{2}}\epsilon_k, \tag{23 a}$$

where

$$\epsilon_k = 9\epsilon/[1-\operatorname{cond}(A).(n-k)^{\frac{1}{2}}9\epsilon]. \tag{23 b}$$

By using (5 c) and (21) (extended to the case $k = 0$), it turns out that the appraisal (9) coincides with (23) where $k = 0$. This simple remark completes the proof that the algorithm is *gutartig*.

If A is not a positive definite Hermitian matrix, then neither the separation theorem nor the Kantorovich inequality hold true and pivoting for size must in any case here be taken into account. An appropriate approach (in floating-point computation) consists of the direct application of the componentwise appraisal (19) to the identity (22), which yields the indeed remarkable result

$$|\Delta A^{-1}| \leqslant |A^{-1}|9\epsilon + |A^{-1}| \begin{pmatrix} 0 & 0 \\ 0 & |A_{22}^{(k)}| \end{pmatrix} |A^{-1}|36\epsilon. \tag{24 a}$$

On the other hand, by again using (19) (extended to the case $k = 0$), it follows from (5) that

$$|\Delta A^{-1}| \leqslant |A^{-1}| \cdot |A| \cdot |A^{-1}|9\epsilon + O(\epsilon^2) \tag{24 b}$$

if the spectral radius $\rho(|A^{-1}| \cdot |A|9\epsilon) < 1$. The comparison of (24 a) with (24 b) reveals that the Gauss–Jordan scheme is certainly almost *gutartig* if for any k the smallest $p^{(k)}$ satisfying

$$|A^{-1}| \begin{pmatrix} 0 & 0 \\ 0 & |A_{22}^{(k)}| \end{pmatrix} |A^{-1}| \leqslant p^{(k)} |A^{-1}| \cdot |A| \cdot |A^{-1}| \tag{25}$$

is of the order of 1; this is automatically the case when A is positive definite for it has been proved (see, for example, Wilkinson [23], p. 285) that no element in any reduced matrix $A_{22}^{(k)}$ exceeds the maximum element in the original matrix A. An alternative approach, also followed by Bauer [6], is based upon the Frobenius theory of non-negative matrices with which polyhedral norms are known to be intimately associated. The Perron root $\pi(|A|)$ of a non-negative matrix $|A|$ may indeed be regarded as the norm associated with a certain equilibrated polytope of a simple structure (see, for example, Householder [13], p. 49). As emphasized by Bauer [3], it plays a prominent role in the problem of optimal scaling of matrices with respect to the inversion problem. A typical result in this connection is the following one: if lub (A) is subordinate to the maximum norm, then

$$\min_{D_1, D_2} \operatorname{cond}(D_1 A D_2) = \pi(|A| \cdot |A^{-1}|), \tag{26}$$

where D_1 and D_2 denote arbitrary non-singular diagonal matrices which can be regarded as achieving a certain 'equilibration' of the given matrix A (it should be stressed that the $p^{(k)}$ in (25) are invariant with respect to any diagonal scaling). As to the pivoting strategy to be followed for the Gauss–Jordan scheme to be almost *gutartig* (if that is

possible at all for the given A), formulae (24) and (26) show that it should be such that (for $1 \leqslant k \leqslant n$)

$$\text{lub}\,(|B_{22}| \cdot |B_{22}^{-1}|) \sim \text{lub}\,(|A| \cdot |A^{-1}|), \tag{27 a}$$

or

$$\pi(|B_{22}| \cdot |B_{22}^{-1}|) \sim \pi(|A| \cdot |A^{-1}|). \tag{27 b}$$

These necessary conditions are to be compared with the conditions (18) that must hold in the positive definite case.

4. On the automatic estimation of significance in matrix inversion

The foregoing analysis has emphasized the welcome fact that, for certain algorithms, the total effect of the accumulation of generated errors is dominated by the effect of propagation of inherent errors (of whatever source) and therefore can be entirely disregarded provided the *precision* p exceeds the minimum precision k that is required to accommodate the most precise input of interest. Taking best advantage of this characteristic property of *gutartig* algorithms, we now present a surprisingly simple procedure designed to provide automatic error monitoring and control in matrix inversion.

By way of numerical illustration, we consider the inversion, using 6-decimal floating-point arithmetic, of the symmetric segment

$$H = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{pmatrix} \tag{28}$$

of the Hilbert matrix. The entry 1/3 cannot be represented exactly by a finite number of decimals; its truncation to, say, $k = 5$ decimal digits involves replacing H by the slightly perturbed matrix

$$A = \begin{pmatrix} 1 & 0{\cdot}5 & 0{\cdot}333330 \\ 0{\cdot}5 & 0{\cdot}333330 & 0{\cdot}25 \\ 0{\cdot}333330 & 0{\cdot}25 & 0{\cdot}2 \end{pmatrix}, \tag{29}$$

the matrix of inherent errors being accordingly

$$\Delta H = A - H = -(10^{-5}/3)\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{30}$$

Like H, A is a positive definite matrix. The Gauss–Jordan scheme (without any pivoting for size and even without a double-precision

accumulator) is consequently a *gutartig* numerical process. Then, since $k < p = 6$, the computed inverse

$$X = \begin{pmatrix} 10^{1}(0{\cdot}900622) & 10^{2}(-0{\cdot}360327) & 10^{2}(0{\cdot}300306) \\ 10^{2}(-0{\cdot}360326) & 10^{3}(0{\cdot}192171) & 10^{3}(-0{\cdot}180159) \\ 10^{2}(0{\cdot}300306) & 10^{3}(-0{\cdot}180159) & 10^{3}(0{\cdot}180148) \end{pmatrix} \tag{31}$$

of A is the theoretical inverse of a perturbed matrix $H+\Delta H^*$, which is known to be such that the following estimate,

$$\Delta H^{-1} = X - H^{-1} = -H^{-1}.\Delta H^*.X \sim -H^{-1}.\Delta H.X, \tag{32}$$

is necessarily sharp. This fundamental consequence of the *gutartig* character of the algorithm is most remarkable, for it turns out that ΔH^* is here quite different from ΔH. From the right-handed residual matrix (computed in double-precision floating-point arithmetic and rounded to six decimals)

$$I - H.X = \Delta H^*.X$$

$$= \begin{pmatrix} 10^{-3}(-0{\cdot}120000) & 10^{-3}(0{\cdot}200000) & 10^{-3}(-0{\cdot}433333) \\ 10^{-3}(0{\cdot}106667) & 10^{-3}(-0{\cdot}900000) & 10^{-3}(0{\cdot}700000) \\ 10^{-4}(-0{\cdot}433333) & 10^{-4}(-0{\cdot}500000) & 10^{-4}(-0{\cdot}500000) \end{pmatrix}, \tag{33}$$

which is such that $\|\Delta H^*.X\|_\infty \sim 0{\cdot}0017 \ll 1$, it follows indeed that the estimate

$$\Delta H^* \sim (\Delta H^*.X)H = -10^{-5}\begin{pmatrix} 16{\cdot}44 & 10{\cdot}17 & 7{\cdot}67 \\ 11{\cdot}00 & 7{\cdot}17 & 4{\cdot}94 \\ 8{\cdot}50 & 5{\cdot}08 & -3{\cdot}69 \end{pmatrix} \tag{34}$$

is certainly correct to two significant figures in view of the identity

$$\Delta H^* = (\Delta H^*.X)(I - \Delta H^*.X)^{-1}H. \tag{35}$$

In consequence of the smallness of the norm of the residual matrix (33), H^{-1} can also be replaced by X in the estimate on the right of formula (32) which then yields, using (30) and (31),

$$H^{-1} \sim X + X.\Delta H.X$$

$$= \begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix} + 10^{-4}\begin{pmatrix} 0{\cdot}9 & -6 & 5 \\ -5 & 50 & -40 \\ 5 & -40 & 40 \end{pmatrix}. \tag{36}$$

Here the first matrix on the right is the well-known theoretical inverse of the Hilbert segment (28).

In principle, the above method of improving the accuracy of an approximate inverse computed by a *gutartig* numerical process can be used only if ΔH is known. It should be realized, however, that the basic estimate (32) proves to be useful (provided the replacement on the right

of H^{-1} by X is justified) in any case where correlated patterns of inherent errors must be taken into account. Whenever the uncertainties in the elements of the matrix A to be inverted can be regarded as entering in a random pattern, which is typically the case for physical problems, an alternative procedure can be exploited, as discussed hereafter, to assess automatically for any entry of the computed inverse X the number of figures that have a meaning and are accordingly worth quoting. *Unnormalized floating-point arithmetic*, with the so-called *significance rules*, is known to provide a proper setting for the automatic estimation of significance, at least as far as the propagation effects of inherent errors are concerned. In infinite precision (binary) unnormalized arithmetic, any non-zero number $x = 2^e.f$, where the *exponent* e is an integer and the *fractional part* f satisfies $0 < |f| < 1$, can be represented by any one of the equivalent ordered pairs $(e-s, 2^s.f)$ where, however, the *adjustment* parameter s may not exceed the number m of *leading zeros* of f. The degree of freedom provided by the flexibility of the unnormalized format can be used to adjust each input operand so that the least significant digit resides in some specified position called the *error front*, say k, within the format of available digits, say p. This particular adjustment obviously reflects the accuracy of the operand since the *coefficient error* d (i.e. the absolute error in the adjusted fractional part) is readily estimated from the number k of *correct figures* by

$$|d| \leqslant 2^{-k}/2. \tag{37}$$

As to the $p-k$ digits on the right, they can serve as guarding figures to separate the effect of inherent errors from that of generated errors. Whenever the stability of the error front is guaranteed the errors in the output numbers can be correctly assessed by mere inspection, which is actually the essential purpose of the unnormalized schemes and can be reasonably achieved if all participating operands in an algorithm are statistically independent. Indeed, for the arithmetic operations expressed in the symbolic form

$$(e_3,f_3) = (e_1,f_1) * (e_2,f_2), \tag{38 a}$$

the following rules of thumb

$$e_3 = \max(e_1, e_2) \quad \text{(in addition/subtraction)}, \tag{38 b}$$

$$m_3 = \max(m_1, m_2) \quad \text{(in multiplication/division)} \tag{38 c}$$

(with appropriate modifications in special cases such as overflow, zero operands, etc.) turn out to work satisfactorily, at least if the operands are not correlated. The original proof, which is due to Ashenhurst and

Metropolis [2] for the base $\beta = 2$ (see Meinguet [15] for the extension to other bases), is framed in terms of the *amplification factor*

$$\alpha = |d_3|/\max(|d_1|, |d_2|), \tag{39}$$

which ideally should be close to 1. In actual fact, the expected value of α is indeed close to 1 whereas α itself is only bounded by 2 (in addition/subtraction), by 3 (in multiplication), and by 4 (in division). It should be realized that the adjustment rules (38 b, c) unify in a most natural way the well-conditioned rules of fixed-point and floating-point arithmetics so that erratic error build-up of any kind is impossible anyhow. More details on unnormalized arithmetics and their interesting applications can be found in the many papers that have been devoted to that challenging matter (for a survey and an extensive list of references, see, for example, Meinguet [15], [16]). Another related technique for error monitoring should be mentioned here; namely normalized floating-point arithmetic with an *index of significance* (see Gray and Harrison [9]). It must be emphasized, however, that this index scheme, like interval arithmetic, involves carrying and manipulating more information than a simple ordered pair (e,f).

As we now explain, the number of figures that are worth quoting for any element of the computed inverse X can be automatically estimated by resorting to significant digit arithmetic, at least if the uncertainties in the entries of the given matrix A can be regarded as statistically independent. Under that assumption, if the elements of A are adjusted so that the least significant digit of the fractional part resides in a common digit position k of a word (at the option of the programmer), then it can be expected that 'on the average' the elements of the right-handed residual matrix (calculated in double-precision unnormalized arithmetic) will be automatically 'lined-up' on the right at the kth digit position according to the last significant digit (whereas in floating-point arithmetic the fractional parts are lined-up on the extreme left according to the first significant digit). As a matter of fact, this stabilizing effect of unnormalized arithmetic on the error front is not a straightforward consequence of the general properties of α we have recalled above, since the elements of X enter here as exact parameters. Its actual justification can be found in Ashenhurst [1] where the significance rules (38) are shown to be adequate in this special case too, at least from the point of view of expected value. Let the index S distinguish the matrices whose elements are correctly adjusted in the above sense. Choosing $k = 5$, we obtain by the significance rules from the given matrix (29) and its

computed inverse (31) the double-precision unnormalized residual matrix

$$R_s = I - A_s.X$$
$$= \begin{pmatrix} 10^2(-0{\cdot}000000\,198980) & 10^2(-0{\cdot}000004\,005300) & 10^2(0{\cdot}000001\,671600) \\ 10^2(-0{\cdot}000000\,134420) & 10^2(-0{\cdot}000002\,594300) & 10^2(0{\cdot}000000\,994700) \\ 10^1(-0{\cdot}000001\,331260) & 10^2(-0{\cdot}000001\,701090) & 10^2(0{\cdot}000000\,501020) \end{pmatrix}. \tag{40}$$

In the present case where A and A_s are both regarded as close approximations to the theoretical H, the elements of A_s must all be represented in the normalized 6-digit format and are therefore identical with the corresponding elements of A. It should be clearly realized, however, that $A_s \neq A$: indeed, all entries of A_s are regarded as correct to five significant digits whereas all elements of A, with only the exception of 0·333330, are regarded as correct to six decimals. The second (and final) part of our procedure for estimating the significance of the computed X consists of exploiting, in unnormalized arithmetic too, the remarkable identity

$$(A^{-1})_s = X(A_s.X)^{-1} = X.(I-R_s)^{-1}. \tag{41}$$

For the reasons mentioned above, (41) yields indeed the correctly adjusted inverse of A, at least if the inverse on the right is itself correctly adjusted. This basic idea of reducing a given problem concerning general data to the same problem concerning appropriately specialized data (here X is indeed replaced by $I-R \sim I$) has often been applied to problems in interval arithmetic (see, for example, Hansen [10] and Hansen and Smith [11]).

Since X has been calculated by a *gutartig* numerical process for $k = 5 < p = 6$, the adjusted error $(X-A^{-1})_s$ is due to round-off only and therefore must coincide with the zero matrix up to the error front. It follows that it would be pointless to modify any digit in X_s, so that in the single-precision unnormalized format the last factor on the right in (41) must reduce 'column-wise' to the unit matrix. Matrix (40) shows that the minimum choice of the adjustment parameter achieving that reduction is 1 for each of the three columns. Hence the required result is

$$(A^{-1})_s = X_s = \begin{pmatrix} 10^3(0{\cdot}009006) & 10^4(-0{\cdot}003603) & 10^4(0{\cdot}003003) \\ 10^4(-0{\cdot}003603) & 10^5(0{\cdot}001921) & 10^5(-0{\cdot}001801) \\ 10^4(0{\cdot}003003) & 10^5(-0{\cdot}001801) & 10^5(0{\cdot}001801) \end{pmatrix}, \tag{42}$$

which is seen to be correct to five decimals. It should be remarked that the complete calculation of the matrix (40) is not at all required. Only the exponent and the number of leading zeros for each single-precision

fractional part are used in the estimation of the adjustment parameters. However, the calculations must be organized in such a way that the relative errors made in computing the residual matrix are small.

On the other hand, if the accumulation effect of the generated errors during matrix inversion is not dominated by the effect of propagation of inherent errors, then the foregoing procedure turns out to be 'conservative' in the sense that certain digits are discarded whereas they would become meaningful by iterative refinement of the inverse (see, for example, Wilkinson [24], p. 121). In such cases, provided that $\|R\| \ll 1$, we can replace $(I-R_s)^{-1}$ in (41) by the approximate matrix $I+R_s$ so that

$$(A^{-1})_s \sim X+X.R_s \tag{43}$$

where unnormalized arithmetic must be used throughout. Using for example the data (31) and (40), we obtain for the correction term in (43) the unnormalized matrix

$$X.R_s = \begin{pmatrix} 10^3(-0{\cdot}094640) & 10^4(0{\cdot}063223) & 10^4(-0{\cdot}057410) \\ 10^4(0{\cdot}053220) & 10^5(-0{\cdot}047761) & 10^5(0{\cdot}040657) \\ 10^4(-0{\cdot}057409) & 10^5(0{\cdot}040657) & 10^5(-0{\cdot}038747) \end{pmatrix} 10^{-6}, \tag{44}$$

which can be used to estimate separately each adjustment parameter and to correct significant rounding errors. Of course, in the present case, (43) yields the same results (up to the error front) as (41), namely (42).

REFERENCES

1. Ashenhurst, R. L. Function evaluation in unnormalized arithmetic. *J. Ass. comput. Mach.* **11**, 168–87 (1964).
2. —— and Metropolis, N. Unnormalized floating-point arithmetic. Ibid. **6**, 415–28 (1959).
3. Bauer, F. L. Optimally scaled matrices. *Num. Math.* **5**, 73–87 (1963).
4. ——, Heinhold, J., Samelson. K., and Sauer, R. *Moderne Rechenanlagen*, Chap. 3. Teubner, Stuttgart (1965).
5. —— Numerische Abschätzung und Berechnung von Eigenwerten nichtsymmetrischer Matrizen. *PMM*, **10**, 178–89 (1965).
6. —— Genauigkeitsfragen bei der Lösung linearer Gleichungssysteme. *Z. angew. Math. Mech.* **46**, 409–21 (1966).
7. Chartres, B. A. Automatic controlled precision calculations. *J. Ass. comput. Mach.* **13**, 386–403 (1966).
8. Fox, L. *An introduction to numerical linear algebra*. Clarendon Press, Oxford (1964).
9. Gray, H. L., and Harrison, C. Jr. Normalized floating-point arithmetic with an index of significance. *Proc. east. jt Computer Conf.* **16**, 244–8 (1959).
10. Hansen, E. Interval arithmetic in matrix computations, Part I. *SIAM Jl numer. Anal.*, Ser. B, **2**, 308–20 (1965).

11. HANSEN, E. and SMITH, R. Interval arithmetic in matrix computations, Part II. *SIAM Jl numer. Anal.* **4**, 1–9 (1967).

12. HOUSEHOLDER, A. S. *Principles of numerical analysis*, Chap. 1. McGraw-Hill, London (1953).

13. —— *The theory of matrices in numerical analysis*, Chap. 2. Blaisdell Publ. Co., London (1964).

14. ISAACSON, E. and KELLER, H. B. *Analysis of numerical methods*, Chap. 1. Wiley, London (1966).

15. MEINGUET, J. Le contrôle des erreurs en calcul automatique. *Laboratoire de recherches M.B.L.E. Rapport R 29*. Bruxelles (1965).

16. —— Sens et portée des arithmétiques de signification. Proceedings of the Colloque international du C.N.R.S., Besançon, 1966 (in press).

17. OETTLI, W. On the solution set of a linear system with inaccurate coefficients. *SIAM Jl numer. Anal.*, Ser. B, **2**, 115–18 (1965).

18. —— and PRAGER, W. Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides *Num. Math.* **6**, 405–9 (1964).

19. —— —— and WILKINSON, J. H. Admissible solutions of linear systems with not sharply defined coefficients. *SIAM Jl numer. Anal.*, Ser. B, **2**, 291–9 (1965).

20. RIGAL, J. L. and GACHES, J. On the compatibility of a given solution with the data of a linear system. *J. Ass. comput. Mach.* **14**, 543–8 (1967).

21. SCHRÖDER, J. Das Iterationsverfahren bei allgemeinerem Abstandsbegriff. *Math. Z.* **66**, 111–16 (1956).

22. STIEFEL, E. *Einführung in die numerische Mathematik*, Chap. 1. Teubner, Stuttgart (1961).

23. WILKINSON, J. H. Error analysis of direct methods of matrix inversion. *J. Ass. comput. Mach.* **8**, 281–330 (1961).

24. —— *Rounding errors in algebraic processes*. H.M.S.O., London (1963).

25. —— *The algebraic eigenvalue problem*. Clarendon Press, Oxford (1965).

PART 2

CONTINUOUS PROBLEMS

6 · Introduction to Continuous Problems

In Chapter 1 we discussed, primarily, interval methods for problems whose solutions are numbers (or finite sets of numbers—vectors, matrices) or interval numbers. By 'continuous problems', we mean those whose solutions are continuous functions defined, say, on a whole interval of argument values. These are problems involving differential equations, integral equations, and so on. There is actually some overlap between these two areas. For example, in the discussion of 'algebraic problems' in Chapter 1, an interval polynomial $Q_k(x)$ was introduced which contains the function e^x for every $x \leqslant 0$. In this chapter we consider the problem of quadrature, whose solution is a number.

Interval methods have been developed for the machine computation of rigorous upper and lower bounds on exact solutions of 'continuous problems', including quadrature, integral equations, both initial and boundary-value problems for systems of non-linear ordinary differential equations and also for certain problems in partial differential equations. For an account of the work on partial differential equations see Chapter 9 by Krückeberg.

If $F(x)$ is an interval-valued function of a real variable and if F is also defined on intervals, say for $X \subset [a, b]$, then we can define

$$\int_a^b F(x)\,dx = \bigcap_{n=1}^{\infty} \sum_{i=1}^{n} F(X_i^{(n)})\left(\frac{b-a}{n}\right),$$

where

$$X_i^{(n)} = a + \left[\frac{i-1}{n}, \frac{i}{n}\right](b-a).$$

This amounts to the same thing mathematically as

$$\int_a^b F(x)\,dx = \left[\int_a^b F_1(x)\,dx, \int_a^b F_2(x)\,dx\right],$$

where $F(x) = [F_1(x), F_2(x)]$.

The first definition is of more interest computationally since, in the first place, it gives a means of computing an interval containing the exact value of the integral. In the second place, formulae for the end points $F_1(x)$ and $F_2(x)$ of an interval function are often not available or at least very inconvenient for computation.

To illustrate the use of the first formula, consider a special case when $F(x)$ is actually real-valued, for example, $F(x) = 1/x$. We obtain, for instance,

$$\int_1^2 \frac{dx}{x} \in \sum_{i=1}^{n} \frac{1}{1+[(i-1)/n, i/n]} \left(\frac{1}{n}\right)$$

for every positive integer n. Such a procedure, however, amounts to only a *first-order* method, that is the width of the bounding interval is of the order $1/n$.

Higher-order methods have been developed which are based on Gaussian quadrature formulae and also arbitrarily high-order methods can be based on Taylor series expansions. For example, see [1] and [2].

In order to use Taylor series expansions on the computer for this problem and for the solution of problems in ordinary differential equations, auxiliary techniques have been developed to enable the computer to derive recursion formulae for the *efficient* evaluation of successive Taylor coefficients. The evaluation can be carried out by the computer either in ordinary machine arithmetic or in rounded-interval arithmetic. In this way the remainder term in the Taylor series can be evaluated in interval arithmetic and thus bounded over a region containing the unknown 'mean value' that occurs in this form of Taylor series expansion.

A complete description of procedures that can be programmed for the computer for the machine computation of an approximate solution with rigorous bounds on the total error (all three kinds are taken into account) for the initial value problem for non-linear systems of ordinary differential equations can be found in [1].

Further work on interval methods for ordinary differential equations is described by Hansen in Chapter 7 and by Krückeberg in Chapter 9.

An interval version of Picard iteration can be used for the machine computation of interval functions that contain exact solutions to integral equations. In [1], such a procedure is described for interval 'step functions'. We conclude this introduction by giving two examples of the construction of nested sequences of interval polynomials for such problems.

The initial-value problem

$$y' = y^2, \qquad y(0) = 1$$

can be written as the integral equation

$$y(x) = 1 + \int_0^x y^2(x')\,dx'.$$

Let P_0 be a constant-interval polynomial with

$$P_0(x) \equiv [1, d] \qquad (x \geqslant 0),$$

and consider the sequence of interval polynomials defined by

$$P_{k+1}(x) = 1 + \int_0^x P_k^2(x')\,dx' \quad (k = 0, 1, 2,\ldots).$$

If $P_1(x) \subset P_0(x)$ for $0 \leqslant x \leqslant a$ $(a > 0)$, then the sequence $\{P_k(x)\}$ $(k = 0, 1, 2,\ldots)$ will be *nested*, that is $P_{k+1}(x) \subset P_k(x)$ for *every* k and all $x \in [0, a]$.

We wish to find numbers $d > 0$ and $a > 0$ such that

$$P_1(x) = 1 + \int_0^x [1, d]^2\,dx' = 1 + [1, d^2]x \subset [1, d]$$

for all $x \in [0, a]$. This amounts to the condition that

$$1 + d^2 a \leqslant d,$$

or

$$a \leqslant (d-1)/d^2,$$

which is satisfied, for example, by $d = 2$, $a = 1/4$.

With this choice $(d = 2, a = 1/4)$ we obtain

$$\begin{aligned} P_0(x) &= [1, 2], \\ P_1(x) &= 1 + [1, 4]x, \\ P_2(x) &= 1 + \int_0^x (1 + [1, 4]x')^2\,dx' \\ &= 1 + \int_0^x (1 + 2[1, 4]x' + [1, 16]x'^2)\,dx' \\ &= 1 + x + [1, 4]x^2 + [1/3, 16/3]x^3, \end{aligned}$$

and so on.

The sequence of interval polynomials $\{P_k(x)\}$ $(k = 0, 1, 2,\ldots)$ obtained in this way converges uniformly to the exact solution $y(x)$ of the given initial-value problem for $x \in [0, 1/4]$ (see Fig. 6.1). Furthermore, for every $k = 0, 1, 2,\ldots$ and every $x \in [0, 1/4]$ we have

$$y(x) \in P_k(x).$$

This sequence of interval polynomials has the disadvantage, however, of rapidly increasing degrees. The degree of $P_k(x)$ will be 2^k-1.

An idea of Krückeberg's can be used to limit the degrees of the $P_k(x)$ to any fixed degree desired—at the cost of some coarsening (*vergröberung*) of the bounds. In this example it would amount to the following. If $n > m$, then for $x \in [0, 1/4]$ we have

$$x^n = x^m . x^{n-m} \in [0, (1/4)^{n-m}]x^m.$$

Thus we can reduce any term of degree greater than m to a term of fixed

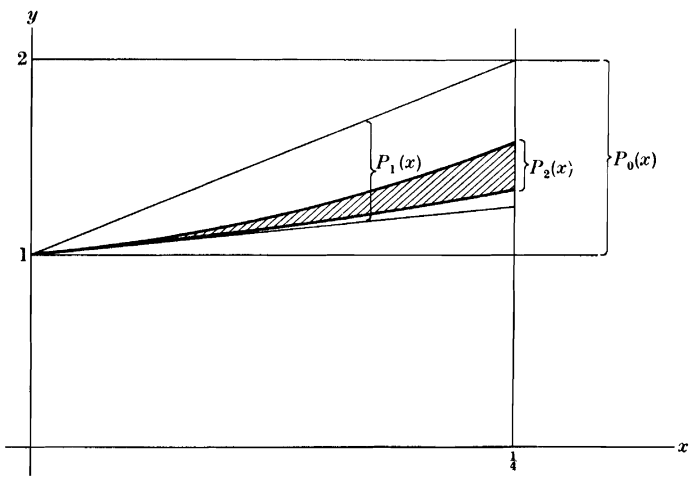


FIG. 6.1

degree m at the expense of a slight loss of sharpness. For instance, the last term in $P_2(x)$ was $[1/3, 16/3]x^3$ and if we choose $m = 2$, we can use the result

$$x^3 \in [0, 1/4]x^2 \qquad (x \in [0, 1/4])$$

to get $$[1/3, 16/3]x^3 \subset [1/3, 16/3][0, 1/4]x^2,$$

or $$[1/3, 16/3]x^3 \subset [0, 4/3]x^2 \qquad (x \in [0, 1/4]).$$

Therefore $$P_2(x) \subset 1+x+([1, 4]+[0, 4/3])x^2,$$

or $$P_2(x) \subset 1+x+[1, 16/3]x^2.$$

We could now use the second-degree polynomial

$$P_2^*(x) = 1+x+[1, 16/3]x^2$$

in place of $P_2(x)$ and compute terms of the sequence $\{P_k^*(x)\}$ obtained by 'cutting the degree' of

$$1+\int_0^x (P_k^*(x'))^2\, dx'$$

down to 2 each time.

This sequence will no longer converge to the exact solution $y(x)$, of course, but we will still have $y(x) \in P_k^*(x)$ for all k and $x \in [0, 1/4]$ and we could stop the iteration when no further improvement results.

For our second example, consider the boundary-value problem

$$y'' = e^{-y}, \qquad y(0) = y(1) = 0,$$

which can be written as the integral equation

$$y(x) = (x-1)\int_0^x x' e^{-y(x')}\,dx' + x\int_x^1 (x'-1)e^{-y(x')}\,dx'.$$

By arguments similar to those given in Chapter 1, if $0 \leqslant s \leqslant d$, then

$$e^s \in Q_k(s) = 1+s+\frac{s^2}{2!}+\cdots+\frac{s^{k-1}}{(k-1)!}+[1, e^d]\frac{s^k}{k!}$$

for every $k = 1, 2, \ldots$. The range of values of e^s when $s \in [s_1, s_2]$ with $0 \leqslant s_1 \leqslant s_2 \leqslant d$ is contained in $Q_k([s_1, s_2])$. Note that this is a slightly different Q_k than the earlier one, since here we bound e^ξ for $\xi \in [0, d]$ by $e^\xi \in [1, e^d]$.

Now define an *interval operator* G_k on interval-valued functions $Y(x)$ by

$$G_k(Y)(x) = (x-1)\int_0^x x' Q_k(-Y(x'))\,dx' + x\int_x^1 (x'-1)Q_k(-Y(x'))\,dx'.$$

If $Y(x) \subset [-d, 0]$ for $x \in [0, 1]$, then $-Y(x) \in [0, d]$ and

$$e^{-y(x)} \in Q_k(-Y(x))$$

providing $y(x) \in Y(x)$. Therefore if $y(x) \in Y(x)$, then $y(x) \in G_k(Y)(x)$ for $x \in [0, 1]$.

Furthermore, if we can find a number $d > 0$ such that

$$G_k([-d, 0])(x) \subset [-d, 0]$$

for all $x \in [0, 1]$, then the sequence given by

$$Y_0(x) \equiv [-d, 0],$$
$$Y_{p+1}(x) = G_k(Y_p)(x), \quad x \in [0, 1]$$

will be a nested sequence of interval polynomials containing an exact solution $y(x)$ of the boundary-value problem. If k is allowed to increase as the iteration proceeds, convergence to an exact solution will be obtained. In any case we will have $y(x) \in Y_p(x)$ for all p and $x \in [0, 1]$. Again we can limit the degrees of the polynomials $Y_p(x)$ by computing, as in the first example, $Y_p^*(x)$ with *vergröberung* to some fixed degree.

For a numerical illustration, take $k = 2$, then

$$e^s \in Q_2(s) = 1+s+[1, e^d]\tfrac{1}{2}s^2, \quad s \in [0, d].$$

In particular, $$e^d \in 1+d+[1, e^d]\tfrac{1}{2}d^2,$$

so that $$e^d \leqslant 1+d+e^d\tfrac{1}{2}d^2$$

and, therefore, if $0 < d < \sqrt{2}$ we can use

$$e^d \leqslant \frac{1+d}{1-\frac{1}{2}d^2}.$$

In any case, $Q_2([0, d]) = 1+[0, d]+\frac{1}{2}[1, e^d][0, d^2]$

and $$G_2([-d, 0])(x) = (x-1)\tfrac{1}{2}x^2Q_2([0, d])-\tfrac{1}{2}x(x-1)^2Q_2([0, d])$$
$$= \tfrac{1}{2}x(x-1)Q_2([0, d]).$$

Thus we wish to find a number $d > 0$ such that

$$\tfrac{1}{2}x(x-1)Q_2([0, d]) \subset [-d, 0]$$

for all $x \in [0, 1]$.

From Chapter 1, we have

$$Q_2([0, d]) \subset 1+[0, d]+\frac{1}{2}\left[1, \frac{1+d}{1-\frac{1}{2}d^2}\right][0, d^2]$$

(provided $d \in [0, \sqrt{2}]$).

Now the minimum value of $x(x-1)$ for $x \in [0, 1]$ is $-1/4$ so it is sufficient that d satisfy the inequality

$$-d \leqslant -\tfrac{1}{8}\left(1+d+\frac{1+d}{2-d^2}d^2\right)$$

provided $d \in [0, \sqrt{2}]$ also.

It can be verified directly that $d = 0{\cdot}15$ satisfies this condition. With this choice of d, we have, using

$$e^{0{\cdot}15} \leqslant \frac{1+0{\cdot}15}{1-\frac{1}{2}(0{\cdot}15)^2}$$

and computing with 3-digit rounded-interval arithmetic, the result that

$$Y_1(x) = G_2(Y_0(x)) = G_2([-0{\cdot}15, 0]) \subset x(x-1)[0{\cdot}5, 0{\cdot}582].$$

An exact solution of the boundary-value problem

$$y'' = e^{-y}, \qquad y(0) = y(1) = 0$$

thus lies in $x(x-1)[0{\cdot}5, 0{\cdot}582]$ for all $x \in [0, 1]$ (see Fig. 6.2).

This two-point boundary problem has two exact solutions. One is in the region indicated. The other, which lies outside this region except near the end-points, can also be found by the method described here by starting with an initial region for $Y_0(x)$ which contains it and is sufficiently narrow.

Further iteration with higher values of k will produce narrower bounding interval polynomials.

See Chapter 8 by Hansen for more discussion of interval methods for such problems.

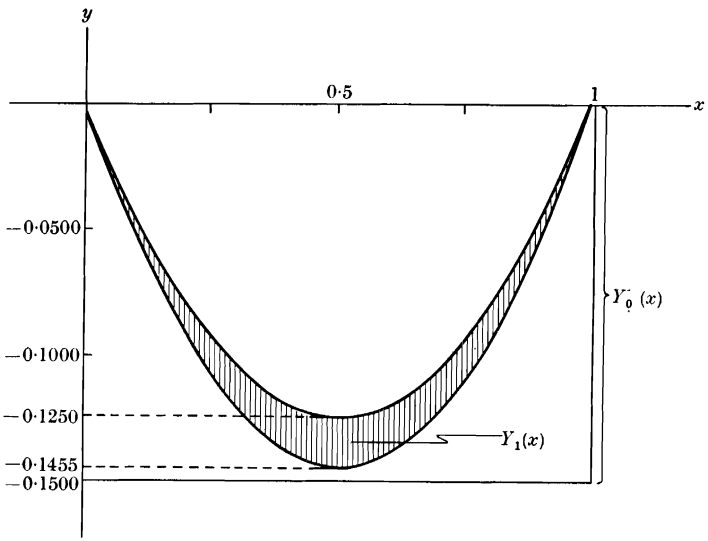


FIG. 6.2

REFERENCES

1. MOORE, R. E. *Interval analysis.* Prentice-Hall, New Jersey (1966).

2. —— *Practical aspects of interval computation.* Aplikace matematiky, Prague **13**, 52–92 (1968).

7 · On Solving Two-point Boundary-value Problems using Interval Arithmetic

1. Introduction

In this chapter we show how interval arithmetic can be used to bound the solution to certain two-point boundary-value problems for ordinary differential equations. Our method can be applied to non-linear equations but in some such cases we assume initial crude bounds to be given. However, for the linear case, no initial bounds are required.

To simplify the presentation, we consider only a single equation of second order. The method to be discussed can be easily extended to the more general case. In [6] (p. 83), Moore indicates that the problem

$$y'' = f(x, y), \qquad y(a) = y_0, \qquad y(b) = y_1$$

can be solved, with strict bounds on the error, by formulating the problem as an integral equation and using the method of Chapter 9 of [6]. See also Chapter 6 of this book. In the following, we present an alternative procedure for solving the more general equation

$$y'' = f(x, y, y') \tag{1.1}$$

with boundary conditions

$$g_1[a, y(a), y'(a)] = 0, \qquad g_2[b, y(b), y'(b)] = 0. \tag{1.2}$$

For brevity, we replace equations (1.2) by the special simple conditions

$$y(a) = y_0, \qquad y(b) = y_n. \tag{1.3}$$

However, use of (1.2) instead of (1.3) introduces no difficulties. Our method is essentially an interval analytic extension of the difference approximation method in common use (see, for example, [2]). The modification of our method to use (1.2) instead of (1.3) follows the same steps as those of section 12 of Chapter 4 of [2].

We assume the problem expressed by (1.1) and (1.2) has a unique solution, bounded in $[a,b]$, and that its first four derivatives are continuous and bounded in $[a,b]$. We impose further conditions at the end of section 3.

We also assume f is a rational function of x, y, and y'. If this is not the case, it may be possible to obtain a system of differential equations entailing only rational functions. For details of such a step, see, for example, section 11.2 of [6]. This assumption is not necessary, in general, since there are means for computing intervals containing the 'value' of irrational functions with interval arguments. For example, see Chapter 1 by Moore. We consider some irrational examples in sections 10 and 11.

2. The basic step of the method

Divide the interval $[a,b]$ into sub-intervals

$$X_i = [x_i, x_{i+1}] \quad (i = 0, 1, \ldots, n-1)$$

where $x_0 = a$ and $x_n = b$. The meshpoints x_i need not be equally spaced although we assume them to be. At each interior meshpoint $x_1, \ldots, x_{n-1}$, we write discrete approximations for the derivatives in the differential equation (1.1). The error in these approximations can be analytically expressed and then bounded by use of interval arithmetic. The simplest central difference approximations are the well-known formulae

$$y'_i = \frac{1}{2h}(y_{i+1} - y_{i-1}) - \frac{h^2}{6} y'''(\xi_i) \quad (i = 1, \ldots, n-1) \tag{2.1}$$

and

$$y''_i = \frac{1}{h^2}(y_{i+1} - 2y_i - y_{i-1}) - \frac{h^2}{12} y^{(4)}(\eta_i) \quad (i = 1, \ldots, n-1), \tag{2.2}$$

where $h = x_{i+1} - x_i$ and y_i denotes $y(x_i)$, etc. The quantities ξ_i and η_i are unknown except that $\xi_i \in X_i^*$ and $\eta_i \in X_i^*$ where

$$X_i^* = X_i \cup X_{i-1} = [x_{i-1}, x_{i+1}].$$

We later show how to bound the error terms. For now, assume we know intervals A_i and B_i such that $y'''(\xi_i) \in A_i$ and $y^{(4)}(\eta_i) \in B_i$ for ξ_i and η_i in X_i^*. Substituting these bounding intervals for the respective quantities in (2.1) and (2.2) and substituting the results into (1.1) (with $x = x_i$), we obtain

$$\frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}) - \frac{h^2}{12} B_i = f\left[x_i, y_i, \frac{1}{2h}(y_{i+1} - y_{i-1}) - \frac{h^2}{6} A_i\right] \quad (i = 1, \ldots, n-1). \tag{2.3}$$

Since y_0 and y_n are given, we thus have $n-1$ equations in the $n-1$ unknowns $y_1,\ldots, y_{n-1}$. If the equations are linear, we can solve them by (say) the interval arithmetic method recommended in [5] (for which a computer program is given in [7]). If the equations are non-linear, a method in [4] or [6] can be used. We thus obtain an interval y_i^I containing y_i $(i = 1,\ldots, n-1)$.

3. Obtaining A_i and B_i

We now consider how the bounds A_i and B_i can be obtained. To do this, we assume we have bounds on y and y'; that is, assume we know $y(x) \in Y_i$ and $y'(x) \in Y_i'$ for $x \in X_i$. In later sections we discuss how to find the intervals Y_i and Y_i'.

Differentiating equation (1.1), we have

$$y''' = f_x + y'f_y + y''f_{y'}, \tag{3.1}$$

where $f_y \equiv \partial f/\partial y$ and $f_{y'} \equiv \partial f/\partial y'$. Substituting for y'' in (3.1) from (1.1), we obtain

$$y''' = p(x, y, y'), \tag{3.2}$$

where

$$p(x, y, y') \equiv f_x + y'f_y + ff_{y'} \tag{3.3}$$

is a function of x, y, and y' alone.

Similarly we find

$$y^{(4)} = q(x, y, y') \tag{3.4}$$

by differentiating (3.2) and substituting for y'' as before.

We can bound y''' over an interval X_i by evaluating $p(X_i, Y_i, Y_i')$ using interval arithmetic. Denote

$$A_i = p(X_i, Y_i, Y_i') \cup p(X_{i-1}, Y_{i-1}, Y_{i-1}'). \tag{3.5}$$

Similarly,

$$B_i = q(X_i, Y_i, Y_i') \cup q(X_{i-1}, Y_{i-1}, Y_{i-1}'). \tag{3.6}$$

The intervals A_i and B_i are the quantities required in (2.3).

We assume that $p(X_i, Y_i, Y_i')$ and $q(X_i, Y_i, Y_i')$ are bounded for all $i = 0,\ldots, n-1$. This rules out many interesting differential equations. For example, we cannot solve $y'' = y/x$ if $0 \in [a, b]$.

4. Improving Y_i'

We assume the bounding intervals Y_i and Y_i' were initially crude. We now consider how to improve these bounds.

Using Taylor series with remainder we easily find

$$y'(x) = \frac{1}{h}(y_{i+1} - y_i) + \frac{1}{2h}[(x_{i+1} - x)^2 y''(\theta_i) - (x - x_i)^2 y''(\phi_i)] \tag{4.1}$$

for any point $x \in X_i$, where $\theta_i \in X_i$ and $\phi_i \in X_i$. Denote

$$Y_i'' = f(X_i, Y_i, Y_i'). \tag{4.2}$$

From (1.1), $y''(\theta_i) \in Y_i''$ and $y''(\phi_i) \in Y_i''$ and hence from (4.1),

$$\begin{aligned} y'(x) &\in \frac{1}{h}(y_{i+1}^I - y_i^I) + \frac{1}{2h}\{(x_{i+1}-X_i)^2 Y_i'' - (X_i - x_i)^2 Y_i''\} \\ &= \frac{1}{h}(y_{i+1}^I - y_i^I) + \frac{h}{2}\{[0,1]^2 Y_i'' - [0,1]^2 Y_i''\} \\ &= \frac{1}{h}(y_{i+1}^I - y_i^I) + \frac{h}{2} w\{[0,1]Y_i''\}[-1,1] \end{aligned}$$

for any $x \in X_i$, where $w\{[0,1]Y_i''\}$ denotes the width of the interval $[0,1]Y_i''$. Denote

$$Y_i' = \frac{1}{h}(y_{i+1}^I - y_i^I) + \frac{h}{2} w([0,1]Y_i'')[-1,1]. \tag{4.3}$$

Then $y'(x) \in Y_i'$ for any $x \in X_i$ and Y_i' is the (improved, in general) bound we sought. We use the same notation Y_i' for the old crude bound and the new improved bound on $y'(x)$ for $x \in X_i$. At any stage of our method, Y_i' denotes the current best approximation. In practice we should use the intersection of the old and new intervals.

5. Improving Y_i

We now use the improved bound Y_i' to improve Y_i. Using Taylor series, we easily find

$$y(x) = \tfrac{1}{2}[y_i + y_{i+1} + (x_{i+1} - x)y'(\mu_i) - (x - x_i)y'(\nu_i)] \tag{5.1}$$

for any $x \in X_i$, where $\mu_i \in X_i$ and $\nu_i \in X_i$. Since $y'(\mu_i) \in Y_i'$ and $y'(\nu_i) \in Y_i'$, we have

$$\begin{aligned} y(x) &\in \tfrac{1}{2}[y_i^I + y_{i+1}^I + (x_{i+1} - X_i)Y_i' - (X_i - x_i)Y_i'] \\ &= \tfrac{1}{2}(y_i^I + y_{i+1}^I) + \frac{h}{2}([0,1]Y_i' - [0,1]Y_i') \end{aligned}$$

for any $x \in X_i$. Denote

$$Y_i = \tfrac{1}{2}(y_i^I + y_{i+1}^I) + \frac{h}{2} w([0,1]Y_i')[-1,1]. \tag{5.2}$$

Then $y(x) \in Y_i$ for $x \in X_i$. This new value of Y_i replaces the original crude value; and, as before for Y_i', the intersection of the two can be used.

6. The iterative method

We are now able to describe the method we propose. We proceed in the following steps.

(*a*) Procure crude bounds Y_i and Y_i' for $i = 0, \ldots, n-1$ (see sections 8–11).

(*b*) Evaluate A_i using (3.5) and B_i using (3.6) for $i = 1,..., n-1$.
(*c*) Using (1.3), solve equations (2.3) for y_i^I ($i = 1,..., n-1$).
(*d*) Find improved bounds Y_i' ($i = 0,..., n-1$) using (4.3).
(*e*) Find improved bounds Y_i ($i = 0,..., n-1$) using (5.2).
(*f*) Iterate steps (*b*)–(*e*).

The iteration can proceed either until the error bounds are sufficiently sharp or until successive iterates differ by a sufficiently small amount. Note that for fixed finite precision arithmetic, a stage will be reached where no improvement occurs.

We have assumed convergence. If this does not occur, the fact is almost immediately revealed. In theory, the likelihood of convergence is enhanced by reducing h. In practice, this may not help because the number of interval equations (2.3) increases and may be difficult to solve sharply.

We wish next to present a general procedure for obtaining crude bounds when the differential equation is linear. To do this, we first require some preliminary concepts which we now consider.

7. Computation with variable intervals

Let M_r and N_r be intervals whose end-points are specifically given numbers. Let W be an unspecified variable interval. We cannot unambiguously express $M_r W$ explicitly in terms of the end-points of M_r and W since the end-points of $M_r W$ depend upon the unknown signs and magnitudes of W. To compute with variable intervals, we can represent them as $N_r W$. Then $M_r(N_r W)$ can be 'computed' by evaluating $N_{r+1} = M_r N_r$ and representing the result in the form $N_{r+1} W$.

If we assume W is symmetric about the origin so that $W = [-w, w]$, we can simplify the arithmetic. We can then replace $N_r = [n_r^L, n_r^R]$ by a positive real number n_r since

$$N_r[-w, w] = n_r[-w, w],$$

where $n_r = |N_r| = \max(|n_r^L|, |n_r^R|)$. Similarly

$$W/M_r = W|1/M_r|. \tag{7.1}$$

In the next section, we take advantage of this simplification.

8. Obtaining Y_i and Y_i' in the linear case

If boundary conditions of the form (3) are given, we choose to seek bounds of the form

$$Y_i = \{y(a)+y(b)\}/2+U \tag{8.1}$$

and

$$Y_i' = \frac{y(b)-y(a)}{b-a} + V, \tag{8.2}$$

where $U = [-u, u]$ and $V = [-v, v]$ so that U and V are symmetric about the origin. If the boundary conditions are not of the form (3), we can simply choose $Y_i = U$ and $Y'_i = V$.

Using the analytically expressed bounds (8.1) and (8.2), we perform steps (*b*)–(*e*) of our method described in section 6. We use the arithmetic described in section 7.

Let $\overline{Y}_i$ and $\overline{Y}'_i$ denote the new bounds on the solution and its derivative, obtained in this way. We find

$$\overline{Y}_i = M_i + c_i U + d_i V \tag{8.3}$$

and

$$\overline{Y}'_i = M'_i + c'_i U + d'_i V \tag{8.4}$$

for $i = 0,\ldots, n-1$, where $c_i \geqslant 0$, $d_i \geqslant 0$, $c'_i \geqslant 0$, and $d'_i \geqslant 0$. Denote $M_i = [p_i, q_i]$, $M'_i = [p'_i, q'_i]$, $\overline{Y}_i = [\bar{y}_i^L, \bar{y}_i^R]$, and $\overline{Y}'_i = [\bar{z}_i^L, \bar{z}_i^R]$. From (8.3) and (8.4),

$$\begin{aligned} \bar{y}_i^L &= p_i - c_i u - d_i v, \\ \bar{y}_i^R &= q_i + c_i u + d_i v, \\ \bar{z}_i^L &= p'_i - c'_i u - d'_i v, \\ \bar{z}_i^R &= q'_i + c'_i u + d'_i v, \end{aligned} \tag{8.5}$$

for $i = 0,\ldots, n-1$.

Let $r = \{y(a)+y(b)\}/2$ and $s = \{y(b)-y(a)\}/(b-a)$. Then from (8.1) and (8.5), the differences between the old and new left end-points of the bounds on $y(x)$ in X_i are

$$\begin{aligned} \Delta y_i^L &= (p_i - c_i u - d_i v) - (r - u) \\ &= p_i - r + (1 - c_i)u - d_i v \quad (i = 0,\ldots, n-1) \end{aligned} \tag{8.6}$$

and the changes in the right end-points are

$$\begin{aligned} \Delta y_i^R &= (q_i + c_i u + d_i v) - (r + u) \\ &= q_i - r - (1 - c_i)u + d_i v \quad (i = 0,\ldots, n-1). \end{aligned} \tag{8.7}$$

Similarly, the changes in the end-points of the interval containing $y'(x)$ in X_i are

$$\Delta z_i^L = p'_i - s - c'_i u + (1 - d'_i)v \tag{8.8}$$

and

$$\Delta z_i^R = q'_i - s + c'_i u - (1 - d'_i)v \tag{8.9}$$

for $i = 0,\ldots, n-1$.

The new bounds are strict improvements over the old if

$$\Delta y_i^L > 0, \quad \Delta y_i^R < 0, \quad \Delta z_i^L > 0, \quad \Delta z_i^R < 0 \quad (i = 0,\ldots, n-1). \tag{8.10}$$

We now argue that for h sufficiently small, the inequalities (8.10) will, in fact, hold provided the initial bounds are not already too sharp.

We performed steps (*b*)–(*e*) of the procedure in section 6. Each of these steps involved use of a formula in which the error term was multiplied by a positive integer power of h. Hence the non-negative numbers c_i, d_i, c_i', and d_i' are all $O(h)$. We assume h is so small that

$$c_i < 1, \qquad d_i' < 1 \quad (i = 1,\ldots, n-1). \tag{8.11}$$

From (8.11), we see that the coefficients $1-c_i$ in (8.6) and (8.7) and $1-d_i'$ in (8.8) and (8.9) are positive. This (along with another condition given later) enables us to satisfy conditions (8.10) by choosing u and v to be related in an appropriate way.

Since we seek only crude bounds, we shall not attempt to obtain a best result but shall sacrifice sharpness (in the crude bounds) for simplicity of method. We do not, of course, drop the requirement that our bounds be strict.

Using (8.6)–(8.9) and noting (8.11), we rewrite (8.10) as

$$0 < (p_i-r)/d_i+u(1-c_i)/d_i-v, \tag{8.12}$$

$$0 < (r-q_i)/d_i+u(1-c_i)/d_i-v, \tag{8.13}$$

$$0 < (p_i'-s)/(1-d_i')-uc_i'/(1-d_i')+v, \tag{8.14}$$

$$0 < (s-q_i')/(1-d_i')-uc_i'/(1-d_i')+v, \tag{8.15}$$

for $i = 0,\ldots, n-1$. We assume $d_i > 0$ although the case $d_i = 0$ causes no difficulty. However, $d_i = 0$ only if f is independent of y. For simplicity, we omit discussion of such cases. Define

$$\alpha_1 = \min\{(p_i-r)/d_i\}, \quad \alpha_2 = \min\{(r-q_i)/d_i\},$$
$$\alpha_1' = \min\{(p_i'-s)/(1-d_i')\}, \quad \alpha_2' = \min\{(s-q_i')/(1-d_i')\},$$
$$\beta = \min\{(1-c_i)/d_i\}, \quad \text{and} \quad \beta' = \max\{c_i'/(1-d_i')\},$$

where the max and min are taken over all $i = 0,\ldots, n-1$. Define $\alpha = \min(\alpha_1, \alpha_2)$ and $\alpha' = \min(\alpha_1', \alpha_2')$. Then (8.12) and (8.13) are satisfied if $u \geqslant 0$ and $v \geqslant 0$ are such that

$$0 < \alpha+\beta u-v \tag{8.16}$$

and (8.14) and (8.15) are satisfied if

$$0 < \alpha'-\beta' u+v. \tag{8.17}$$

The right members of (8.16) and (8.17) can be viewed as lines in the (u, v) plane. Relation (8.16) says the point (u, v) must lie below the line

$$\alpha+\beta u-v = 0 \tag{8.18}$$

and (8.17) says (u, v) must lie above the line

$$\alpha' - \beta' u + v = 0. \tag{8.19}$$

It is easily seen that $0 < \beta = O(1/h)$ and $0 < \beta' = O(1)$. Hence for h sufficiently small,

$$\beta' < \beta \tag{8.20}$$

and there exist points (u, v) satisfying both (8.16) and (8.17). Hereafter, we assume (8.20) to hold.

In general, conditions (8.11) and (8.20) can be satisfied by choosing h sufficiently small. However, this is not always the case. If equations (2.3) are nearly linearly dependent, y_i^I may be large in magnitude. Subsequently, terms which are supposedly relatively small may not be so and our method can fail. Note that for the eigenvalue problem wherein equations (2.3) are, in fact, linearly dependent, our method fails completely.

We choose (u, v) to lie on the line whose slope is the average of the slopes of the lines (8.18) and (8.19) and which passes through their point of intersection. That is, we choose (u, v) to lie on the line

$$\alpha - \alpha' + (\beta + \beta')u - 2v = 0. \tag{8.21}$$

A point on this line satisfies (8.16) and (8.17) if $u = u_0 + \Delta u$ for all $\Delta u > 0$ where

$$u_0 = -(\alpha + \alpha')/(\beta - \beta'). \tag{8.22}$$

It is easily seen that $u_0 = O(1)$. Note we also require $u > 0$.

Substituting for v from (8.21) into (8.6) and (8.7) and using the definitions of α, α', β, and β', we find

$$\Delta y_i^L \geqslant \frac{d_i}{2}(\beta - \beta')(u - u_0) \tag{8.23}$$

and

$$\Delta y_i^R \leqslant -\frac{d_i}{2}(\beta - \beta')(u - u_0) \tag{8.24}$$

respectively. Whatever value of $u > u_0$ we use in (8.1), we now see that we could replace it by

$$u^* = u - \frac{d}{2}(\beta - \beta')(u - u_0),$$

where $d = \min(d_i)$ for $i = 0, \ldots, n-1$. That is, we could reduce $u - u_0$ by

$$\Delta(u - u_0) = \left\{1 - \frac{d}{2}(\beta - \beta')\right\}(u - u_0). \tag{8.25}$$

Now $1 - (\beta - \beta')d/2 \geqslant (1 + c_i + d\beta')/2 > 0$ and hence $u - u_0$ can always be reduced by a positive fixed fraction of itself if $u > u_0$.

Similarly, from (8.8) and (8.9) we find that $v-v_0$ can be reduced by an amount

$$\Delta(v-v_0) = \frac{\beta-\beta'}{\beta+\beta'}(1-d')(v-v_0), \tag{8.26}$$

where $d' = \max(d_i')$ for $i = 0,\ldots, n-1$ and

$$v_0 = -\frac{\alpha\beta'+\beta\alpha'}{\beta-\beta'}.$$

Hence both $u-u_0$ and $v-v_0$ can be reduced by fixed positive fractions of themselves if $u > u_0$ and $v > v_0$. We have assumed y and y' bounded for all $x \in [a, b]$. Suppose we choose u and v satisfying (8.21) and so large that (8.1) and (8.2) do, in fact, bound y and y' in $[a, b]$. If

$$u > u_1 = \max(0, u_0) \quad \text{and} \quad v > v_1 = \max(0, v_0),$$

we can reduce both u and v, keeping (8.21) satisfied, until as the limit of an infinite sequence of steps, $u = u_1$ and $v = v_1$. (Actually we will find $u \leqslant u_1$ and $v \leqslant v_1$ since we have used bounds rather than true values of Δy_i^L, etc.) Thus, letting $u = u_1$ in (8.1) and $v = v_1$ in (8.2) yields actual bounds on y and y' for $x \in [a, b]$.

It is quite easy to obtain results which are slightly sharper, in general. Replace $<$ by $\leqslant$ in (8.12)–(8.15) and substitute for v in terms of u from (8.21). Find the smallest value of u satisfying all these relations for all $i = 0,\ldots, n-1$. This value, substituted into (8.1), yields bounds on y. Substituting this value of u into (8.21) and solving for v yields a value which when used in (8.2) yields bounds on y'.

In general, still better results can be obtained by solving a linear programming problem. We minimize $\phi(u, v) = u$ subject to the constraints (8.10). The values of u and v for the solution point (u, v) yield bounds as before.

Proof of the validity of the statements in the last two paragraphs can be obtained in the manner used to prove $u = u_1$ and $v = v_1$ provide actual bounds.

In practice, a relatively large value of h could be used in obtaining the crude error bounds. If the bounding procedure cannot be completed because (8.11) or (8.20) fails to hold, then h can be reduced and the process repeated. However, the crude bounding procedure is quite simple to apply. Moreover, it yields sharper results for moderately small h. Hence there is no great advantage in using large h.

9. Example

We illustrate the above analysis with an example considered by Collatz on pp. 178 and 179 of [1]. Consider

$$y'' = 2x^{-2}y - 1/x \tag{9.1}$$

with boundary conditions

$$y(2) = y(3) = 0. \tag{9.2}$$

Following Collatz, we let $h = 1/3$. In general practice, however, it is necessary to choose a machine representable value of h.

Differentiating (9.1), we find

$$y^{(4)} = 4x^{-3}(4y/x - 1 - 2y') = q(x, y, y'). \tag{9.3}$$

We shall not need y''' since y' does not occur in (9.1) so we shall not have to use (2.1). We shall be evaluating (9.3) with the variables replaced by intervals. Hence we ought to write the equation in such a way as to obtain sharpest results. The given form is better, for example, than $4x^{-4}\{4y - x(1+2y')\}$.

Equations (8.1), (8.2), and (9.2) dictate that we seek bounds of the form $Y_i = U$ and $Y_i' = V$. We thus replace y by U and y' by V in (9.3) and replace x by suitable intervals. To reduce the labour, we have used intervals X_i^* to find B_i rather than use X_i and X_{i-1} separately. In general practice this should not be done since

$$q(X_{i-1}, U, V) \cup q(X_i, U, V) \subset q(X_i^*, U, V);$$

that is, the left-hand member of this relation yields sharper results, usually. We find

$$B_1 = q(X_1^*, U, V) = [-0{\cdot}500, -0{\cdot}210] + U + V,$$

$$B_2 = q(X_2^*, U, V) = [-0{\cdot}315, -0{\cdot}148] + 0{\cdot}541U + 0{\cdot}630V.$$

Substituting these results into (2.3) and using (9.2), we obtain

$$-900y_1 + 441y_2 = [-21{\cdot}3, -21{\cdot}0] + 0{\cdot}454U + 0{\cdot}454V,$$

$$288y_1 - 585y_2 = [-12{\cdot}1, -12{\cdot}0] + 0{\cdot}161U + 0{\cdot}187V.$$

Solving these equations, we get

$$y_1^I = [0{\cdot}0439, 0{\cdot}0449] + 0{\cdot}000843U + 0{\cdot}000875V,$$

$$y_2^I = [0{\cdot}0419, 0{\cdot}0429] + 0{\cdot}000690U + 0{\cdot}000753V.$$

We next find $\quad Y_0'' = [-0{\cdot}500, -0{\cdot}428] + 0{\cdot}500U$

using (4.2), and

$$Y_0' = [0{\cdot}0483, 0{\cdot}219] + 0{\cdot}170U + 0{\cdot}00263V$$

using (4.3). Similarly, we find Y_1' and Y_2'. Next we obtain

$$Y_0 = [-0{\cdot}0148, 0{\cdot}0680] + 0{\cdot}0571U + 0{\cdot}00132V$$

as well as Y_1 and Y_2 using (5.2).

Writing (8.6)–(8.9) for $i = 0$, 1, and 2, we have

$$\begin{aligned}
\Delta y_0^L &= -0{\cdot}0148 + 0{\cdot}9429u - 0{\cdot}00132v,\\
\Delta y_1^L &= 0{\cdot}018 + 0{\cdot}9564u - 0{\cdot}00245v,\\
\Delta y_2^L &= -0{\cdot}0113 + 0{\cdot}9676u - 0{\cdot}00113v,\\
\Delta y_0^R &= 0{\cdot}068 + 0{\cdot}9429u - 0{\cdot}00132v,\\
\Delta y_1^R &= -0{\cdot}0688 + 0{\cdot}9564u - 0{\cdot}00245v,\\
\Delta y_2^R &= -0{\cdot}0537 + 0{\cdot}9676u - 0{\cdot}0113v,\\
\Delta z_0^L &= 0{\cdot}0483 - 0{\cdot}17u + 0{\cdot}99737v\\
\Delta z_1^L &= -0{\cdot}0805 - 0{\cdot}128u + 0{\cdot}99511v,\\
\Delta z_2^L &= -0{\cdot}192 - 0{\cdot}096u + 0{\cdot}99774v,\\
\Delta z_0^R &= -0{\cdot}219 - 0{\cdot}17u + 0{\cdot}99737v,\\
\Delta z_1^R &= -0{\cdot}0685 - 0{\cdot}128u + 0{\cdot}99511v,\\
\Delta z_2^R &= 0{\cdot}0635 - 0{\cdot}096u + 0{\cdot}99774v.
\end{aligned} \tag{9.4}$$

Thus (8.18) and (8.19) become (approximately)

$$0 = -51{\cdot}5 + 390u - v$$

and

$$0 = -0{\cdot}22 - 0{\cdot}17u + v,$$

respectively. Rounding to one significant digit (for convenient hand calculation) we approximate (8.21) by

$$v = 200u - 30. \tag{9.5}$$

It does not matter that we approximate (8.21) so poorly since we choose not to compute u_1 and v_1. Instead we use the alternative method described above.

Substituting for v from (9.5) into (9.4) we find $\Delta y_i^L \geqslant 0$, $\Delta y_i^R \leqslant 0$, $\Delta z_i^L \geqslant 0$, and $\Delta z_i^R \leqslant 0$ for $i = 0$, 1, and 2 if $u = 0{\cdot}152$. If we had used (8.22), we would have found $u_0 = 0{\cdot}133$. The alternative method (which is better) has yielded a worse result because we rounded (8.21) so drastically to get (9.5). Using higher-precision arithmetic to obtain (9.5), we could have got $u_0 = 0{\cdot}133$. Using $u = 0{\cdot}152$, equation (8.1) reveals that $y \in [-0{\cdot}152, 0{\cdot}152]$ for all $x \in [2, 3]$. Solving (9.5) for v and using (8.2), we find that $y' \in [-0{\cdot}4, 0{\cdot}4]$.

We could now use our iterative process beginning with step (*b*) in section 6. However, we already know what the result of performing

steps (*b*)–(*e*) will be. Except for the fact that the arithmetic might differ slightly, we would obtain the improvements given by (9.4) for the bounds. We thus find $Y_0 = [-0{\cdot}023, 0{\cdot}0773], Y_1 = [0{\cdot}01, 0{\cdot}765]$, $Y_2 = [-0{\cdot}017, 0{\cdot}0591]$, $Y'_0 = [0{\cdot}021, 0{\cdot}246]$, $Y'_1 = [-0{\cdot}102, 0{\cdot}09]$, and $Y'_2 = [-0{\cdot}208, -0{\cdot}048]$.

We now begin our iterative process. We find $y_1^I = [0{\cdot}0434, 0{\cdot}0447]$ and $y_2^I = [0{\cdot}0418, 0{\cdot}0427]$. As before we have evaluated B_1 and B_2 in the form $B_i = q(X_i^*, Y_{i-1} \cup Y_i, Y'_{i-1} \cup Y'_i)$. To improve sharpness in the final time through the iterative process, we use

$$B_i = q(X_{i-1}, Y_{i-1}, Y'_{i-1}) \cup q(X_i, Y_i, Y'_i).$$

We find $y_1^I = [0{\cdot}0440, 0{\cdot}0446]$ and $y_2^I = [0{\cdot}0422, 0{\cdot}0427]$. Very little improvement could be obtained by further iteration.

Our step-size h is too large to yield high accuracy. However, if we consider the mid-points of y_1^I and y_2^I to be approximate values of y_1 and y_2, we know that the relative errors are less than 0·007 and 0·006, respectively.

Collatz [1] solved this same problem approximately and using an explicit expression for $y^{(4)}$ in terms of x obtained estimates

$$y_1^I = [0{\cdot}043288, 0{\cdot}044708] \quad \text{and} \quad y_2^I = [0{\cdot}041464, 0{\cdot}042884].$$

In practice, of course, we do not know $y^{(4)}$ explicitly in terms of x alone. Without this information, we have obtained error bounds, not estimates, which are sharper.

10. Crude bounds for non-linear equations

We now consider ways in which crude bounds on y and y' can be obtained for non-linear differential equations of the form (1.1) with boundary conditions given by (1.3).

It should be noted that no initial bounds on y' are required if $f(x, y, y')$ in (1.1) is independent of y'. In this case (4.3) provides bounds on y' assuming bounds on y are known.

In very special situations initial bounds may be quite simple to obtain. Suppose, for example, the differential equation is

$$y''(x) = \frac{g(x)}{1+\{y(x)\}^2+\{y'(x)\}^2}.$$

Suppose $g(x)$ is bounded for $x \in [a, b]$. Evaluating $g([a, b])$ in interval arithmetic, let G be the interval obtained. Then $y''(x) \in G$ for all $x \in [a, b]$. Hence (4.3) provides bounds Y'_i $(i = 0, \ldots, n-1)$. We merely let $h = b-a$

so that $y_i = y(a)$ and $y_{i+1} = y(b)$. These quantities are given by (1.3). Similarly, (5.2) yields a bound on $y(x)$ for all $x \in [a, b]$.

In problems for which it is applicable, a crude form of the method discussed by Moore in Chapter 6 could be used to get initial bounds on y.

We now quote a theorem due to Gendzhoian [3] which can be useful:

THEOREM. *Given* $y'' = f(x, y, y')$, $y(0) = y(1) = 0$. *Assume that for* $0 \leqslant x \leqslant 1$ *and* $y^2 + y'^2 < \infty$, *the following conditions hold*:

(i) f *is continuous in* x, y, *and* y'.

(ii) f *is continuously differentiable with respect to* y *and* y'.

(iii) $0 \leqslant f_y \leqslant M$ *and* $|f_{y'}| < M$.

Let $N \geqslant 0$ *be such that* $|f(x, 0, 0)| < 2e^{N/2}$ *and let* $\alpha = \frac{1}{2}\{M + (M^2+4)^{1/2}\}$ *and* $R = \max(N, \alpha)$. *Then* $-v(x) \leqslant y(x) \leqslant v(x)$ *for* $0 \leqslant x \leqslant 1$ *where*

$$v(x) = 1 + e^R - e^{Rx} - e^{R(1-x)}.$$

In certain cases, it can be determined that the conditions of this theorem hold. Note that an upper bound for the constant N can be obtained by evaluating $f(x, 0, 0)$ in interval arithmetic with x replaced by the interval $[0, 1]$.

If for all $x \in [a, b]$, the function $f(x, y, y')$ does not grow too rapidly as a function of y and y', we can obtain crude bounds on y and y' in a way similar to that of the last section.

Denote $Y = [y^L, y^R]$, $Y' = [z^L, z^R]$. If we substitute the fixed interval $X = [a, b]$ for x and the variable intervals Y for y and Y' for y' in $f(x, y, y')$, we have

$$f(X, Y, Y') = [g^L, g^R], \tag{10.1}$$

where g^L and g^R are functions of y^L, y^R, z^L, and z^R.

From (4.3), $y' \in \overline{Y}'$ for $x \in X$ where

$$\overline{Y}' = \frac{y(b) - y(a)}{b - a} + \frac{b - a}{2} w\{[0, 1][g^L, g^R]\}[-1, 1]. \tag{10.2}$$

Thus from (5.2), $y \in \overline{Y}$ for $x \in X$ where

$$\begin{aligned} \overline{Y} &= \tfrac{1}{2}\{y(a) + y(b)\} + \frac{b - a}{2} w([0, 1]\overline{Y}')[-1, 1] \\ &= \tfrac{1}{2}\{y(a) + y(b)\} + \tfrac{1}{2}\{y(b) - y(a) + (b - a)^2 w([0, 1][g^L, g^R])\}[-1, 1]. \end{aligned} \tag{10.3}$$

Denote $\overline{Y} = [\bar{y}^L, \bar{y}^R]$, $\overline{Y}' = [\bar{y}^L, \bar{y}^R]$, $\Delta y^L = \bar{y}^L - y^L$, $\Delta y^R = \bar{y}^R - y^R$, $\Delta z^L = \bar{z}^L - z^L$, and $\Delta z^R = \bar{z}^R - z^R$.

The bounds (10.2) and (10.3) are strictly better than the bounds Y and Y', respectively, if

$$\Delta y^L > 0, \qquad \Delta y^R < 0, \qquad \Delta z^L > 0, \qquad \Delta z^R < 0. \tag{10.4}$$

This will be the case if g^L and g^R grow at less than a linear rate as functions of their arguments provided that $-y^L$, y^R, $-z^L$, and z^R are sufficiently large positive numbers. In particular cases, linear growth of g^L and g^R may be acceptable.

For a linear differential equation, $\bar{y}^L, \bar{y}^R, \bar{z}^L$, and $\bar{z}^R$ are linear functions of y^L, y^R, z^L, and z^R. Hence it was necessary (in general) in the last section to subdivide X and use (2.3) to assure that, corresponding to (10.4), we could satisfy (8.10). With the assumption that g^L and g^R grow sufficiently slowly, the additional step is unnecessary.

The following example illustrates the ideas just discussed. We develop the results in a way that might occur in practice. That is we impose conditions as they appear necessary or convenient.

Consider the problem

$$y'' = 40(yy')^{1/3}, \qquad y(1) = 40, \qquad y(2) = 320$$

whose solution is $y = 40x^5$. Note that $f(x, y, y') = 40(yy')^{1/3}$ is independent of x. This simplification is neither necessary nor particularly helpful. Substituting Y for y and Y' for y' in f, we obtain $Y'' = 40(YY')^{1/3}$.

From (10.2),

$$\overline{Y}' = 280+20w\{[0, 1]Y^{1/3}(Y')^{1/3}\}[-1, 1].$$

For convenience, assume $0 \in Y$ and $0 \in Y'$. Then

$$\overline{Y}' = 280+20w\{[(y^L)^{1/3}, (y^R)^{1/3}][(z^L)^{1/3}, (z^R)^{1/3}]\}[-1, 1].$$

Since $y(1) > 0$ and $y(2) > 0$, assume $y^R \geqslant -y^L$. Since $y(2)-y(1) > 0$, assume $z^R \geqslant -z^L$. Then

$$[(y^L)^{1/3}, (y^R)^{1/3}][(z^L)^{1/3}, (z^R)^{1/3}]$$
$$=[\min\{(y^Lz^R)^{1/3}, (y^Rz^L)^{1/3}\}, (y^Rz^R)^{1/3}] \subset [-(y^Rz^R)^{1/3}, (y^Rz^R)^{1/3}].$$

Since we may enlarge $\overline{Y}'$ if we like, we accept

$$\overline{Y}' = 280+40(y^Rz^R)^{1/3}[-1, 1]. \tag{10.5}$$

From (10.3), we obtain

$$\overline{Y} = [40, 320]+40(y^Rz^R)^{1/3}[-1, 1]. \tag{10.6}$$

Hence

$$\Delta y^L = 40-40(y^Rz^R)^{1/3}+y^L,$$
$$\Delta y^R = 320+40(y^Rz^R)^{1/3}-y^R.$$

Choose $y^L = -y^R$. Then $\Delta y^L > 0$ and $\Delta y^R < 0$ if

$$y^R-40(y^Rz^R)^{1/3}-320 > 0. \tag{10.7}$$

From (10.5),

$$\Delta z^L = 280-40(y^Rz^R)^{1/3}-z^L,$$
$$\Delta z^R = 280+40(y^Rz^R)^{1/3}-z^R.$$

Choose $z^L = -z^R$. Then $\Delta z^L > 0$ and $\Delta z^R < 0$ if

$$z^R - 40(y^R z^R)^{1/3} - 280 > 0. \tag{10.8}$$

Choose $z^R = y^R$. Then both (10.7) and (10.8) are satisfied if

$$y^R - 40(y^R)^{2/3} - 320 > 0. \tag{10.9}$$

The largest root of this cubic in $(y^R)^{1/3}$ is near 40·2 and (10.9) is satisfied if $(y^R)^{1/3} \geqslant 40{\cdot}2$. Since $(40{\cdot}2)^3 < 64965$, we conclude that inequalities (10.4) are satisfied if $-y^L = y^R = -z^L = z^R \geqslant 64965$. Using the argument applied in the last section, we conclude that both y and y' are contained in the interval $64965[-1, 1]$ for all $x \in X$. The best possible bounds on y and y' are $[40, 320]$ and $[200, 3200]$, respectively.

Thus the bounds are not very good. However, subdividing X and using these crude bounds in the iterative method described in section 6, good bounds can be obtained.

11. Additional notes

We have shown how, under certain conditions, strict bounds can be obtained on the solution of a two-point boundary-value problem. We not only get bounds on the value of the solution at the mesh points but also uniform bounds on the solution between mesh points. If desired, the method could be easily extended to yield interval polynomial bounds between the mesh points.

We have implicitly assumed that a and b were rational numbers that can be expressed in single precision in the computer. If this is not the case, then h, x_1, x_2,... are irrational, in general. These numbers could be replaced by intervals. However, it seems easier to replace x by, say, $x = a + (b-a)t$. Then t takes the values 0 and 1 at the end-points of the interval in which the differential equation is to be solved. Alternatively, we could choose h to be rational and let only $x_1 - a$ and/or $b - x_{n-1}$ be irrational. In this case alternative expressions for (2.1) and (2.2) must be written for $i = 1$ and/or $i = n-1$.

Equations (2.1) and (2.2) are commonly replaced by alternative expressions in practice (see [2]). If the necessary extra derivatives of f can be easily obtained, it seems probable that higher order approximations should be used. Similarly (4.1) and (5.1) could be replaced. For example, suppose the boundary conditions are of the form $y'(a) = y'_0$ and $y'(b) = y'_n$. Then in the crude error bounding method in section 10, it may be better to use, say, an interval version of

$$y'(x) = \tfrac{1}{2}\{y'(a) + y'(b) + (b-x)y''(\theta) - (x-a)y''(\phi)\}$$

in place of (10.2).

To avoid use of $y^{(4)}$, we could replace (2.2) by

$$y_i'' = h^{-2}(y_{i+1} - 2y_i - y_{i-1}) + \frac{h^3}{6}\{y'''(\xi_i) - y'''(\eta_i)\}, \tag{11.1}$$

where $\xi_i \in X_i$ and $\eta_i \in X_{i-1}$. If this equation is used instead of (2.2), we can drop the condition that y have a bounded fourth derivative.

Equation (11.1) may be especially useful in obtaining crude error bounds. For example, consider the differential equation

$$y'' = y' + \sin y.$$

For this example, we find

$$y''' = y'(1 + \cos y) + \sin y$$

and

$$y^{(4)} = y'(1 + 2\cos y) - (y')^2 \sin y + (1 + \cos y)\sin y.$$

In order to use the procedure in section 10, we could replace

$$Y_i'' = f(X_i, Y_i, Y_i') = Y_i' + \sin Y_i$$

by $\overline{Y}_i'' = Y_i' + [-1, 1]$ since $\sin y_i \in [-1, 1]$. Similarly, we could replace

$$Y_i''' = p(X_i, Y_i, Y_i') = Y_i'(1 + \cos Y_i) + \sin Y_i$$

by $\overline{Y}_i''' = [0, 2]Y_i' + [-1, 1]$. However, if we attempt to do this for $y^{(4)}$, the result is not linear in Y_i and Y_i'. Hence the method in section 10 could not be used. But if we replace (2.2) by (11.1), we do not require a bound on $y^{(4)}$. The bounds on y_i'' and y_i''' are linear in Y_i and Y_i' and hence we can apply the crude bounding procedure of section 8.

The procedure whose steps are listed in section 6 yields bounds on y' over the intervals X_i. If bounds on y' at the mesh-points x_i are desired, we obtain sharper results by noting that $y_i' \in Y_{i-1}' \cap Y_i'$. Sharper results, in general, can be obtained using (2.1), which becomes

$$y_i' \in \frac{1}{2h}(y_{i+1}^I - y_i^I) - \frac{h^2}{6} A_i.$$

Note we can attempt to improve y_i^I $(i = 1, \ldots, n-1)$ by replacing y_i^I by $y_i^I \cap Y_i \cap Y_{i-1}$.

REFERENCES

1. Collatz, L. *The numerical treatment of differential equations*, 3rd edn. Springer Verlag, Berlin (1966).
2. Fox, L. *The numerical solution of two-point boundary problems in ordinary differential equations*. Clarendon Press, Oxford (1957).
3. Gendzhoian, G. V. On the two-sided Chaplygin approximations of the solution of two-point boundary problems (in Russian). *Izv. Akad. Nauk armyan. SSR* **17**, 21–6 (1964).

4. HANSEN, ELDON. On solving systems of equations using interval arithmetic. *Math. Comput.* **22**, 374–84 (1968).

5. —— and SMITH, ROBERTA. Interval arithmetic in matrix computations, Part II. *SIAM Jl numer. Anal.* **4**, 1–9 (1967).

6. MOORE, RAMON E. *Interval analysis.* Prentice-Hall, New Jersey (1966).

7. SMITH, ROBERTA and HANSEN, ELDON. A computer program for solving a system of linear equations and matrix inversion with automatic error bounding using interval arithmetic. *Lockheed Missiles and Space Co. report LMSC* 4–22–66–3 (1966).

8 · Ordinary Differential Equations

1. Introduction

LET there be given a system of n ordinary differential equations of first order

$$y' = f(t, y) \tag{1}$$

with the initial condition

$$y(t_a) = s; \qquad s \in R^n \tag{2}$$

so that the solution of (1) and (2) is

$$\tilde{y}(t; t_a, s) \tag{3}$$

for $t \geqslant t_a$. Usually a numerical approximation for $\tilde{y}(t; t_a, s)$ is sought. If interval methods are used, condition (2) can be generalized. Instead of a point $s \in R^n$, a set $W_a \subset R^n$ can be used so that the result is a set of solutions. This set of solutions will be denoted by

$$\overline{W}(t) = \{z : z = \tilde{y}(t; t_a, s),\ s \in W_a\} \tag{4}$$

so that $\overline{W}(t_a) = W_a$.

When using interval arithmetic it is advantageous if W_a can be described by an interval vector or by a product of an interval vector and a point matrix. Intervals will be denoted by the symbol $\lfloor\ \rfloor$, so that $\lfloor c \rfloor$ is an interval and $\lfloor w \rfloor$ is an interval vector.

Using Taylor series, Moore [1] obtains very good results in the numerical integration of ordinary differential equations. This chapter describes a new numerical process that can be realized in several different ways. We will call it 'Three-Process Method' or 3PM.

2. The 3PM process

Assume we are given a system (1) or n ordinary differential equations and a set of initial values W_a^* at the time t_a. Our problem is to construct a set W_b^{**} so that

$$W_b^{**} \supseteq W_b^*, \tag{5}$$

$$W_b^* = \overline{W}^*(t_b), \tag{6}$$

and

$$\overline{W}^*(t) = \{z : z = \tilde{y}(t; t_a, s),\ s \in W_a^*\} \tag{7}$$

where

$$t_b = t_a + h, \quad h > 0. \tag{8}$$

The step length can be determined by process (I) given below.

Three procedures will be described for solving the problem. The procedures are defined by three sub-problems that must be solved. There are many numerical realizations for each procedure so that it is possible to find a large number of combined realizations for the whole 3PM.

2.1. *Process (I)*

Assume we are given an initial time value t_a and an initial set W_a^* at the time t_a. Also given is an integer $k \geqslant 0$.

Our problem is to determine a step length $h > 0$ and an interval polynomial with vector coefficients

$$\lfloor \hat{P}(t-t_a) \rfloor = \sum_{\nu=0}^{k} \lfloor p_\nu \rfloor (t-t_a) \tag{9}$$

so that for all $t \in [t_a, t_b]$,

$$\lfloor \hat{P}(t-t_a) \rfloor \supseteq \overline{W}^*(t). \tag{10}$$

Here t_b is determined by (8) and $\overline{W}^*(t)$ by (7). In most cases it is sufficient to construct $\lfloor \hat{P} \rfloor$ for $k = 0$.

2.2. *Process (II)*

Assume that at the time t_a there is given an initial point $\dot{V}_a^*$ with $\dot{V}_a^* \in W_a^*$. The dot over a variable denotes that the variable is a single value and not a set. Our problem is to find a set V_b^{**} with

$$V_b^{**} \supseteq \dot{V}_b^* \tag{11}$$

where

$$\dot{V}_b^* = \tilde{y}(t_b; t_a, \dot{V}_a^*). \tag{12}$$

Obviously process (II) contains the ordinary problem of numerical integration, starting with a single initial point. For the performance of process (II), the result of process (I) must be used.

2.3. *Process (III)*

Assume there is given a decomposition of W_a^* as the sum of the point $\dot{V}_a^*$ and a set U_a^* so that

$$W_a^* \subseteq \dot{V}_a^* + U_a^*, \quad 0 \in U_a^*. \tag{13}$$

Our problem is to find a set U_b^{**} with the property

$$U_b^{**} \supseteq U_b^* \tag{14}$$

where

$$U_b^* = W_b^* - \dot{V}_b^*. \tag{15}$$

The set U_b^* can be interpreted as the image of the perturbations U_a^* of $\dot{V}_a^*$ if integration is performed from t_a to t_b. For process (III) the result of process (I) must be used.

2.4. *Composition of processes (I), (II), and (III)*

By composition of processes (I), (II), and (III) the whole 3PM can be constructed in a simple way. This is illustrated in diagram (16) (p. 94).

From (16) it can be seen that 3PM is a one-step method. It differs from other integration methods in the separation of the integration into the two processes (II) and (III).

Some realizations of (I), (II), and (III) will now be considered.

3.1. *Realization of process (I)*

Moore ([1], p. 131 et seq.) gives a realization of process (I) for $k = 0$. This realization works very well. A realization for $k > 0$ can be found by using the Picard–Lindelöf iteration.

There may exist an interval vector $\lfloor w_a^* \rfloor$ with the property

$$W_a^* \subseteq \lfloor w_a^* \rfloor .$$

By using Moore's method for $k = 0$, an interval polynomial of degree $k = 0$ can be obtained with

$$\lfloor \hat{P}_0(t-t_a) \rfloor = \lfloor p_0 \rfloor \tag{17}$$

so that, for $t_b = t_a + h$ $(h > 0)$,

$$\overline{W}^*(t) \subseteq \lfloor p_0 \rfloor \tag{18}$$

for all $t \in [t_a, t_b]$. The Picard–Lindelöf iteration now leads to

$$\lfloor \hat{P}_{\nu+1} \rfloor := \lfloor w_a^* \rfloor + \int_0^t \lfloor f(t_a+\tau, \lfloor \hat{P}_\nu(\tau) \rfloor) \rfloor \, d\tau \tag{19}$$

for $\nu = 0, 1, 2, \ldots$.

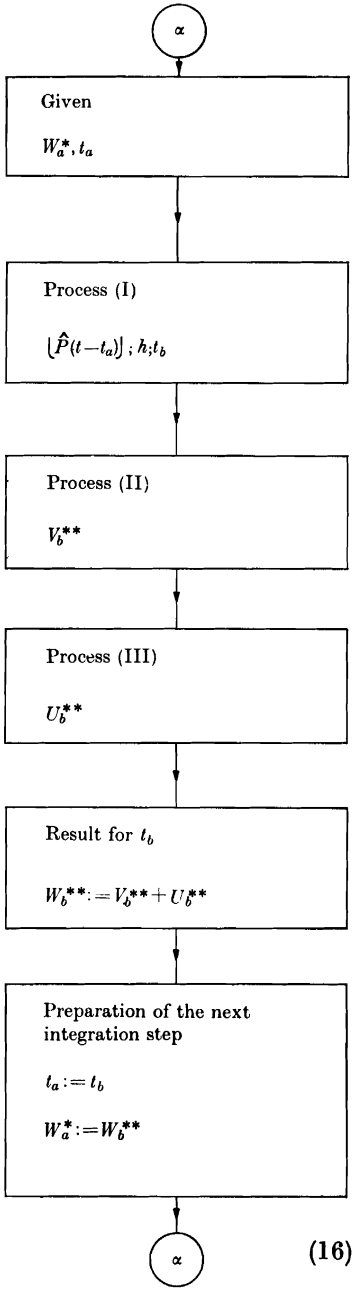
In (19) $\lfloor f \rfloor$ is a representation for f in interval arithmetic operations. By using interval polynomials for the integration (19), it is easy to get $\lfloor \hat{P}_{\nu+1} \rfloor$. But it is necessary to have the ability to limit the degree of $\lfloor \hat{P}_{\nu+1} \rfloor$. This can be done by *vergröberung*. After each step, $\nu \to \nu+1$, and the degree of $\lfloor \hat{P}_{\nu+1} \rfloor$ can be reduced to $\nu+1$. In this way—after k steps using (19)—we obtain $\lfloor \hat{P} \rfloor = \lfloor \hat{P}_k \rfloor$, where

$$\overline{W}^*(t) \subseteq \lfloor \hat{P}(t-t_a) \rfloor \tag{20}$$

for all $t \in [t_a, t_b]$. In practice it is sufficient to have the result (17) for $k = 0$. Only the cases $k = 1$ or $k = 2$ may also be of practical interest.

3.2. *Realization of process (II)*

For the realization of process (II), only a point-integration need be done, so that nearly all one-step integration methods are available if the remainder can be written down. The simplest realization is given by the Taylor series (see Moore [1]).



(16)

A point $\dot{v}_a^*$ must be chosen within $\lfloor w_a^* \rfloor$. Then $\dot{v}_a^*$ is a realization of $\dot{V}_a^*$. The Taylor series through terms of second order now has the form

$$\lfloor v_b^{**} \rfloor := \dot{v}_a^* + h \lfloor f(t_a, \dot{v}_a^*) \rfloor + \\ + \frac{h^2}{2!} \lfloor f'\{[t_a, t_b], \lfloor \hat{P}([0,h]) \rfloor\} \rfloor \qquad (21)$$

where V_b^{**} is realized by

$$V_b^{**} = \lfloor v_b^{**} \rfloor . \qquad (22)$$

Now it is true that

$$\dot{V}_b^* \in \lfloor v_b^{**} \rfloor \qquad (23)$$

because in the remainder of (21) the whole set of solutions $\overline{W}^*(t)$ is included in $\lfloor \hat{P}([0,h]) \rfloor$.

3.3 *Realization of process (III)*

It is very important to have a good realization of process (III). Only then is it possible to get small bounds for the error propagation. By linearization of the given differential equation in the neighbourhood of $\tilde{y}(t; t_a, \dot{V}_a^*)$ and by using the theory of matricants a realization can be found. The interval vector

$$\lfloor \check{u}_a^* \rfloor := \lfloor w_a^* \rfloor - \dot{v}_a^* \qquad (24)$$

is a description of U_a^*. Interval functions $\lfloor l_{ij} \rfloor$ must be obtained which contain the functional matrix of f for each element so that

$$L(t,y) = \{l_{ij}(t,y)\}, \qquad (25)$$

$$l_{ij}(t,y) = \frac{\partial f_i(t,y)}{\partial y_j}, \qquad (26)$$

$$\lfloor l_{ij}(t,y) \rfloor \ni l_{ij}(t,y) \quad \text{for } t_a \leqslant t \leqslant t_b. \qquad (27)$$

Let $\lfloor p_0 \rfloor$ be the result of process (I) for $k = 0$. Then

$$\lfloor \gamma \rfloor := \lfloor L([t_a, t_b], \lfloor p_0 \rfloor) \rfloor \qquad (28)$$

must be computed. Now $\lfloor\gamma\rfloor$ contains all matrices L with arguments in the set $\overline{W}^*(t)$ for $t \in [t_a, t_b]$. Hence it follows by the theory of matricants that for

$$\lfloor Q\rfloor := \dot{f} + \sum_{\nu=1}^{\infty} \frac{h^\nu}{\nu!}(\lfloor\gamma\rfloor)^\nu, \tag{29}$$

$$\lfloor \check{u}_b^{**}\rfloor := \lfloor Q\rfloor \lfloor \check{u}_a^{*}\rfloor \tag{30}$$

so that for the interval vector $\lfloor \check{u}_b^{**}\rfloor$, the relation

$$\lfloor \check{u}_b^{**}\rfloor \supseteq U_b^* \tag{31}$$

holds. The machine calculation of $\lfloor Q\rfloor$ can be done by interval arithmetic without difficulty.

If $\lfloor Q\rfloor$ contains a geometric rotation, the result $\lfloor \check{u}_b^{**}\rfloor$ may not be very good. Moore discusses this difficulty in his book [1]. At Bonn we have evaluated a special method for carrying out a mapping that is better than (30). It is assumed that $U_a^* = W_c^* - \dot{V}_a^*$ has the following representation:

$$\lfloor \check{u}_a^{*}\rfloor \supseteq \dot{T}_a \lfloor\theta_a\rfloor \supseteq U_a^*, \tag{32}$$

$$\lfloor w_a^{*}\rfloor \supseteq \dot{T}_a \lfloor\theta_a\rfloor + \dot{v}_a^* \supseteq W_a^*. \tag{33}$$

The product $\dot{T}_a\lfloor\theta_a\rfloor$ is defined in the sense of the 'united extension' (see Moore [1]):

$$\dot{T}_a\lfloor\theta_a\rfloor = \bigcup_{\theta\in\lfloor\theta_a\rfloor} \dot{T}_a\theta. \tag{33}$$

Process (I) can be realized independently of this representation by using $\lfloor w_a^*\rfloor$. Also $\lfloor Q\rfloor$ can be computed in the old way. Only the mapping (30) must be obtained in a more complicated way:

$$\begin{gathered} \lfloor Q\rfloor \subseteq \dot{Q}_1 + \lfloor Q_2\rfloor, \qquad \dot{Q}_1 \in \lfloor Q\rfloor, \\ \dot{Q}_1 \dot{T}_a \subseteq \dot{T}_b + \lfloor T\rfloor, \\ \lfloor T\rfloor \lfloor\theta_a\rfloor + \lfloor Q_2\rfloor \dot{T}_a \lfloor\theta_a\rfloor \subseteq \lfloor\theta\rfloor. \end{gathered} \tag{34}$$

Then

$$\lfloor Q\rfloor(\dot{T}_a\lfloor\theta_a\rfloor) \subseteq \dot{T}_b\lfloor\theta_a\rfloor + \lfloor\theta\rfloor = U_b^{**} \tag{35}$$

and

$$\dot{T}_b\lfloor\theta_a\rfloor + \lfloor\theta\rfloor \supseteq U_b^*. \tag{36}$$

Now the set U_b^{**} is not described by an interval vector. This implies that the following change should be made in (16):

$$W_b^{**} = V_b^{**} + U_b^{**} \subseteq \lfloor v_b^{**}\rfloor + \dot{T}_b\lfloor\theta_a\rfloor + \lfloor\theta\rfloor. \tag{37}$$

By separation of v_b^{**} in the form

$$\lfloor v_b^{**}\rfloor \subseteq \dot{v}_{1b}^{**} + \lfloor v_{2b}^{**}\rfloor \quad (\dot{v}_{1b}^{**} \in \lfloor v_b^{**}\rfloor), \tag{38}$$

the relation

$$W_b^{**} \subseteq \dot{v}_{1b}^{**} + \dot{T}_b\lfloor\theta_a\rfloor + \lfloor\theta_1\rfloor \tag{39}$$

holds with

$$\lfloor\theta_1\rfloor \supseteq \lfloor\theta\rfloor + \lfloor v_{2b}^{**}\rfloor. \tag{40}$$

By performing the operations indicated in (39) it is possible to get a description of W_b^{**} in the form (33). If $\lfloor S \rfloor \geq \dot{T}_b^{-1}$ then

$$\dot{T}_b \lfloor \theta_a \rfloor + \lfloor \theta_1 \rfloor \subseteq \dot{T}_b (\lfloor \theta_a \rfloor + \lfloor S \rfloor \lfloor \theta_1 \rfloor), \tag{41}$$

$$\lfloor \theta_b \rfloor \supseteq \lfloor S \rfloor \lfloor \theta_1 \rfloor + \lfloor \theta_a \rfloor, \tag{42}$$

so that

$$W_b^{**} \subseteq \dot{v}_{1b}^{**} + \dot{T}_b \lfloor \theta_b \rfloor \tag{43}$$

and (43) is of the form (33). For starting the next integration step all variables must be given a new notation:

$$\begin{aligned} \dot{v}_a^* &:= \dot{v}_{1b}^{**}, \\ \dot{T}_a &:= \dot{T}_b, \\ \lfloor \theta_a \rfloor &:= \lfloor \theta_b \rfloor, \\ \lfloor w_a^* \rfloor &\supseteq \dot{T}_a \lfloor \theta_a \rfloor + \dot{v}_a^* \supseteq W_a^*. \end{aligned} \tag{44}$$

By this more complicated method, very good results can be obtained. By a small modification it is possible to also get 'inside' interval vectors $\lceil \theta_a \rceil$ with the property that the set so described is contained in the set of all solutions.

4. Examples

To illustrate the preceding analysis, we consider two problems we have solved by the methods described. Our first example is the astronomical three-body problem. The differential equations used as the test problem were those for the three-dimensional problem sun–Jupiter–8th moon of Jupiter. This problem was integrated (see Krückeberg [4]) by Taylor series of order three with $h = 1/16$ for 1600 steps (= 100 days) by 3PM. The results are

$$\begin{aligned} \lfloor y_1 \rfloor &= [-1{\cdot}2852300\underline{40},\ 1{\cdot}2852300\underline{03}], \\ \lfloor y_2 \rfloor &= [0{\cdot}8599642\underline{591},\ 0{\cdot}8599642\underline{669}], \\ \lfloor y_3 \rfloor &= [0{\cdot}3014062\underline{070},\ 0{\cdot}3014062\underline{088}], \\ \lfloor y_1' \rfloor &= [0{\cdot}9963448\underline{543},\ 0{\cdot}9963448\underline{606}], \\ \lfloor y_2' \rfloor &= [0{\cdot}5962818\underline{843},\ 0{\cdot}5962819067], \\ \lfloor y_3' \rfloor &= [-0{\cdot}6042486\underline{288},\ -0{\cdot}6042486\underline{228}]. \end{aligned}$$

For our second example, we consider the integration of $y'' = -y$. This very simple example is very interesting from the following point of view. The initial set is rotated through an angle $\zeta = 2\pi$ as the time variable runs from t to $t+2\pi$. It is difficult to find small bounds for the error propagation (see Moore [1]). Without any special technique, the bounds can be overestimated by a factor of about 500 for each rotation.

Moore has reduced this factor to about 16. Using the technique described above, it is possible to perform 12340 integration steps with $h = \pi/10$ and, after 617 rotations, get the following 'inside' and 'outside' interval vectors for the mapping of the starting 'window' $\lfloor\theta_0\rfloor$:

$$\lfloor\theta_0\rfloor = \begin{bmatrix}[-1, +1] \\ [-1, +1]\end{bmatrix},$$

$$\lceil\theta\rceil = \begin{bmatrix}[-0{\cdot}997, +0{\cdot}997] \\ [-0{\cdot}997, +0{\cdot}997]\end{bmatrix}; \qquad \lfloor\theta\rfloor = \begin{bmatrix}[-1{\cdot}002, +1{\cdot}002] \\ [-1{\cdot}002, +1{\cdot}002]\end{bmatrix}.$$

The results were computed using single word length in an interval version of Fortran, called Fortran-i, wherein Fortran expressions are automatically interpreted as interval arithmetic expressions.

REFERENCES

1. Moore, R. E. *Interval analysis.* Prentice-Hall, New Jersey (1966).
2. Nickel, K. Über die Notwendigkeit einer Fehlerschranken-Arithmetik für Rechenautomaten. *Num. Math.* **9** (1) 69–79 (1966).
3. Krückeberg, F. Zur numerischen Integration und Fehlererfassung bei Anfangswertaufgaben gewöhnlicher Differentialgleichungen. *Schriften des Rhein.-Westf. Institutes für Instrumentelle Mathematik an der Universität Bonn,* Nr. 1. Bonn (1961).
4. —— Zur numerischen Integration und Abschätzung des Integrationsfehlers bei gewöhnlichen Differentialgleichungen. *Tagungsbericht über mathematische Methoden der Himmelsmechanik und Astronautik (März 64) Oberwolfach-Berichte,* Heft 1, Herausgeber: Prof. Stiefel, Zürich.
5. Kulisch, U. and Apostolatos, N. Approximation der erweiterten Intervallarithmetik durch die einfache Maschinenintervallarithmetik, *Computing* **2** (3) 181–94 (1967).

9 · Partial Differential Equations

1. Introduction

INTERVAL arithmetic can be used in problems in partial differential equations. But today it seems to be difficult to give a general description of the possibilities. To give some impressions two different examples are selected.

2. Example I

Schröder [6] has studied error estimation for a certain boundary-value problem. The equation has the form

$$\begin{aligned} -\Delta u + f(x,u) &= 0 \quad \text{on } G, \\ u &= \gamma(x) \quad \text{on } \Gamma. \end{aligned} \tag{1}$$

Let ϕ be an approximation for the solution u and define the defect function $d(\phi)$ such that

$$\begin{aligned} d(\phi) &= \Delta y - f(x,y), \quad x \in G; \\ \phi &= \phi(x,y), \quad x, y \in G. \end{aligned} \tag{2}$$

The problem now is to find small bounds for $d(\phi)$ within G in practical cases; here $d(\phi)$ can be a very complicated expression. In the given example we have

$$d(x,y) = d(\phi) = \Delta\phi + \exp\{\phi(x,y) - P(x,y)\}, \tag{3}$$

$$G: x \in [-1,+1]; \; y \in [-1,+1]$$

with

$$\phi(x,y) = \phi_0(x,y) + \phi_1(x,y) \tag{4}$$

and

$$\phi_0(x,y) = \frac{1}{\pi}\{H(x) + H(y) - H(1)\}, \tag{5}$$

$$H(x) = h(x) + h(-x), \tag{6}$$

$$h(x) = 2(1+x)\ln\{4+(1+x)^2\} + \{4-(1+x)^2\}\arctan\{0{\cdot}5(1+x)\} - \pi, \tag{7}$$

$$\phi_1(x,y) = (1-x^2)(1-y^2)\sum_{i=1}^{6} b_i f_i(x,y), \tag{8}$$

and

$$\begin{aligned} f_1 &= 1, & b_1 &= 5{\cdot}6176774\times 10^{-2},\\ f_2 &= x^2+y^2, & b_2 &= -2{\cdot}0087935\times 10^{-2},\\ f_3 &= x^4+y^4, & b_3 &= 6{\cdot}2069297\times 10^{-4},\\ f_4 &= x^2y^2, & b_4 &= 1{\cdot}1764105\times 10^{-2},\\ f_5 &= x^6+y^6, & b_5 &= -5{\cdot}7364814\times 10^{-4},\\ f_6 &= x^4y^2+x^2y^4, & b_6 &= -2{\cdot}4416037\times 10^{-3}, \end{aligned} \tag{9}$$

and

$$P(x,y) = \frac{1}{\pi}(P_1+P_2+P_3+P_4),$$

$$\begin{aligned} P_1 &= q(x,y), & P_2 &= q(-y,x),\\ P_3 &= q(-x,-y), & P_4 &= q(y,-x), \end{aligned} \tag{10}$$

$$\begin{aligned} q(x,y) = (1+x)(1+y)\ln\{(1+x)^2+(1+y)^2\}+\\ +\{(1+x)^2-(1+y)^2\}\arctan\{(1+y)/(1+x)\}. \end{aligned}$$

It is very easy to compute $d(x,y)$ for a special list of values x_i, y_i. But it seems to be impossible to construct uniform bounds for $d(x,y)$. Interval arithmetic is a successful instrument here. The functions P, ϕ, and $\Delta\phi$ can be described by interval polynomials in two variables in the form

$$\begin{aligned} P(x_0+s,y_0+t) &\in \lfloor a_0^*\rfloor + \lfloor a_1^*\rfloor s+\lfloor a_2\rfloor t = \lfloor Q_a\rfloor,\\ \phi(x_0+s,y_0+t) &\in \lfloor b_0^*\rfloor + \lfloor b_1^*\rfloor s+\lfloor b_2\rfloor t = \lfloor Q_b\rfloor,\\ \Delta\phi(x_0+s,y_0+t) &\in \lfloor c_0^*\rfloor + \lfloor c_1^*\rfloor s+\lfloor c_2\rfloor t = \lfloor Q_c\rfloor, \end{aligned} \tag{11}$$

for $$s\in[0,h]; \qquad t\in[0,h]; \qquad s_0, t_0 \in G.$$

This is possible if interval polynomial representations of $\ln(z)$, $\arctan(z)$ are known and arithmetic operations with interval polynomials can be performed. It is important that the degree of the resulting interval polynomials can be reduced and bounded by *Vergröberung*. Now from $\lfloor Q_a\rfloor$, $\lfloor Q_b\rfloor$, and $\lfloor Q_c\rfloor$ new interval polynomials can be constructed so that

$$d(x_0+s,y_0+t) \in \lfloor B_0\rfloor + \lfloor B_1\rfloor s+\lfloor B_2\rfloor t \tag{12}$$

and the bounds of $d(x_0+s,y_0+t)$ in the sub-square $[x_0,x_0+h]$, $[y_0,y_0+h]$ are

$$\lfloor B_0+\min(0,\lfloor B_1 h)+\min(0,\lfloor B_2 h) \leqslant d \leqslant B_0\rfloor + B_1\rfloor h+B_2\rfloor h. \tag{13}$$

By dividing G into about 100, 1000, or 10 000 sub-squares and performing this interval-estimation, more or less close bounds for d can be constructed uniformly in G.

According to Schröder [6] more estimations are necessary (see p. 158, equation (4.6) of [6]) for determination of an error-constant α. This

problem was also solved by interval arithmetic. The so-constructed value $\alpha^* = 1{\cdot}0471266 \times 10^{-3}$ is only a little larger than the value $\alpha = 1{\cdot}0577 \times 10^{-3}$ which was computed using only a finite set of points x_i, y_i within G. Now it is easy to get correct bounds for u. The interval computations in this problem were performed by Wauschkuhn [7].

From a theoretical point of view it is of interest to use not only interval polynomials for defect estimation. The polynomials can be generalized to certain classes of functions with a range of values within a half-ordered space (see Krückeberg [3]).

3. Example II

In some cases it is possible to construct directly the operator for solving a given partial differential equation. If the problem has the form

$$\frac{\partial u(t,x)}{\partial t} = A(t)\frac{\partial u(t,x)}{\partial x} + B(t)u(t,x) + c(t),$$
$$u(0,x) = \phi(x), \quad u \in R^n \tag{14}$$

and if the special example is

$$\frac{\partial^2 u}{\partial t^2} = t^2\frac{\partial^2 u}{\partial x^2} + |t|\frac{\partial u_0}{\partial x} + t^3 \sin t,$$
$$u(0,x) = x^2 - x + 1, \tag{15}$$
$$\frac{\partial u}{\partial t}(0,x) = 1 - x,$$

the solving operator can be written in general form

$$u(t,x) = \Omega(B,t)\{\Omega(C,t)[\phi] + \Omega(-B,t)c\},$$
$$C(t) = \Omega(-B,t)A(t)\Omega(B,t). \tag{16}$$

By performing (16) for (15) using Formac the following result can be obtained:

$$u(t,x) = x^2 - x + 1 + t(1-x^2) + (2x-1)\frac{|t|t^2}{3!} + \frac{t^4}{6} - \frac{x|t|t^3}{6} -$$
$$- \frac{t^5}{10} + \frac{t^6}{10.9} - \frac{t^7}{9.7.4} + t(18 - t^2)\sin t + 6(4 - t^2)\cos t - 24 \tag{17}$$

(see V. Scharf [5]). But the coefficients in (17) are computed in Formac with rounding errors. By using interval arithmetic in combination with Formac it is possible to get correct bounds for the coefficients. In this way upper and lower bounds for the solution u can be constructed.

It seems to be a very successful procedure to combine a system like Formac with the ideas of interval analysis. Furthermore, in this way 'inside' intervals can be constructed.

REFERENCES

1. Moore, R. E. *Interval analysis.* Prentice-Hall, New Jersey (1966).
2. Nickel, K. Über die Notwendigkeit einer Fehlerschranken-Arithmetik für Rechenautomaten. *Num. Math.* **9** (1) 69–79 (1966).
3. Krückeberg, F. Defekterfassung bei gewöhnlichen und partiellen Differentialgleichungen. *Vortrag Oberwolfach im Juni* 1966, Band 9. Birkhäuser-Verlag, Basel (ISNM) (1968).
4. Kulisch, U. and Apostolatos, N. Approximation der erweiterten Intervallarithmetik durch die einfache Maschinenintervallarithmetik. *Computing* **2** (3) 181–94 (1967).
5. Scharf, V. Ein Verfahren zur Lösung des Cauchy-Problems für lineare Systeme von partiellen Differentialgleichungen. Dissertation, Bonn, 1966.
6. Schröder, J. Operator-Ungleichungen und ihre numerische Anwendung bei Randwertaufgaben. *Num. Math.* **9**, 149–62 (1966).
7. Wauschkuhn, U. *Methoden der Intervall-Analysis zur gleichmäßigen Erfassung des Wertebereiches von Funktionen in einer und mehreren Veränderlichen.* Diplom-Arbeit, Bonn (1967).

10 · The Centred Form

1. Introduction

In this chapter we shall prove that a conjecture of Moore's concerning the centred form is correct. We first discuss some introductory ideas.

Consider a rational function $f(x)$ of a real variable x. Since interval arithmetic does not obey the distributive law, it is meaningless, in general, to replace x by an interval X and write $f(X)$. If we specify the order in which the arithmetical steps to evaluate $f(x)$ are to be performed, however, $f(X)$ is then meaningful. It is common practice to write $f(X)$ without the algorithmic steps for evaluation. A particular but arbitrary and unspecified rule of evaluation is usually implied.

Let r be some rule for evaluating $f(x)$ and let $f_r(X)$ denote the interval obtained by using this rule to evaluate $f(x)$ with x replaced by X. We ask the question: what rule r will be such that the width $w\{f_r(X)\}$ of $f_r(X)$ is as small as possible? This is an open question for arbitrary $f(x)$.

Moore [1] discusses certain types of rules. In particular, he discusses what he calls the 'centred form'. Let c be the midpoint of X and write

$$f(x) = f(c)+g(x-c, c). \tag{1.1}$$

This defines $g(x-c, c)$ which, following Moore, we write simply as $g(x-c)$. The function $f(x)$ is said to be in centred form when written in this way.

Note that $x-c$ divides $f(x)-f(c)$. (In fact, one way to obtain g is to perform this division explicitly.) Hence

$$g(x-c) = (x-c)h(x-c)$$

so that

$$f(x) = f(c)+(x-c)h(x-c). \tag{1.2}$$

If we write

$$f(X) = f(c)+(X-c)h(X-c), \tag{1.3}$$

this still does not define $f(X)$ since we have not specified how $h(X-c)$ is to be evaluated. In whatever way $h(X-c)$ is evaluated, however, 'good' results are obtained as we shall see later.

We could write $h(x-c)$ itself in centred form as

$$h(x-c) = h(0)+(x-c)k(x-c)$$

and now we must ask how $k(x-c)$ is to be written, etc. If $f(x)$ is a polynomial, we eventually get

$$f(x) = f(c)+(x-c)f'(c)+\ldots+(x-c)^n\frac{f^{(n)}(c)}{n!},$$

the terminating Taylor series for $f(x)$ expanded about the point c. This result would be in nested form although we have not written it that way.

2. Moore's conjecture

If $f(x_1,\ldots, x_n)$ is a rational function of n variables, Moore defines the centred form in the same way. Thus

$$f(x_1,\ldots,x_n) = f(c_1,\ldots,c_n)+g(x_1-c_1,\ldots,x_n-c_n), \tag{2.1}$$

where c_i $(i = 1,\ldots, n)$ is the mid-point of a given interval X_i. For any function $p(x_1,\ldots, x_n)$, let $\bar{p}(X_1,\ldots, X_n)$ denote the united extension (see [1]) of p for $x_i \in X_i$ $(i = 1,\ldots, n)$. Moore ([1], p. 45) conjectures that with f written in centred form,

$$w\{f(X_1,\ldots,X_n)\}-w\{\bar{f}(X_1,\ldots,X_n)\} = O(d^2), \tag{2.2}$$

where

$$d = \max_{1\leqslant i\leqslant n} d_i$$

and $d_i = w(X_i)$. We assume $\bar{f}(x_1,\ldots,x_n)$ is bounded for

$$x_i \in X_i \quad (i = 1,\ldots,n).$$

We shall now prove that the conjecture is true.

3. The one-dimensional case

In this section we shall prove Moore's conjecture for the case in which f is a rational function of a single variable x. In the next section, we give a different proof valid for any number of variables.

From Theorem 4.4 of [1], if we evaluate the function $h(X-c)$ occurring in (1.3), we obtain

$$h(X-c) = \bar{h}(X-c)+E \tag{3.1}$$

where

$$w(E) = O\{w(X-c)\} = O(d). \tag{3.2}$$

Hence, from (1.3),

$$f(X) = f(c)+(X-c)\{\bar{h}(X-c)+E\}$$
$$\subset f(c)+(X-c)\bar{h}(X-c)+(X-c)E. \tag{3.3}$$

Note that from (3.2),

$$w([-1,1]E) = O(d). \tag{3.4}$$

But $X-c = \frac{1}{2}d[-1,1]$. Hence from (3.3) and (3.4),

$$f(X) \subset f(c)+\tfrac{1}{2}d[-1,1]\bar{h}(X-c)+O(d^2). \tag{3.5}$$

The notation in (3.5) requires explanation. For any functions g and G, the notation $G(X) = g(X)+O(d^2)$ indicates that $g(X)$ and $G(X)$ differ by an interval function whose width is $O(d^2)$.

Let $t \in X$ be such that

$$|h(t-c)| = \max_{x \in X} |h(x-c)|.$$

Then
$$[-1, 1]\bar{h}(X-c) = [-1, 1]|h(t-c)|. \tag{3.6}$$

From (3.5) and (3.6),

$$w\{f(X)\} = d|h(t-c)|+O(d^2). \tag{3.7}$$

Denote $X = [a, b]$. Then obviously

$$w\{\bar{f}(X)\} > |f(b)-f(a)|. \tag{3.8}$$

By Taylor series with remainder,

$$f(a) = f(t)+(a-t)f'(t)+\tfrac{1}{2}(a-t)^2 f''(\xi), \tag{3.9}$$

$$f(b) = f(t)+(b-t)f'(t)+\tfrac{1}{2}(b-t)^2 f''(\eta), \tag{3.10}$$

where t is as defined above. Note $\xi \in X$ and $\eta \in X$. Subtracting (3.9) from (3.10) and substituting into (3.8),

$$w\{\bar{f}(X)\} \geqslant d|f'(t)|+O(d^2). \tag{3.11}$$

From (1.2), $f'(t) = (t-c)h'(t-c)+h(t-c)$

and hence
$$w\{\bar{f}(X)\} \geqslant d|h(t-c)|+O(d^2). \tag{3.12}$$

Comparing this result with (3.7), we see that the proof is complete.

The steps used in this proof can be carried out in the multi-dimensional case. However, the final relations similar to (3.7) and (3.12) do not produce the desired proof except in special cases. In the next section we give another proof valid for any number of variables x_i. However, to simplify the presentation, we consider only the two-dimensional case.

4. The two-dimensional case

Since $f(x)$ is assumed to be rational, we can write

$$f(x_1, x_2) = f(c_1, c_2)+p(x_1-c_1, x_2-c_2)/q(x_1-c_1, x_2-c_2) \tag{4.1}$$

where p and q are multinomials of the form

$$p(x_1-c_1, x_2-c_2) = \sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}(x_1-c_1)^i(x_2-c_2)^j, \tag{4.2}$$

$$q(x_1-c_1, x_2-c_2) = \sum_{i=0}^{m}\sum_{j=0}^{m} q_{ij}(x_1-c_1)^i(x_2-c_2)^j, \tag{4.3}$$

for some integers k and m. Letting $x_1 = c_1$ and $x_2 = c_2$ in (4.1), we see that $p_{00} = p(0,0) = 0$.

Assume we evaluate p in the form it is given in (4.2). For a given term in the sum we obtain

$$\begin{aligned} p_{ij}(X_1-c_1)^i(X_2-c_2)^j &= p_{ij}([-1,1]d_1/2)^i([-1,1]d_2/2)^j \\ &= 2^{-i-j}p_{ij}d_1^i d_2^j[-1,1] \\ &= O(d^{i+j}). \end{aligned}$$

Hence

$$\begin{aligned} p(X_1-c_1, X_2-c_2) &= p_{10}(X_1-c_1)+p_{01}(X_2-c_2)+O(d^2) \\ &= p_{10}[-1,1]d_1/2+p_{01}[-1,1]d_2/2+O(d^2) \\ &= \tfrac{1}{2}(|p_{10}|d_1+|p_{01}|d_2)[-1,1]+O(d^2). \end{aligned}$$

Similarly,

$$q(X_1-c_1, X_2-c_2) = q_{00}+\tfrac{1}{2}(|q_{10}|d_1+|q_{01}|d_2)[-1,1]+O(d^2).$$

We assume $q(X_1-c_1, X_2-c_2)$ does not contain zero so that no division by zero occurs. That is,

$$|q_{00}| > \tfrac{1}{2}(|q_{10}|d_1+|q_{01}|d_2)+O(d^2).$$

Without loss of generality, we can assume $q_{00} > 0$. Hence we compute

$$f(X_1, X_2) = f(c_1, c_2)+\frac{\tfrac{1}{2}(|p_{10}|d_1+|p_{01}|d_2)}{q_{00}-\tfrac{1}{2}(|q_{10}|d_1+|q_{01}|d_2)}[-1,1]+O(d^2).$$

Therefore

$$\begin{aligned} w\{f(X_1, X_2)\} &= \frac{|p_{10}|d_1+|p_{01}|d_2}{q_{00}-\tfrac{1}{2}(|q_{10}|d_1+|q_{01}|d_2)}+O(d^2) \\ &= \frac{1}{q_{00}}(|p_{10}|d_1+|p_{01}|d_2)+O(d^2). \end{aligned} \tag{4.4}$$

Let G and g be any functions such that

$$G(x_1, x_2) = g(x_1, x_2)+O(d^2).$$

Then
$$\bar{G}(X_1, X_2) = \bar{g}(X_1, X_2)+O(d^2).$$

Hence
$$\bar{f}(X_1, X_2) = f(c_1, c_2)+\bar{r}(X_1, X_2)+O(d^2) \tag{4.5}$$

where
$$r(x_1, x_2) = \frac{p_{10}(x_1-c_1)+p_{01}(x_2-c_2)}{q_{00}+q_{10}(x_1-c_1)+q_{01}(x_2-c_2)}. \tag{4.6}$$

Now
$$w\{\bar{r}(X_1, X_2)\} \geqslant |r(b_1, b_2)-r(a_1, a_2)|$$

for any $a_1 \in X_1$, $b_1 \in X_1$, $a_2 \in X_2$, and $b_2 \in X_2$. Denote $\epsilon_1 = \operatorname{sgn}(p_{10})$ and

$\epsilon_2 = \operatorname{sgn}(p_{01})$ and let $a_1 = c_1 - \epsilon_1 d_1/2$, $b_1 = c_1 + \epsilon_1 d_1/2$, $a_2 = c_2 - \epsilon_2 d_2/2$, and $b_2 = c_2 + \epsilon_2 d_2/2$. Then

$$w\{\bar{r}(X_1, X_2)\} \geqslant \frac{\frac{1}{2}(|p_{10}|d_1 + |p_{01}|d_2)}{q_{00} + \frac{1}{2}(q_{10}\epsilon_1 d_1 + q_{01}\epsilon_2 d_2)} + \frac{\frac{1}{2}(|p_{10}|d_1 + |p_{01}|d_2)}{q_{00} - \frac{1}{2}(q_{10}\epsilon_1 d_1 + q_{01}\epsilon_2 d_2)}$$

$$= \frac{1}{q_{00}}(|p_{10}|d_1 + |p_{01}|d_2) + O(d^2).$$

Comparing this result with (4.4) and noting (4.5) yields the desired result since $w\{\bar{f}(X_1, X_2)\} \leqslant w\{f(X_1, X_2)\}$.

REFERENCE

1. Moore, R. E. *Interval analysis.* Prentice-Hall, New Jersey (1966).

11 · Distributions in Intervals and Linear Programming

1. Introduction

THIS book is concerned with a class of methods for evaluating errors in the numerical solution of a mathematical problem. These arise from two sources, initial errors in the data and errors generated during the course of the computational process itself. In either case it seems reasonable to attribute an underlying theoretical probability distribution to error.

The purpose of this chapter is therefore twofold. First, to examine some concepts of interval analysis from the probabilist's viewpoint, and secondly, to describe the application of probabilistic methods to a particular algebraic problem, linear programming, indicating the sorts of things that may be said. It should be mentioned that the use of probabilistic technique in the error analysis of numerical methods for ordinary and partial differential equations is a more difficult matter. Indeed, the rigorous treatment of stochastic differential equations is a deep branch of probability theory (at least with the current Kolmogorov model for the theory).

In section 2, some of the concepts and methods of interval analysis and its variants [5] are placed in a probabilistic framework. Section 3 is based on a suggestion of R. W. Hiorns, made after the author's talk at the symposium. It proposes *quantile arithmetic* (essentially) an extension of triplex arithmetic involving a statistically more sophisticated treatment of error at moderate extra computational expense.

For the sake of completeness, some basic results of the theory of linear programming are set out in section 4. The last section presents a summary of recent work [1, 2, 4, 5, 8, 9, 11] on the *distribution problem of stochastic linear programming*. For present purposes, this problem concerns the distributions of the solution vector and optimal value of a linear program when there are random errors in the data. It is thus intimately connected

with *parametric programming* [7, 10, 12], the study of how these entities change with parametric variation of the data. Unfortunately, computational errors will largely be ignored. Of course in many practical problems, and, hopefully, with the usual simplex methods for linear programming, these will be overwhelmed by errors in data. In Meinguet's terminology (see Chapter 5), it is assumed that simplex methods can be made *gutartig*.

2. Distributions in intervals and interval analysis

Consider a real number x subject to random error. To this corresponds a *random variable* (r.v.) $\mathbf{X}$. The statistical properties of $\mathbf{X}$ are given by its *distribution function* F, a non-decreasing, right continuous mapping of the line into the closed unit interval, giving the probability that $\mathbf{X}$ lies below level x as

$$F(x) = P\{\mathbf{X} \leqslant x\}.$$

For simplicity, we may assume F generates a distribution absolutely continuous with respect to Lebesgue measure on the line, so that $\mathbf{X}$ has a *probability density function* f, i.e.

$$F(x) = \int_{-\infty}^{x} f(x)\, dx.$$

(For practical purposes the integral may be taken in the Riemann sense.) Let us further assume that $f(x) > 0$ for all $x \in (\underline{x}, \bar{x})$, an open interval whose closure will be denoted by X^I, and $f(x) = 0$ otherwise. The closed interval $X^I = [\underline{x}, \bar{x}]$ is called the *support* of $\mathbf{X}$ and is denoted $\operatorname{supp} \mathbf{X}$.

Interval arithmetic deals with two parameters of the distribution of $\mathbf{X}$, namely, the end-points of $\operatorname{supp} \mathbf{X} = X^I$. Triplex arithmetic deals with these two parameters and a third, the *main value*, a computationally significant interior point of $\operatorname{supp} \mathbf{X}$. Sometimes, of course, interest is centred on the single parameter $\bar{x} - \underline{x}$, called the *width* $w(X^I)$ by Moore (see [6] and Chapter 1) and the *span* by Nickel (see Chapter 2). The *degree of significance* discussed by Meinguet in Chapter 5 is another single parameter (an interior point of X^I) summarizing the probabilistic information about $\mathbf{X}$.

A probabilist also attempts to summarize information about the distribution of $\mathbf{X}$ in a few parameters, usually the *mean* or *expected value*

$$E\mathbf{X} = \int_{X^I} x f(x)\, dx,$$

the *median* $m\mathbf{X}$, a number, not necessarily unique, such that $\mathbf{X}$ lies on either side of it with probability 1/2 (both of these are location

parameters), and the *variance* or *dispersion* $V\mathbf{X} = E(\mathbf{X}-E\mathbf{X})^2$ (a scale parameter), when they exist. When the distribution of $\mathbf{X}$ is known only implicitly in a problem, he expends much effort in attempting to find analytic approximations to the mean and variance of $\mathbf{X}$, or bounds on its support. Another common practice is to attempt to show that if $\{\mathbf{X}_n\}$ is a sequence of random variables, then as $n \to \infty$, $\mathbf{X}_n$ converges in a suitable sense to a random variable with known distribution function, for example, Gaussian. This will be dealt with further in section 5. We now attempt some rather trivial probabilistic intuition.

First notice that if a computation involving interval numbers is interpreted as involving random variables, the exact interval for the result of the computation is the support of the corresponding random variable. (The result *is* a random variable, since arithmetic (in fact continuous) operations on random variables generate the same.) Thus interval arithmetic may be looked on as a method of approximating supports of random variables. Krückeberg's elegant interval approximations of the evolution of initial value regions of differential systems (see Chapter 8) may be interpreted similarly in higher dimensions. Here one is concerned with a vector of random variables, or a *random vector* $\mathbf{X}$, whose probability distribution is specified by a *joint* distribution function, now a suitable mapping of several variables into the unit interval. In this case, of course, the support of $\mathbf{X}$ is no longer reasonably a multi-dimensional interval, although it may usually be safely assumed connected and approximated with interval methods. (Recall that absolute continuity is being assumed.)

Returning to one dimension, Nickel's convergent Newton's method for finding a zero of a function (see Chapter 3) may also be given an interpretation in terms of supports of probability distributions. It makes use of the positive probability of successive intervals overlapping to obtain convergence. In probabilistic terms, at each stage of the process a new probability distribution of error is defined on the intersection of the intervals by convolving the distributions on the union of the intervals, truncating the convolution off the overlap and re-normalizing the resulting distribution on it.

Consider next the random variable $\mathbf{X}$ above. There are many symmetric distributions on an interval X^I and for these $E\mathbf{X} = (\bar{x}-\underline{x})/2$. For example, one such is the uniform distribution with density

$$f(x) = \begin{cases} 1/(\bar{x}-\underline{x}) & \text{if } x \in X^I, \\ 0 & \text{otherwise.} \end{cases}$$

At the beginning of a computation in interval or triplex arithmetic, one often assumes such distributions for the input data. However, at the end of the computation, even if the assumption were correct, the distribution of error in the exact interval for the result, and *a fortiori* in the computed interval, will in general be skewed (see, for example, the computations reported by Nickel in Chapter 2). It is thus perhaps of questionable value to write the result of the computation as the arithmetic mean of the interval plus or minus a deviation.

If, at the beginning of a computation in triplex arithmetic, one thinks of the main value of an input number as $E\mathbf{X}$ of the corresponding random variable $\mathbf{X}$, then no difficulty is encountered in interpreting the main value of the sum or difference of a pair of variables $\mathbf{X}$ and $\mathbf{Y}$ as $E(\mathbf{X}\pm\mathbf{Y})$, since expectation is a linear functional. On the other hand, to preserve a similar interpretation for products and quotients, it is necessary to assume $\mathbf{X}$ and $\mathbf{Y}$ are *independent* random variables in order to ensure that, for example, $E\mathbf{X}.\mathbf{Y} = E\mathbf{X}.E\mathbf{Y}$. It is difficult to dispense with this assumption in practice, for consideration of statistical dependency in general would be computationally prohibitive.

However, interval arithmetic treats even two occurrences of the same variable as independent. This may be at least partially overcome by the inclusion of a power operation in the arithmetic. A consequence of treating a variable as independent of itself is that when variables occur many times in the numerator of a rational expression, the resulting interval grows, because the support of the corresponding random variable is the interval sum of the supports of the individual terms.

The effect is reduced by using methods such as *centred forms* (see [6] and Chapter 10) to reduce the occurrence of the variable, and pushing the interval quantities (i.e. the random variables) into the denominator of rational expressions, so as to use the spread (variation) to drive convergence to a constant. For example, using Newton's method for solving a linear equation system $AX = b$ in interval (random) variables,

$$\mathbf{X}_{n+1} = E\mathbf{X}_n - \mathbf{A}^{-1}(E\mathbf{A}E\mathbf{X}_n - E\mathbf{b}).$$

A similar idea is involved in Hansen's methods (see [6] and Chapter 4) for this problem. It is an idea familiar to probabilists. Indeed, most of the methods for reducing interval spread are also used in probability theory. For example, Moore's *united extensions* may be interpreted as taking account of certain statistical dependencies between variables entering an expression.

There is, however, a basic probabilistic limitation in the treatment

of error distributions by current interval methods, even if they were exact. Specifically, this is the simple fact that while the support of the error distribution of the result of some finite computations on interval (random) variables may become arbitrarily large, the error distribution itself may be concentrated in a small interval with probability close to one. In such a case, the error distribution of the result has low dispersion, but long *tails* in which little probability is massed. It is upon consideration of this possibility that quantile arithmetic is based.

3. Quantile arithmetic

Let us consider the approximation of our random variable $\mathbf{X}$ of the previous section by a discrete random variable $\tilde{\mathbf{X}}$. For computational tractability we will assume a three-point distribution for $\tilde{\mathbf{X}}$, although in principle a distribution on more points could be used in the sequel (at the expense, of course, of increased computation). Therefore, suppose $\tilde{\mathbf{X}}$ has the following discrete density function:

$$g(x) = \begin{cases} \alpha & \text{if } x = x_1, \text{ where } P\{\mathbf{X} \leqslant x_1\} = \alpha, \\ 1-2\alpha & \text{if } x = x_2, \text{ where } P\{\mathbf{X} \leqslant x_2\} = 1/2, \\ \alpha & \text{if } x = x_3, \text{ where } P\{\mathbf{X} \leqslant x_3\} = 1-\alpha, \\ 0 & \text{otherwise; where } 0 \leqslant \alpha \leqslant 1/2. \end{cases}$$

Thus $\underline{x} \leqslant x_1 \leqslant x_2 \leqslant x_3 \leqslant \bar{x}$, and x_1, x_2, and x_3 are, respectively, the αth *quantile* the 1/2th *quantile* (or median), and the $(1-\alpha)$th *quantile* of the (absolutely continuous) distribution of X with support X^I. The situation is illustrated in Fig. 11.1.

Now (dropping the tilde notation) an arithmetic will be defined on the space $\mathscr{Q}_\alpha$ of all *independent* random variables of the form $\tilde{\mathbf{X}}$. ($\mathscr{Q}_\alpha$ may be considered as a suitable subset of the space of essentially bounded random variables on the probability space generated by Lebesgue measure restricted to the unit interval.) The number α will be a parameter of the arithmetic which in a machine implementation would normally be fixed. In general, the more concentrated the error distributions of variables approximated by elements of $\mathscr{Q}_\alpha$, the larger may α be chosen. The choice of α is largely a matter of probabilistic interpretation, for we shall see that the machine implementation of quantile arithmetic is the same for positive $\alpha < 0{\cdot}17$. Probably choosing $\alpha = 1/20$ is most reasonable from the probabilistic point of view, so that the 'true' value of the number x lies between x_1 and x_3 with probability 9/10. However, $\alpha = 1/10$ or $\alpha = 1/100$ might also be reasonable for disperse and concentrated error distributions respectively. Formally, $\mathscr{Q}_{1/2}$ is isomorphic to R

and $\mathcal{Q}_0$ to triplex numbers with the main value interpreted as the median of the error distribution of the associated number. Except where explicitly stated otherwise, $0 \leqslant \alpha < 1/2$ will be assumed fixed below and $\mathcal{Q}_\alpha$ will be denoted simply by $\mathcal{Q}$.

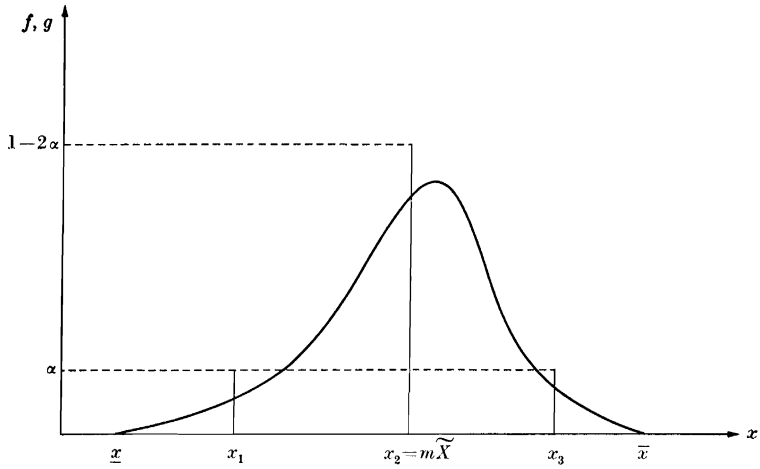


FIG. 11.1

Let $\mathbf{X}, \mathbf{Y} \in \mathcal{Q}$ and let $*$ denote any one of the four arithmetic operations. Then (due to the assumption of independence) the result of performing the binary operation $*$ on $\mathbf{X}$ and $\mathbf{Y}$ is a random variable, $\mathbf{Z}$ having a nine-point distribution with density

$$f(x) = \begin{cases} p_i p_j & \text{if } z = x_i * y_j,\ i, j = 1, 2, 3, \\ 0 & \text{otherwise,} \end{cases}$$

where $p_1 = \alpha$, $p_2 = 1-2\alpha$, and $p_3 = \alpha$, and (x_1, x_2, x_3) and (y_1, y_2, y_3) are the defining triples of $\mathbf{X}$ and $\mathbf{Y}$ respectively. In order to define an (exact) arithmetic on $\mathcal{Q}$ a rule must be given for the approximation of $\mathbf{Z}$ by an element $\mathbf{X} * \mathbf{Y} \in \mathcal{Q}$, except, of course, in case $y_i = 0$ is a divisor for some $i = 1, 2, 3$, when the division operation must remain undefined. (Notice that this is a considerably weaker restriction than its analogue in interval arithmetic [6]. In general, for absolutely continuous error distribution, the fact that zero lies in the support of a distribution does not render the corresponding variable ineligible as a divisor.) A suitable rule to generate the triple (w_1, w_2, w_3) defining $\mathbf{X} * \mathbf{Y} \in \mathcal{Q}$ is the following:

(i) order the nine numbers $x_i * y_j$ in order of increasing magnitude as, say, $z_1, z_2, \ldots, z_9$, with associated probabilities $q_1, q_2, \ldots, q_9$;

(ii) take w_1 to be the largest z_i such that $\sum_{j=1}^{i} q_j \leqslant \alpha$;

(iii) take w_2 to be the smallest z_i such that $\sum_{j=1}^{i-1} q_j \geqslant 1/2$; (3.1)

(iv) take w_3 to be the smallest z_i such that $\sum_{j=1}^{i-1} q_j \geqslant 1-\alpha$.

Noting that a real number β may be identified with a random variable taking the value β with probability one, the real numbers may be embedded in $\mathcal{Q}$ as random variables with defining triples of the form (β, β, β). Of course, $\beta * \mathbf{X} \in \mathcal{Q}$ is defined by the triple $(\beta * x_1, \beta * x_2, \beta * x_3)$ using the rule above. The rule assumes that $\mathbf{X}$ and $\mathbf{Y}$ are independent and thus are not identical. To complete the description of the arithmetic, rules for binary operations involving a single element $\mathbf{X} \in \mathcal{Q}$ are needed. Addition and subtraction may be handled by reducing to multiples $m.X$, m an integer, before computation. Multiplication and division may similarly be reduced before computation to powers. This requires the following rule to generate the triple (w_1, w_2, w_3) defining $\mathbf{X}^p \in \mathcal{Q}$, for p an integer:

take
$$w_1 = \min\{x_1^p, x_2^p, x_3^p\},$$
$$w_2 = x_2^p, \qquad (3.2)$$
and
$$w_3 = \max\{x_1^p, x_2^p, x_3^p\}.$$

Unlike interval arithmetic, special account of dependence must be taken to prevent *under-* as well as over-estimation of the appropriate values.

The stochastic interpretation of elements of $\mathcal{Q}$ may be submerged, and $\mathcal{Q}$ taken to be the space of all ordered triplex $X = (x_1, x_2, x_3)$ of real numbers $x_1 \leqslant x_2 \leqslant x_3$ together with the arithmetic defined in the previous paragraph. The elements of $\mathcal{Q}$ may then be called *quantile numbers*. The following table gives a comparison between operations on the triples $X = (-1, -\frac{1}{3}, \frac{1}{3})$ and $Y = (-\frac{1}{2}, 1, \frac{3}{2})$ performed in exact triplex arithmetic, and exact quantile arithmetic with (for example) $\alpha = 1/20$.

Arithmetic	$X+Y$	$X-Y$	$X.Y$	X/Y	X^2
Triplex	$(-\frac{3}{2}, \frac{2}{3}, \frac{11}{6})$	$(-\frac{5}{2}, -\frac{4}{3}, \frac{5}{6})$	$(-\frac{3}{2}, -\frac{1}{3}, \frac{1}{2})$	$(-1, -\frac{1}{3}, 2)$	$(\frac{1}{9}, \frac{1}{9}, 1)$
Quantile	$(-\frac{1}{6}, \frac{2}{3}, \frac{4}{3})$	$(-2, -\frac{4}{3}, \frac{5}{6})$	$(-1, -\frac{1}{3}, \frac{1}{3})$	$(-\frac{2}{3}, -\frac{1}{3}, \frac{2}{9})$	$(\frac{1}{9}, \frac{1}{9}, 1)$

Notice that the intervals defined by quantile numbers are in every case not wider than those defined by triplex numbers. (For $\alpha = 0$ they would agree, since then quantile arithmetic is simply a complicated version of

triplex arithmetic with a power operation.) The second entries of triplex and quantile numbers agree in the table. This will be so in general as long as $0 \leqslant \alpha < (2-\sqrt{2})/4 \simeq 0{\cdot}17$. In fact, for any positive α in this interval, the probability attached to the second entry, $(1-2\alpha)^2$, exceeds $1/2$ and the resulting arithmetic is the same.

The key to this statement and to the investigation of $\mathcal{Q}$ as an abstract mathematical system is an understanding of the possible orderings of the nine numbers $x_i * y_j$ generated by a binary arithmetic operation on elements of $\mathcal{Q}$ (along with their associated probabilities). Some of these are illustrated in the lattice diagrams of Fig. 11.2. Fig. 11.2 (*a*) refers to ordering relations that hold for addition, multiplication of non-negative numbers, or division of non-positive numbers. Fig. 11.2 (*b*) refers to subtraction, multiplication of non-positive numbers, or division of non-negative numbers. Of course, for specific $X, Y \in \mathcal{Q}$ a linear order results, but these and similar diagrams show that there are many possibilities. It is the existence of these possibilities that makes $\mathcal{Q}$ a rather peculiar system, cf. [6], section 3.1.

Indeed, quantile addition and multiplication are *commutative*, but *not associative*, i.e. if $X, Y, Z \in \mathcal{Q}$,

$$X+Y = Y+X \quad \text{and} \quad X.Y = Y.X,$$

but neither $(X+Y)+Z = X+(Y+Z)$, nor $(X.Y).Z = X.(Y.Z)$, necessarily. For example, taking (as in the sequel) $\alpha < 0{\cdot}17$,

$$\{(-2,-1,1)+(-1,1,2)\}+(-3,-2,2) = (-4,-2,0),$$

while

$$(-2,-1,1)+\{(-1,1,2)+(-3,-2,2)\} = (-4,-2,2).$$

This is perhaps not surprising when it is remembered that a 27-point discrete probability distribution is being approximated by a 3-point distribution in two stages in two different ways. It follows immediately that when one or more of X, Y, or Z are real numbers, the associative laws hold. As in interval arithmetic, the real numbers 0 and 1 are *identities* for quantile addition and multiplication, i.e.

$$0+X = X+0 = X \quad \text{and} \quad 1.X = X.1 = X.$$

However, quantile arithmetic is *not* in general even *sub-distributive*, i.e. defining inclusion for two quantile numbers with identical second entries in the obvious way, $X.(Y+Z) \not\subset X.Y+X.Z$. For example,

$$(-2,-1,2).\{(-1,1,2)+(-3,-2,1)\} = (-4,1,3),$$

while

$$(-2,-1,2).(-1,1,2)+(-2,-1,2).(-3,-2,1) = (-3,1,4).$$

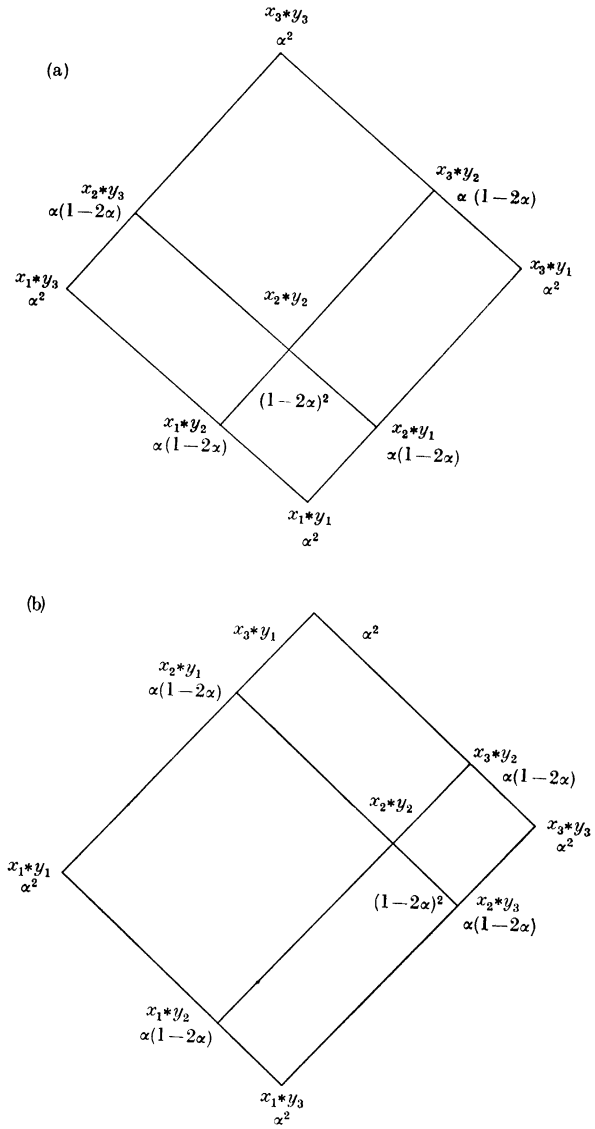

(a)
x_3*y_3
α^2
x_3*y_2
$\alpha\ (1-2\alpha)$
x_2*y_3
$\alpha(1-2\alpha)$
x_3*y_1
α^2
x_1*y_3
α^2
x_2*y_2
$(1-2\alpha)^2$
x_1*y_2
$\alpha(1-2\alpha)$
x_2*y_1
$\alpha(1-2\alpha)$
x_1*y_1
α^2
(b)
x_3*y_1
α^2
x_2*y_1
$\alpha(1-2\alpha)$
x_3*y_2
$\alpha(1-2\alpha)$
x_2*y_2
x_3*y_3
α^2
x_1*y_1
α^2
$(1-2\alpha)^2$
x_2*y_3
$\alpha(1-2\alpha)$
x_1*y_2
$\alpha(1-2\alpha)$
x_1*y_3
α^2

FIG. 11.2

If β is a real number, it is easily verified that the distributive law holds, i.e.

$$\beta.(Y+Z) = \beta.Y+\beta.Z.$$

Quantile arithmetic is *not* generally *inclusion monotonic* with respect to any of the arithmetic operations. For example, even though

$$(-3,-2,1) \subset (-3,-2,2),$$

$$(-2,0,1)+(-3,-2,1) = (-4,-2,1)$$

while

$$(-2,0,1)+(-3,-2,2) = (-4,-2,0).$$

If β is real and $X \subset Y$, it is immediate that

$$\beta+X \subset \beta+Y,$$

$$\beta-X \subset \beta-Y \quad \text{and} \quad X-\beta \subset Y-\beta,$$

$$\beta.X \subset \beta.Y,$$

$$\beta/X \subset \beta/Y, \quad \text{and} \quad X/\beta \subset Y/\beta.$$

More important, however, is the fact that if $f(x^1,x^2,...,x^n)$ is a rational expression in several real variables, then for the corresponding quantile expression $F(X^1,X^2,..,X^n)$, where $x_2^i = x^i$, $i = 1, 2,..., n$,

$$f(x^1,x^2,...,x^n) \subset F(X^1,X^2,...,X^n).$$

This is the principal property required of any interval method and for quantile arithmetic follows easily from the definitions.

Turning to machine implementation, it will again be prudent, after an arithmetic operation resulting in a quantile number X, to round the machine number x_1 down and the machine number x_2 up, using a low-order bit, cf. [6], section 3.2. Since quantile arithmetic ignores all but the most obvious statistical dependencies amongst parts of a calculation, its estimates of the quantiles of the error distribution of the result are likely to be on the conservative side even in exact form. In most cases, error in input data to a calculation is indicated in deviation form. The upper and lower deviations are then used in interval and triplex arithmetic to specify appropriate numbers. A similar procedure for quantile arithmetic would interpret the deviations to be the αth and $(1-\alpha)$th quantiles of the error distribution, and this suggests that α should be small, say 1/20 or 1/100, for interpretation. However, for computation, α may be taken to be 1/10 and the probabilities α^2, $\alpha(1-2\alpha)$, and $(1-2\alpha)^2$ manipulated as the integral percentages 1, 8, and 64, respectively.

For each arithmetic operation $X * Y$ between quantile numbers, a list of nine floating-point numbers with remainders must be computed and stored. The corresponding probabilities may be appended as the last

six bits of each number. As in triplex arithmetic, it may be useful to compute $x_2 * y_2$ to a higher order of precision than the others. It may immediately be stored at this precision as the second entry of the result before being reduced to the precision of the others for placing in the list. Next the computer must order the list. For addition and subtraction, the relations of Fig. 11.2 may be used to reduce the number of comparisons necessary. The probabilities of the numbers in the ordered list are then cumulated until they exceed 10, when the previous entry in the list is rounded down and stored as the first entry of the result, cf. rule (1. ii). The cumulation process continues until the sum of the probabilities exceeds 90, when the next entry in the list is rounded up and stored as the last entry of the result, cf. rule (1. iv). This completes the computation.

The implementation of the power operation does not involve probabilities, see rule (2), and is straightforward.

Set against the increased burden of computation over interval or triplex arithmetic, quantile arithmetic has the advantage of providing the tighter control over error that results from considering error *distribution*. The product of a computation in quantile arithmetic (with $\alpha < 0{\cdot}17$) is the ordinarily computed value, interpreted as the median of the error distribution of the result, bounded by approximations of the αth and $(1-\alpha)$th quantiles of this distribution. Both the propagation of initial error and machine error is bounded only stochastically in quantile arithmetic. Therefore it is expected that some processes unstable in the sense that the intervals expand rapidly when performed in interval or triplex arithmetic, may be stable when performed in quantile arithmetic. Quantile arithmetic may thus be particularly useful in continuous problems (see [6] and Chapters 3 and 9).

4. Some results in linear programming

In this section the analytic and computational theory of linear programming will be outlined in a form useful for the next section. The reader may refer to [3] for details.

Linear programming deals with a dual pair of optimization problems:

$$\pi = \max_{x \geqslant 0} c'x \quad \text{subject to } Ax \leqslant b, \tag{4.1}$$

and

$$\delta = \min_{y \geqslant 0} b'y \quad \text{subject to } A'y \geqslant c, \tag{4.2}$$

where A is an $m \times n$ matrix, c, b, x, and y are vectors of appropriate dimension, prime denotes transpose, and $\leqslant$ denotes a coordinate-wise

vector partial ordering. The *programs* (4.1) and (4.2) are called *primal* and *dual* respectively. The primal or dual *constraints* often appear in equation form. In particular, those of (4.1) and (4.2) become

$$Ax+\bar{y} = b \quad \text{and} \quad A'y-\bar{x} = c, \tag{4.3}$$

upon the addition of *slack* vectors $\bar{y} \geqslant 0$ and $\bar{x} \geqslant 0$. In general, equation constraints simply reduce the dimensions of, for example, the primal slack vector and the corresponding dual non-negativity constraints. (The discussion in the sequel is easily modified to cover equation constraints.) In any case, the set of *feasible* vectors x and y, over which the optima are to be taken, lie in *constraint sets* which are closed convex polytopes. When the constraint set of a program is not vacuous, the program is said to be *feasible*. When the constraint set is non-empty and bounded, so that the optimal *value* of the program is finite, the program is said to be *proper*. The optimal value is achieved at an *optimal vector*.

The basic results of the (finite dimensional) theory of linear programming are given by the following two theorems, which rest on the separation theorem for convex sets.

THEOREM (Duality). *The following cases are mutually exclusive*:

(i) *the primal and dual programs are both feasible, when both have optimal vectors, x^0 and y^0 say, and $\pi = \delta$;*
(ii) *the primal program is feasible and the dual is not, when the primal program is improper, i.e. its value is unbounded;*
(iii) *the dual program is feasible and the primal is not, when the dual program is improper;*
(iv) *neither program is feasible.*

THEOREM (Complementary slackness). *The vectors $x^0 \geqslant 0$ and $y^0 \geqslant 0$ are optimal for the dual programs if $\bar{x}^{0\prime}x^0 = 0$ and $y^{0\prime}\bar{y}^0 = 0$, where $\bar{x}^0 = A'y^0-c$ and $\bar{y}^0 = b-Ax^0$.*

The principal computational algorithm for linear programming (of which many variants exist) is the *simplex method*. It is a direct method based on the fact that the optimum of a linear functional over a convex polytope is attained at an extreme point (vertex) of the polytope. Assuming for the moment that the primal program is feasible, each vertex x of the primal constraint set (a feasible vector) corresponds to an $m \times m$ non-singular sub-matrix B of the $m \times (n+m)$ matrix $(A\ I)$ whose m columns (a vector space basis for R^m) are called a *primal basis*. Similarly, assuming the dual program feasible, each vertex of the dual

constraint set corresponds to a *dual basis* formed from an $n \times n$ non-singular sub-matrix C of the $n \times (m+n)$ matrix $(A' \ -I)$. Specifically, the *basic* coordinates of the primal vertex x (dual vertex y) are given by the entries of $B^{-1}b$ $(C^{-1}c)$ corresponding to the columns A (A') in the basis. (The other coordinates are zero.) It is therefore not surprising that the simplex method is a modification of Gauss–Jordan techniques for the solution of simultaneous linear equations. At each pivot step, both primal and dual bases are changed simultaneously so as to ensure that the optimality criteria of the complementary slackness theorem are satisfied. Eventually, either an optimum for both programs is reached, or one of the dual programs is found to be infeasible.

The computations are carried out in an $(m+1) \times (n+1)$ *tableau* whose initial and final optimal forms are shown below.

A	b
$-c'$	0

*	x^0 $\bar{y}^0$
$-y^{0\prime} \quad -\bar{x}^{0\prime}$	$-\pi = -\delta$

Depending on the existence of an initial feasible primal or dual vector, one of a primal or dual set of rules for choosing the next pivot entry is used. (When neither program has an obvious initial feasible vector, suitable starting procedures are available to find one.) The pivot choice rules preserve the appropriate feasibility at each step. Geometrically the process searches, simultaneously, a vertex of the constraint set of one problem, and a vertex generated by the constraints of its dual, but lying outside the feasible region, i.e. a vector satisfying only *some* of the constraints (including non-negativity constraints). If a pair of feasible vertices are found, then both the complementary slackness theorem and the duality theorem, case (i), guarantee that the process has terminated successfully. The duality theorem applies, since at each step the values of the primal and dual functionals, evaluated at the current vertices, agree. Minus their common value is exhibited in the $(m+1, n+1)$th entry of the tableau. The other cases ((ii)–(iv)) of the duality theorem are translated into forms concerning tableau entries to provide the remaining stopping criteria.

Under the appropriate choice of *non-degeneracy hypothesis*, viz. the vector b (c) does not lie in a proper sub-space of $R(A)$ $(R(A'))$, or, equivalently, the hyperplane $b'y = \text{const.}$ $(c'x = \text{const.})$ is not parallel to a face of the dual (primal) constraint set, the search process is monotonic. That is, the value of the functional of the appropriate problem (and thus

that of its dual) is strictly incremented at each vertex searched. Since there are at most $\binom{m+n}{m}$ vertices, the process must terminate. In the general case, a combination of primal and dual pivot choice rules may be used to ensure that no vertex is searched twice. (Degenerate programs arise from a set of parameter values c, A, b of Lebesgue measure zero in R^{n+mn+m}.)

At successful termination, the optimal dual pair of bases correspond to non-singular sub-matrices P and D of $(A\ I)$ and $(A'\ -I)$, respectively. The basic coordinates of x^0 and $\bar{y}^0$ are given by $P^{-1}b$, exhibited in the $(n+1)$th column of the tableau, and the basic coordinates of y^0 and $\bar{x}^0$ are given by $D^{-1}c$, exhibited as minus the entries of the $(m+1)$th row of the tableau. Under the non-degeneracy hypotheses, the basic coordinates of these vectors are positive, the remainder being zero. Moreover,

$$\pi = c'x^0 = c'(P^{-1}b)_{x^0} = b'(D^{-1}c)_{y^0} = b'y^0 = \delta, \tag{4.4}$$

in an obvious notation.

5. The distribution problem of stochastic linear programming

Now suppose that the parameters of the linear program (4.1) (and its dual (4.2)) are random variables, so that $\mathbf{c}, \mathbf{A}, \mathbf{b}$ form a random vector in R^{m+mn+n} with joint distribution function F. The optimization required in this version of (4.1) is no longer clear, for one could think of choosing a vector x either *before* or *after* the random variables are realized. We shall consider only the latter case. That is, we suppose that after the random variables are realized, the resulting ordinary program is solved for $\pi(c, A, b)$ and $x^0(c, A, b)$, and ask, *a priori*, for the distributions of the random variable $\boldsymbol{\pi} = \pi(\mathbf{c}, \mathbf{A}, \mathbf{b})$, and the n-dimensional random variable $\mathbf{x}^0 = x^0(\mathbf{c}, \mathbf{A}, \mathbf{b})$. (The arithmetic nature of the computations leading to $\boldsymbol{\pi}$ and the elements of $\mathbf{x}^0$ ensure that they *are* random variables.) This distributional problem was first posed by Tintner [11], and is, of course, the problem appropriate to the case of error in the parameter data. (The reader is referred to [4] for a discussion of *decision problems* in which the vector $\mathbf{x}$ must be chosen before the random variables are realized.) The value $\pi(c, A, b)$ is a piece-wise rational function of the parameters c, A, and b, so that if both expectations exist,

$$E\pi(\mathbf{c}, \mathbf{A}, \mathbf{b}) \neq \pi(E\mathbf{c}, E\mathbf{A}, E\mathbf{b}), \tag{5.1}$$

in general. The right side of (5.1) is the value of the ordinary linear program formed by replacing the random variables by their expectations.

It will be seen that when parameter dispersions are small this *expected value program* provides a useful reference point. For convenience, only results for the maximization problem will be discussed below. (Dual results for the minimization problem follow immediately.)

Consider first the distribution problem when only the vector $\mathbf{c}$ of (4.1) is random. In this case the problem becomes one of obtaining the distribution of the maximum of a random linear functional over a closed (not necessarily bounded) convex polytope in R^n, cf. [4]. (When only $\mathbf{b}$ is random, the problem may be treated similarly by considering the dual program (4.2).) From a consideration of the properties of a maximum, it is quite easy to show that $\pi(c)$ is a concave, continuous, piece-wise linear function. Using Jensen's inequality, expression (5.1) may be sharpened in this case to

$$E(\pi(\mathbf{c})) \leqslant \pi(E\mathbf{c}), \tag{5.2}$$

when the expectations exist. The equality can hold only if $\operatorname{supp}\mathbf{c}$ is small. To see this, let Σ denote the set of extreme points, i.e. vertices, $x^1, x^2, \ldots, x^K$, of the constraint set $\{x: x \geqslant 0, Ax \leqslant b\} \subset R^n$. It was seen in the previous section that $K \leqslant \binom{m+n}{m}$, and it follows from the discussion there that by varying c, the elements of Σ may be explicitly calculated using simplex techniques, for given A and b. Simons [10] has shown that to each x^k there corresponds a closed convex polytope in R^n

$$\Delta_k = \Delta(x^k) = \{c: -\infty < c'x^0 < \infty, x^0 = x^k \in \Sigma\}, \tag{5.3}$$

which may be called a *decision region* [2]. Alternatively, we may think of a decision region as corresponding to an optimal basis of the simplex method, or to the non-singular $m \times m$ matrix P defined by it. The interiors of the decision regions are disjoint, and their union $\Delta = \bigcup_{x^k \in \Sigma} \Delta_k$ is a closed convex subset of R^n. If the constraint set is bounded, $\Delta = R^n$. For variation of the parameters within the kth decision region Δ_k, x^k remains optimal and $\pi(c) = c'x^k$ is a linear function, so that (5.2) holds as an equation. From these considerations it is easy to derive necessary and sufficient conditions for $E\boldsymbol{\pi}$ to be finite [4].

THEOREM. *$E\pi(\mathbf{c})$ is finite if and only if* (i) $P\{\mathbf{c} \in \Delta\} = 1$ *and* (ii) $E\mathbf{c}$ *is finite.*

Theoretically, the distribution of the K random variables $\mathbf{c}'x^k$ can be calculated from the joint distribution function of the random vector $\mathbf{c}$, and thus the distribution of $\boldsymbol{\pi} = \max_k\{\mathbf{c}'x^k: k = 1, \ldots, K\}$ determined.

From this distribution, the discrete joint distribution of $\mathbf{x}^0$ with support Σ is easily found. In practice, it may be sufficient to derive or approximate the distribution of $\boldsymbol{\pi}$ for only some of the x^ks. This possibility will be discussed in the context of the general distribution problem, to which we now turn.

When all parameters of (4.1) are random, i.e. $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$ form a random $n+mn+m$ vector, the problem becomes one of maximizing a random linear functional over a random set in R^n. It is now *necessary* to interpret a decision region as corresponding to an optimal basis. Indeed, $\mathbf{A}$ and $\mathbf{b}$ random imply that $\mathbf{P}^{-1}\mathbf{b}$, and hence $\mathbf{x}^0$, will be random, even within a single region. Formally, the kth decision region becomes a set in R^{n+mn+m} defined by

$$\Delta_k = \{c, A, b: P_k^{-1}b \geqslant 0, D_k^{-1}c \geqslant 0\}, \tag{5.4}$$

corresponding to one of the $K = \binom{n+m}{m}$ possible simultaneous choices of m columns of $(A\ I)$ and n columns of $(A' - I)$. Using the form of an optimal tableau, it is easily seen that Δ_k is closed, but it is no longer necessarily a polytope, or even convex. The closed set Δ is again a union of interior-disjoint regions, but is in general neither convex nor polytopic.

A first approach to the general distribution problem is to assume that the support of the random vector formed from $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$ lies in a single decision region. This assumption is most reasonable for a sensitivity analysis of small errors and resembles the assumptions on coefficients for linear equations discussed by Hansen (Chapter 4). This approach was first taken by Babbar [1], who assumed the finiteness of the means and variances of the random variables. It is then possible to express the elements of the random vector $\mathbf{x}^0$ in terms of random determinants, using Cramer's rule, and to approximate the distributions of these determinants by normal distributions. (Without loss of generality, although it might be necessary to consider the dual problem instead, it may be assumed that $\mathbf{A}$ is of rank m with probability one [1].) The approximate distributions of the determinants may in turn be used to give $\boldsymbol{\pi}$ approximately as a ratio of weighted sums of normal variates. Using rather sophisticated techniques, the distribution function $P\{\boldsymbol{\pi} \leqslant \pi\}$ may be approximated, so that in particular $E\boldsymbol{\pi}$ and $V\boldsymbol{\pi}$ may be estimated.

A more straightforward approximation technique, due to Prékopa [8], is to develop $\pi(c, A, b)$ in a finite Taylor series about the value $\underline{\boldsymbol{\pi}}$ of the expected value program and to obtain the distributions of the leading terms. Denoting $E\mathbf{c}$, $E\mathbf{A}$, and $E\mathbf{b}$ by $\underline{\mathbf{c}}$, $\underline{\mathbf{A}}$, and $\underline{\mathbf{b}}$, respectively, it is of course necessary to assume that $\underline{\mathbf{c}}$, $\mathbf{A}$, $\underline{\mathbf{b}}$ belongs to the interior of a

decision region. Using the usual expansions for inverse matrices, it then follows that

$$\pi - \underline{\pi} = -\underline{\mathbf{y}}^{0\prime}(P - \underline{\mathbf{P}})\underline{\mathbf{x}}^0 + (c - \underline{\mathbf{c}})'\underline{\mathbf{x}}^0 + (b - \underline{\mathbf{b}})'\underline{\mathbf{y}}^0 + \rho \qquad (5.5)$$

in an open region about $\underline{\mathbf{c}}$, $\underline{\mathbf{A}}$, $\underline{\mathbf{b}}$, where $\underline{\mathbf{x}}^0$ and $\underline{\mathbf{y}}^0$ are optimal vectors for the primal and dual expected value programs, and ρ is the remainder consisting of second and higher order terms. Denoting the linear terms in (5.5) by λ, it may be shown that the corresponding random variable $\boldsymbol{\lambda}$ has mean 0 and variance σ^2 involving $\underline{\mathbf{x}}^0$ and $\underline{\mathbf{y}}^0$ and the variances and covariances of $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$. Certain simplifications of the expression for σ^2 are possible when elements of $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$ are independent random variables. One may treat the distribution of $\boldsymbol{\pi}$ for highly concentrated error distributions of the parameters by considering a sequence $\{\mathbf{c}^s, \mathbf{A}^s, \mathbf{b}^s\}$ of random vectors whose joint distributions are concentrated in an open region O contained in some Δ_k, and are becoming more and more highly concentrated about $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$. To make this idea rigorous a form of stochastic convergence is required, and Prékopa's results concern the weakest form—*convergence in distribution.* A sequence of random variables (or random vectors) $\{\mathbf{X}^s\}$ converges in distribution to a random variable (or random vector) $\mathbf{X}$ with (joint) distribution function G, denoted $\mathbf{X}^s \xrightarrow{D} \mathbf{X}$, if the sequence of corresponding (joint) distribution functions $\{G^s\}$ converges to G at all its points of continuity. For example, if

$$G^s(\xi) = P\{\mathbf{X}^s \leqslant \xi\} \to \Phi(\xi) = \frac{1}{\sqrt{(2\pi)}\nu} \int_{-\infty}^{\xi} e^{-(\eta-\mu)^2/2\nu^2}\, d\eta \quad (-\infty < \xi < \infty),$$

then $X^s \xrightarrow{D} N(\mu, \nu^2)$, the normal random variable with mean μ and variance ν^2.

THEOREM. *If, as* $s \to \infty$, $P\{(c^s, A^s, b^s) \in O\} \to 1$ *and:*

(i) $\mathbf{c}^s, \mathbf{A}^s, \mathbf{b}^s \xrightarrow{D} \mathbf{c}, \mathbf{A}, \mathbf{b}$,

(ii) $\boldsymbol{\lambda}^s/\sigma^s \xrightarrow{D} N(0, 1)$,

and (iii) $\boldsymbol{\rho}^s/\sigma^s \xrightarrow{D} 0$;

then $(\boldsymbol{\pi}^s - \boldsymbol{\pi})/\sigma^s \xrightarrow{D} N(0, 1)$, *i.e.* $\boldsymbol{\pi}^s$ *is asymptotically* $N(\boldsymbol{\pi}, (\sigma^s)^2)$.

In order to ensure condition (ii) of the theorem, it is sufficient to assume the random vectors $\mathbf{c}^s$, $\mathbf{A}^s$, $\mathbf{b}^s$ have joint (multivariate) normal distributions. A similar limit theorem is available for the case when, instead of condition (i), the dimensions of the random vectors $\mathbf{c}^s$, $\mathbf{A}^s$, $\mathbf{b}^s$, m and n, are increasing together to infinity (as in Hilbert matrix problems). Here, however, the analogue of condition (iii) is unreasonable,

and the quadratic terms of the Taylor series must be considered. These theorems justify, for highly concentrated error distributions, normal approximations to error in $\boldsymbol{\pi}$ centred at $\underline{\boldsymbol{\pi}}$, the computed value. For highly concentrated distributions, the assumptions of the theorems are reasonable. However, they are unreasonable for genuine randomness in the parameters (e.g. due to weather), and probably even for measurement error arising in practice. In general, the support of the parameter distribution will intersect several decision regions and higher order approximations are necessary within each. Hanson [5] has given such approximations to the means and variances of the elements of $\mathbf{x}^0$ within a single decision region.† These may be used to generate corresponding approximations for $\boldsymbol{\pi}$. Hanson presents a simple example of (4.1) with only $\mathbf{A}$ random, in which, due to the conditioning of the constraints, a small variation in the parameters leads to an underestimate of $E\boldsymbol{\pi}$ by $\underline{\boldsymbol{\pi}}$ of over 30 per cent. Estimating $E\boldsymbol{\pi}$ by $\underline{\boldsymbol{\pi}}$, and more generally the distribution of $\boldsymbol{\pi}$ with a symmetric distribution, may thus be wildly in error, cf. expressions (1) and (2).

It is therefore useful to have an existence theory for the general distribution problem. This has essentially been provided by Bereanu [2], who considered the case when the support of the (marginal) joint distribution of $\mathbf{A}, \mathbf{b}$ lies in the positive orthant of R^{mn+m}. The present discussion is based on the more general definition (5.4) of a decision region, and follows trivially from Bereanu's considerations. An immediate necessary condition for $E\boldsymbol{\pi}$ to be finite is

$$P\{(\mathbf{c}, \mathbf{A}, \mathbf{b}) \in \Delta\} = 1. \tag{5.6}$$

(Recall that Δ is the union of the K decision regions Δ_k given by (5.4).) When $P\{\mathbf{A} > 0, \mathbf{b} \geqslant 0\} = 1$, it follows from the duality theorem, case (i), that (5.6) is satisfied. Necessary and sufficient conditions for the finiteness of $E\boldsymbol{\pi}$, in terms of (5.6) and the moments of $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$, are not so far known explicitly in the general case. As in the special case when only $\mathbf{c}$ is random, the distributions of $\boldsymbol{\pi}$ and $\mathbf{x}^0$ can in principle be obtained from a consideration of the decision regions which intersect the support of $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$. Indeed, it has been mentioned that the decision regions are interior disjoint, and for the case when the joint distribution of $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$ is absolutely continuous (with respect to Lebesgue measure on R^{n+mn+m}), it may be shown that

$$P\{(c, A, b) \in \Delta_k \cap \Delta_{k'}, k \neq k'\} = 0.$$

† I am indebted to Dr. B. Bereanu for this reference.

Consider the restrictions $\mathbf{c}_k$, $\mathbf{A}_k$, $\mathbf{b}_k$ of the random vector formed from $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$ to the decision regions Δ_k, $k = 1,\ldots, K$. The basic coordinates of the corresponding restrictions $\mathbf{x}_k^0$ or $\mathbf{x}^0$ are given by $\mathbf{P}_k^{-1}\mathbf{b}_k$. Similarly, the restrictions π_k of π to Δ_k are given by $\mathbf{c}_k' \mathbf{x}_k^0 = \mathbf{c}_k'(\mathbf{P}_k^{-1}\mathbf{b}_k)_{x_k^0}$, cf. (4.4). With suitable modifications for the behaviour of π and $\mathbf{x}^0$ on the intersecting boundaries of the decision regions, distributional considerations may thus be decomposed into similar considerations for regions whose intersections have zero probability, and on which the forms of π and $\mathbf{x}^0$ are known explicitly. (Rigorously, the procedure is one of representing the random variables in terms of sums of their conditional expectations with respect to a suitable partition of R^{n+mn+m}. In the absolutely continuous case, the decision regions themselves will suffice without disjunctification for this partition.) In theory, therefore, the distributions of π and $\mathbf{x}^0$ are completely determined.

In practice, however, since the multi-dimensional transformations and integrals involved in a given problem are prohibitively complicated, some kind of approximative computational technique is required.

A practical method is Monte Carlo simulation. Some experience along these lines has been reported by Sengupta [9], where the sample data on $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$ were drawn from a time series of actual observations arising from an agricultural resource allocation problem. Sengupta's method was to compute the value of the functional at all the extreme points of the constraint set for each sample parameter triple and to rank these values in descending order. Upon computing the sample means and variances for the ranks, it was found that the sample variance of the first rank (i.e. of the optimal value) exceeded that of the second (called a *truncated maximand*), the variance of the second rank exceeded that of the third, etc. Using methods for obtaining the sampling distribution of an extreme value in the theory of order statistics, it is possible to give sufficient conditions for this agreement between the rankings of the sample means and variances to hold, both in finite samples and asymptotically [2]. Perhaps more interesting from the present point of view is the fact that Sengupta was able to fit a beta distribution skewed in the downward direction (i.e. with a long upper tail) to the sample distribution of π.

Another possibility for approximating the distributions of π and $\mathbf{x}^0$ is to restrict attention to a few decision regions, using interval methods (see [6] and Chapter 4) to compute, in quantile arithmetic, say, the quantile vectors $\mathbf{P}^{-1}\mathbf{b}$ and $\mathbf{D}^{-1}\mathbf{c}$ and the quantile number π. Beginning with the optimal basis of the expected value program, a tree structure

of optimal tableaux could be pursued, following negative overlaps of the intervals for basic coordinates until the returns, in terms of the probabilities attached to the corresponding decision regions, were negligible. These probabilities would be compounded from those assigned to the negative lower quantiles of basic coordinates at each branch. (The parametric primal-dual version of the simplex method [3] using the product form of the inverse might prove useful for this procedure.) In this connection it should be mentioned that Hanson [5] has estimated the probability that the random vectors $\mathbf{P}_k^{-1}\mathbf{b}_k$ and $\mathbf{D}_k^{-1}\mathbf{c}_k$ corresponding to the decision region of the optimal basis of the expected value program are *not* non-negative. Assuming that these vectors are jointly normally distributed and that with high probability only one of the elements of $\mathbf{P}_k^{-1}\mathbf{b}_k$ is negative at a time, this probability may be compounded from normal probabilities after estimating appropriate means and variances. Oettli, Prager, and Wilkinson [7] have developed a parametric linear programming method to solve a certain interval problem for linear equation systems.† Investigation of similar problems using parametric methods may prove fruitful for the distribution problem.

REFERENCES

1. Babbar, M. M. Distributions of solutions of a set of linear equations (with an application to linear programming). *J. Am. statist. Ass.* **50**, 854–69 (1955).

2. Bereanu, B. On stochastic linear programming distribution problems. Stochastic technology matrix. *Z. Wahrscheinlichkeitstheorie verw. Geb.* **8**, 148–52 (1967).

3. Dantzig, G. B. *Linear programming and extensions.* Princeton (1963).

4. Dempster, M. A. H. On stochastic programming: I. Static linear programming under risk. *J. math. Analysis Applic.* **11**, 304–43 (1968).

5. Hanson, M. A. Errors and stochastic variations in linear programming. *Aust. J. Stat.* **2**, 41–6 (1960).

6. Moore, R. E. *Interval analysis.* Prentice-Hall, New Jersey (1966).

7. Oettli, W., Prager, W., and Wilkinson, J. H. Admissible solutions of linear systems with not sharply defined coefficients. *SIAM Jl num. Anal.* **2**, 291–9 (1965).

8. Prékopa, A. On the probability distribution of the optimum of a random linear program. *SIAM Jl Control* **4**, 211–22 (1966).

9. Sengupta, J. K. The stability of truncated solutions of stochastic linear programming. *Econometrica* **34**, 77–104 (1966).

† I am indebted to Dr. E. Hansen for this reference.

10. Simons, E. A note on parametric linear programming. *Mgmt Sci.* **8**, 355–8 (1962).

11. Tintner, G. Stochastic linear programming with applications to agricultural economics. *Proceedings of the second symposium on linear programming* (editor H. Antosiewicz). National Bureau of Standards, Washington (1955).

12. Williams, A. C. Marginal values in linear programming. *SIAM Jl appl. Math.* **11**, 82–94 (1963).

Index

PRINTED IN GREAT BRITAIN
AT THE UNIVERSITY PRESS, OXFORD
BY VIVIAN RIDLER
PRINTER TO THE UNIVERSITY