A MORE COMPLETE INTERVAL ARITHMETIC

W.M. Kahan
University of Toronto

Lecture notes prepared for a summer course at the University of Michigan, June 17-21, 1968.

Contents:

A MORE COMPLETE INTERVAL ARITHMETIC

Abstract

So far, all published schemes for Interval Arithmetic (see references) have prohibited division by any interval-number containing zero. This prohibition is inconvenient and unnecessary; we propose to avoid it by adjoining what we call *exterior* intervals to the usual *interior* intervals of Interval Arithmetic, thereby obtaining a number system which resembles the extended real numbers closed at ∞ .

1. Notation

Lower case italics $a,b,c,\ldots$ are used for real numbers, lower case Greek $\alpha,\beta,\gamma,\ldots$ for extended real numbers, upper case $A,B,C,\ldots$ for interval-numbers.

The real numbers are identified in the usual way with the points on a straight line. The set Ω of *extended real numbers* is obtained from the reals by adding the symbol ∞ , just as the projective line is obtained by adding a point at ∞ to close the ordinary straight line. Arithmetic operations upon extended real numbers have the usual identification with geometrical operations upon points in the projective line (cf. Coxeter (1949) ch.11) subject to certain reservations concerning the indefinite forms $0/0$, $\infty\pm\infty$, ∞/∞ and $\infty\cdot 0$ whose values we shall later define to be the interval-number Ω consisting of the whole projective line. Although this assignment may occasionally waste

information, it cannot be misleading*.

Interval-numbers are by definition non-empty subsets of the extended real numbers corresponding to intervals on the projective line. We distinguish *exterior* intervals, which contain ∞ in their interior, from *interior* intervals, which do not; a further distinction concerns *finite interior*

---

* However, implementing *ordinary* arithmetic with extended real numbers correctly on a computer with finite word-length is a complicated business. I know of no such implementation in hardware that is not misleading, despite occasional mistaken advertisements to the contrary. For example, consider Control Data's 6000 series of computers; when they execute the *FORTRAN* sequence

```
X = 2.0**1069
Y = 4.0*X
Z = Y - 2.0*(X+X)
T = (((Y-X) - X) - X) - X
U = 1.0/T
```

they produce correctly $X = 2^{1069}$, $Y = \infty$ and $Z =$ *indeterminate*, but misleadingly $T = \infty$ with *overflow* and $U = 0.0$ with *overflow*. See CDC's reference manual (1967) pp.3-15 to 3-20.

intervals which neither contain nor touch the point at ∞ . We shall write $[\alpha,\beta]$ for *closed* intervals which include both *end-points* α and β ; we shall write $]\alpha,\beta[$ for *open* intervals which include neither end-point; we shall also allow intervals $]\alpha,\beta]$ and $[\alpha,\beta[$ which include that end-point next to one bracket but not the other. Only closed finite interior intervals are discussed in Moore's book (1966). Our scheme too can be restricted formally to just the closed intervals, thereby simplifying its implementation on some computers (see §5.*i*); although such a restriction may occasionally waste information when an end-point is included tnat we might have preferred to exclude, the restriction need never be misleading.

Here is how the symbol strings $[\alpha,\beta]$, $]\alpha,\beta]$, $[\alpha,\beta[$ and $]\alpha,\beta[$ shall be interpreted as interval numbers for all extended real α and β . Represent the projective line as a circle so oriented that, as the real number x increases from a to $b > a$, the point x moves counter clockwise. When $\alpha \neq \beta$, the string $[\alpha,\beta]$ represents the closed interval described by moving on the circle from α to β counter clockwise; reversing the first or the second bracket merely causes the adjacent α or β respectively to be deleted from the interval. When $\alpha = \beta$ certain almost arbitrary conventions are invoked;

$[\alpha,\alpha] \equiv \alpha$;

$[\alpha,\alpha[\equiv$ all extended reals except α ;

$]\alpha,\alpha]$ ≡ Ω ≡ all extended reals ;

$]\alpha,\alpha[$ ≡ the empty set.

Here are some examples.

$x \in [-1,1]$ <=> $-1 \le x \le 1$

$\xi \in [1,-1]$ <=> $\xi = \infty$ or $\xi \le -1$ or $\xi \ge 1$

$\xi \in [1,-1[$ <=> $\xi = \infty$ or $\xi < -1$ or $\xi \ge 1$

$x \in [1,\infty\ [$ <=> $x \ge 1$

$x \in]\infty,1\ ]$ <=> $x \le 1$

$\xi \in [1,1\ [$ <=> $\xi \ne 1$

$x \in [\infty,\infty\ [$ for all real x .

Finally, just as the n-tuple $(\xi_1;\xi_2;\ldots;\xi_n)$ is identified with a point in an n-dimensional extended real space, so shall $(X_1;X_2;\ldots;X_n)$ be identified with the region(s) in that space where each coordinate $\xi_i \in X_i$. When every X_i is an interior interval, that region is just a (possibly infinite) parallelepiped.

Now one purpose of Interval Arithmetic can be explained. Ideally, a numerical computation free of error can be regarded as a mapping $\underline{\mu}$ from a space of *data-points* $(\alpha;\beta;\gamma;\ldots)$ into a space of *results* $(\xi;\eta;\zeta;\ldots)$. Rounding errors and other uncertainties distort this mapping, thereby generating *misinformation* to the extent that the differences between computed results and ideal results are unknown. Interval Arithmetic purports to eliminate misinformation, at the cost of extra computation and some loss of information, by providing

rigorously justifiable estimates for the results. An Interval Arithmetic computation can be regarded as a mapping $\underline{M}$, from regions $(A;B;\Gamma;\ldots)$ in data-space to regions $(\Xi;H;Z;\ldots)$ in result space, so related to $\underline{\mu}$ that $\underline{\mu}$ maps each point $(\alpha;\beta;\gamma;\ldots)$ in $(A;B;\Gamma;\ldots)$ into a point $(\xi;\eta;\zeta;\ldots)$ contained in $(\Xi;H;Z;\ldots)$. Given $\underline{\mu}$, there are several easy routines for deriving a related $\underline{M}$, and often enough the most naïvely derived $\underline{M}$ approximates $\underline{\mu}$ adequately for practical purposes. However, the problem of precision in conventional arithmetic has its analogue in Interval Arithmetic - to obtain from $\underline{\mu}$ an $\underline{M}$ whose result-regions are not too much bigger than necessary.

2. Arithmetic

Although our main objective is to extend the rules of rational arithmetic from real variables to extended real variables and then to interval-variables, it turns out to be more convenient first to deal with an intermediate system consisting of the extended reals in Ω and Ω itself; here Ω is an interval-number consisting of all the extended reals. In this intermediate system, the arithmetic operators $+$, $-$, $\cdot$ and $/$ are defined to act upon the symbols Ω and ∞ in the following natural ways:

(i) *Any arithmetic operation with* Ω *as an operand reproduces* Ω.

(ii) $0/0 \equiv \infty+\infty \equiv \infty-\infty \equiv \infty/\infty \equiv 0\cdot\infty \equiv \infty\cdot 0 \equiv \Omega$.

(iii) $+\infty \equiv -\infty \equiv \xi\cdot\infty \equiv \infty\cdot\xi \equiv \infty+x \equiv x+\infty \equiv \infty/x \equiv \xi/0 \equiv \infty$
for all extended reals $\xi \neq 0$ and $x \neq \infty$.

(iv) $x/\infty \equiv 0$ for all $x \neq \infty$.

The foregoing definitions yield a closed system in which the commutative and associative laws remain as valid as for real variables, but there are two important failures.

Cancellation: If $(\alpha\cdot\xi)/(\beta\cdot\xi) \neq \alpha/\beta$ then $(\alpha\cdot\xi)/(\beta\cdot\xi) = \Omega$;
if $(\alpha-\xi) - (\beta-\xi) \neq \alpha-\beta$ then $(\alpha-\xi) - (\beta-\xi) = \Omega$.

Distribution: If $\alpha\cdot\xi + \beta\cdot\xi \neq (\alpha+\beta)\cdot\xi$ then $\alpha\cdot\xi + \beta\cdot\xi = \Omega$;
if $\alpha/\xi + \beta/\xi \neq (\alpha+\beta)/\xi$ then $\alpha/\xi + \beta/\xi = \Omega$.

These failures imply that the value assigned to a rational expression involving extended real variables may change if the distributive or cancellation law is invoked before the expression is evaluated. Fortunately, the value cannot vary arbitrarily; it turns out that a rational expression cannot have more than two values in our system, and cannot have two values unless one of them is Ω . The other is just that value which would be assigned to the expression if it were regarded as a rational function of real variables, with the symbol ∞ denoting a limit for a variable, or a pole of the function. Unfortunately, information can be lost in our system whenever an expression must be assigned the value Ω no matter how it is reordered despite that it deserves a better value. For example,

$$x\cdot(x-y)\cdot(x+y)/(x\cdot x + y\cdot y)$$

yields the value Ω no matter how it is reordered when $x = y = 0$, even though its limit as $x \to 0$ and $y \to 0$ is 0, as can be seen by rearranging it to read

$$x\cdot\left(1 - 2/\left(1 + (x/y)^2\right)\right) \quad .$$

(This expression could be evaluated more precisely if we introduced the square function into our system thus;

$$\Omega^2 \equiv [0,\infty] \quad , \text{ and otherwise } \xi^2 \equiv \xi\cdot\xi \quad ;$$

doing so foreshadows the interval arithmetic to come.)

The distinction between rational expressions $f(\xi_1;\xi_2;\ldots;\xi_n)$ in n extended real variables and rational functions $f(x_1;x_2;\ldots;x_n)$ in n real variables is an important distinction which should not be allowed to escape into the ambiguities of our notation. The expressions must be evaluated by rules, familiar to compiler-writers, which do not allow parentheses to be removed by the distributive laws nor, in practical implementations, the associative laws. The functions are representable in infinitely many ways by different expressions, all equivalent by the laws of rational algebra for all arguments except possibly on a subset of dimension less than n. Two different expressions representing the same function can differ only where one of the expressions evaluates to Ω in our intermediate extended system. However, as exemplified above, a function may be

continuous in the topology of the projective line (in which ∞ is an allowed limit-point) at some argument without there being any way to express that function as an expression whose evaluation in our system yields the correct limiting value instead of just Ω . This limitation can be partly circumvented by a further extension of the system to include all interval-numbers.

Having specified how the arithmetic operators $+$, $-$, $\cdot$ and $/$ shall act upon degenerate intervals $\alpha = [\alpha,\alpha]$ and upon Ω , we are ready to define interval arithmetic for more general intervals. Our definition is like Moore's (1966, p.8); if $*$ is one of the operators $+$, $-$, $\cdot$ or $/$, we define

$$A*B \equiv \bigcup \alpha*\beta \quad \text{over} \quad \alpha\epsilon A \quad \text{and} \quad \beta\epsilon B$$

to obtain the interval-valued sum, difference, product or quotient of two interval-numbers A and B . Unlike Moore's definition, ours covers all interval-numbers A and B . Here are some examples:

$[0,1] + [1,2] = [1,3]$, $[3,3.1] -]0,0.1[=]2.9,3.1[$,

$]-4,-1]\cdot[-6,5[=]-20,24[$, $-[-1,2] = (-1)\cdot[-1,2] = [-1,-1]\cdot[-1,2]$

$= [-2,1]$,

$[-1,1]/[-3,-1/2] = [-2,2]$, $[-3,-1/2]/[-1,1] = [1/2,-1/2]$,

$[1,2] - [1,2] = [-1,1]$, $[2,1] - [2,1] =]\infty,\infty] = \Omega$,

$]0,1]/]0,1] =]0,\infty[$, $[0,1]/[0,1] = \Omega$,

$1/(1 + 2/[-1,1[) = 1/(1 +]2,-2]) = 1/]3,-1] = [-1,1/3[$,

$[-1,1[/([-1,1[+ 2) = [-1,1[/[1,3[= [-1,1[$.

Evidently the commutative and associative laws remain applicable to interval expressions, but not the cancellation or distributive laws. Instead we have

Sub-distributivity: $A\cdot(B+C) \subseteq A\cdot B + A\cdot C$,

$(B+C)/A \subseteq B/A + C/A$.

Sub-cancellation: $(A\cdot B)/(C\cdot B) \supseteq A/C$,

$(A-B)-(C-B) \supseteq A-C$.

In general, most rules by which parentheses may be manipulated without changing the value of a real expression are inapplicable to interval expressions, to which a host of weaker rules are applicable instead. For example,

$x\cdot(B+C) = x\cdot B + x\cdot C$ for all real x ;

$A\cdot(B+C) = A\cdot B + A\cdot C$ if $B\cdot C \subseteq]0,\infty[$ // and $\infty \notin A$;

$(A-x) - (C-x) = A-C$ for all real x .

Furthermore, there are theorems about interval arithmetic that are not needed for real arithmetic; an example is

Inclusion-monotonicity: If $A \subseteq X$ and $B \subseteq Y$ then

$A*B \subseteq X*Y$ for any operation $*$ in the set $\{+ , - , \cdot , / \}$.

Whereas the real numbers are *totally ordered* ($x < y$ or $x = y$ or $x > y$) , the interval-numbers are at best *partly ordered*; we can write $A > B$ only when some real x exists such that $A \subseteq]x,\infty[$ and $B \subseteq]\infty,x[$, and $A \geq B$ whenever $A \subseteq [x,\infty[$

and $B \subseteq]\infty,x]$ *. This ordering cannot apply to exterior intervals, nor to ∞ .

The foregoing differences between real arithmetic and our interval arithmetic are common to all other proposed schemes for interval arithmetic. In fact, everything that can be done with other brands† of interval arithmetic can be done with ours, and sometimes more simply in our scheme because it has been designed to admit fewer exceptions. What follows is intended to support the foregoing claims by indicating roughly the extent to which the scheme described by Moore (1966) is a subset of ours. But formal proofs for our theorems and other claims have been omitted to save space, thereby increasing the risk that our mistakes may have escaped detection.

3. Functions

Consider first the n-th power function α^n for positive integers n . The familiar definition

---

* Note that $A \not\geq A$ unless A is degenerate; here the symbol "$\geq$" is not the same as "$\geqq$" which is used in works on partially ordered sets to stand for " > or = " . Cf. Birkhoff (1967) *Lattice theory* p.1.

† Except the scneme of Chartres (1966), who computes a non-void *subset* rather than a *superset* of the range of an expression.

$$\alpha^n \equiv \alpha\cdot\alpha\cdot\ldots\cdot\alpha \quad (n \text{ times})$$

could be used for intervals as well as extended reals, but a more useful definition for intervals is

$$A^n \equiv \bigcup \alpha^n \text{ over all } \alpha\epsilon A \ .$$

Now we find that $A^n \subseteq A\cdot A\cdot\ldots\cdot A$, with equality only in certain special cases like $A \geq 0$; and in an example used earlier we find that evaluating the expression

$$X\cdot\left(1 - 2/\left(1 + (X/Y)^2\right)\right)$$

for $X = Y = 0$ yields the desired limiting value 0 when $(X/Y)^2$ is evaluated as $\Omega^2 = [0,\infty]$ instead of just yielding Ω when Ω^2 is degraded to $\Omega\cdot\Omega = \Omega$.

More generally, consider any function $\phi(\xi_1;\xi_2;\ldots;\xi_n)$ defined over some domain in extended real n-space. If that domain includes the region $(X_1;X_2;\ldots;X_n)$, we shall represent the *range* of ϕ over that region by $R\phi(X_1;X_2;\ldots;X_n) \equiv \bigcup \phi(\xi_1;\xi_2;\ldots;\xi_n)$ over $\xi_i\epsilon X_i$. On the other hand, if ϕ is a rational function of its arguments, then it is representable over almost all of its domain by each of infinitely many rational expressions in those arguments; to each such rational expression $\phi(\xi_1;\xi_2;\ldots;\xi_n)$ corresponds an interval expression denoted by $\Phi(X_1;X_2;\ldots;X_n)$ and obtained from ϕ by formally substituting X_i in place of ξ_i . Evidently $R\phi(X_1;X_2;\ldots;X_n) \subseteq \Phi(X_1;X_2;\ldots;X_n)$.

A persistent technical problem in interval arithmetic is to find an expression for ϕ which will turn the last $\subseteq$ into an $=$. Sometimes this problem is soluble; for example consider $\phi(\xi) \equiv \xi/(\xi+2) = 1/(1 + 2/\xi)$ and observe that $R\phi([-1,1[) = \Phi([-1,1[)$ for the second expression but not for the first. (This example is worked out above.) The same is true for another example,

$$\phi(x_1;x_2) \equiv (x_1+x_2)/(x_1-x_2) = 1 - 2/(1 - (x_1/x_2)) \quad ,$$

which has been treated by Moore (1966, pp.28 and 45-7) in two other ways for both of which $R\phi \neq \Phi$ at some arguments X_1 and X_2 . Sometimes the exterior interval-numbers in our system permit us to find expressions ϕ for which $R\phi = \Phi$ more easily than in other forms of interval arithmetic. But in general the computation of $R\phi$ requires the location of maxima and minima, and hence the solution of polynomial equations when ϕ is rational, as well as the evaluation of limits or bounds for indefinite forms.

Interval arithmetic can be made easier via the provision of interval functions like those provided for ASA-standard FORTRAN (1964) , ALGOL 60 (1963), and Triplex ALGOL (Apostolatos et al. (1968)). In general, we want $R\phi(X;Y;...)$ for functions $\phi(\xi;\eta;...)$ like

$$\text{abs}(\xi) \ , \ \text{sign}(\xi) \ , \ \sin(\xi) \ , \ \text{sqrt}(\xi) \ , \ \exp(\xi) \ , \ \xi^{\eta} \ , \ldots \ .$$

Of course, attention must be paid to infinite values like

cotan(0) ≡ ∞ , indefinite forms like $0^0 \equiv]0,\infty[$, and undefined values like sqrt(-4) ; but in general the definitions of interval-functions like

*R*abs(*X*) , *R*sign(*X*) , *R*sin(*X*) , *R*sqrt(*X*) , *R*exp(*X*) , X^Y , ...

should be obvious. Provided those definitions are understood, the following theorem can be proved, generalizing a statement of Moore (1966, p.11) .

Theorem: If $f(x_1;x_2;\ldots;x_n)$ is an arithmetic expression in FORTRAN or an unconditional arithmetic expression in ALGOL, and if each variable x_i appears only once in that expression, then

$$Rf(X_1;X_2;\ldots;X_n) = F(X_1;X_2;\ldots;X_n)$$

for all intervals $X_1,\ldots,X_n$ contained in f's domain, except possibly when Ω appears during F's evaluation. Here F is the interval expression obtained from f by replacing each real variable x_i by the corresponding interval-number X_i and each real function by its corresponding interval-function.

Corollary: If also $f(x_1;x_2;\ldots x_n)$ is a monotonically non-decreasing function of x_k , x_{k+1} ,..., and x_n throughout $(X_1;X_2;\ldots;X_n)$ when $X_k = X_{k+1} = \ldots = X_n$, and we define expressions

$$g(x_1;x_2;\ldots;x_k) \equiv f(x_1;x_2;\ldots;x_k;x_k;\ldots;x_k) \ ,$$
$$G(X_1;X_2;\ldots;X_k) \equiv F(X_1;X_2;\ldots;X_k;X_k;\ldots;X_k) \ ,$$

then $Rg(X_1;X_2;...;X_k) = G(X_1;X_2;...;X_k)$, except possibly when Ω appears during G's evaluation.

Otherwise, if a function $g(x) \equiv f(x;x)$ is monotonically increasing although $f(x;y)$ is not, the best way to compute $Rg(X)$ for $X = [a,b]$ will be via $Rg(X) = [g(a) , g(b)] \subset G(X)$. Here are some examples to illustrate the theorem and its corollary. First let

$f(a;b;c;d;x;y;z) \equiv (ax^2 + by + c)/(d - z)$;

$Rf([0,1];[1,2];3;[4,5];[0,1];[0,1],[0,1]) = [3/5,2] = F(...)$.

Then let $g(a;b;c;d;x) \equiv f(a;b;c;d;x;x;x) = (ax^2 + bx + c)/(d-x)$;

$Rg([0,1];[1,2];3;[4,5];[0,1]) = [3/5,2] = G(...)$ too.

For our second example consider four expressions representing $g(x) \equiv g(1;-2;1;2;x)$, namely

$g_1(x) \equiv (x^2-2x+1)/(2-x)$, $g_2(x) \equiv (x(x-2)+1)/(2-x)$,

$g_3(x) \equiv (x-1)^2/(2-x)$, $g_4(x) \equiv 1/((1/(x-1)-1/2)^2 - 1/4)$;

and let $X \equiv [-1,1]$, $Y \equiv [1/2,3/2]$. We find

$G_2(X) = [-2,4] \supset G_1(X) = [-1,4] \supset G_3(X) = [0,4] \supset G_4(X) = [0,4/3]$
$= Rg(X)$,

$G_1(Y) = [-7/2,9/2] \supset G_2(Y) = [-5/2,3/2] \supset G_3(Y) = G_4(Y) = [0,1/2]$
$= Rg(Y)$.

(G_4 can be evaluated in our scheme, but not in anyone else's.)

More generally, let $F(X_1;...;X_n)$ be obtained by substituting X_i for x_i in some arithmetic expression

$f(x_1;\ldots;x_n)$ whose variables x_i may appear arbitrarily often. Sometimes the following theorems help to approximate Rf .

Theorem: If $X_i \subseteq Y_i$ for each $i = 1,2,\ldots,n$ then

$$Rf(X_1;X_2;\ldots;X_n) \subseteq Rf(Y_1;Y_2;\ldots;Y_n) \text{ and}$$

$$F(X_1;X_2;\ldots;X_n) \subseteq F(Y_1;Y_2;\ldots;Y_n) .$$

Theorem: If for each $i = 1,2,\ldots,n$ we have $X_i = \bigcup_j X_{ij}$, where the X_{ij} need not be disjoint, then

$$Rf(X_1;X_2;\ldots;X_n) \subseteq \bigcup_j F(X_{1j};X_{2j};\ldots;X_{nj}) \subseteq F(X_1;X_2;\ldots;X_n).$$

The difference between Rf and $\bigcup_j F$ can be made arbitrarily small when F is continuous throughout $(X_1;X_2;\ldots;X_n)$ by diminishing the sizes of the subintervals X_{ij} ; this can be proved with the aid of notions introduced in the next section. Here we have tried to convey some feeling for the combinatorial approach to the computation of Rf via symbolic rearrangement of expressions before evaluation. Moore (1966, ch.6) offers several other interesting ideas in this area, but the area remains largely *terra incognita*.

Further work is needed also on a problem peculiar to our scheme - the occasional intrusion of Ω . The appearance of this symbol during an expression's evaluation is usually symptomatic of a loss of information that can be recovered only by analytical means appropriate to real but not complex variables. For example consider $\phi(\xi;\eta) \equiv 1/(\xi^2 + \eta^2)$;

evidently $R\phi([1,\infty];[1,\infty]) = [0,1/2] \subsetneq \Phi([1,\infty];[1,\infty]) = \Omega$, whereas $R\phi([1,\infty[;[1,\infty[) =]0,1/2] = \Phi([1,\infty[;[1,\infty[)$. The difference between these two evaluations is related to the fact that $\phi(\xi;\eta)$ has a limit as $\xi \to \infty$ and $\eta \to \infty$ through real values, but not when ξ and η are allowed to pass through complex values. Resorting to open intervals is not always a cure, but often helps. At first sight one might be tempted to "cure" the problem by distinguishing among $+\infty$, $-\infty$ and ∞ ; but this distinction soon leads to further distinctions among $0+$, $0-$ and 0 as in

$$\lim_{x\to 0+} \exp(1/x) = +\infty \quad , \quad \lim_{x\to 0-} \exp(1/x) = 0+ \quad ,$$

from which distinctions follow others and yet other complications, even to the point of jeopardizing the last two theorems. For simplicity's sake we shall not discuss such a "cure" here (but cf. §5.*i*).

4. Metric Notions

To provide a context in which convergence, continuity and approximation can be discussed, we shall introduce *metrics* or *distance functions* $d(\xi;\eta)$. The discussion here is very superficial because we are merely generalizing slightly certain notions explored in detail by Moore (1966, ch.4).

A *metric* $d(\xi;\eta)$ is a real valued function satisfying the usual four rules (Dieudonné (1960, p.27))

$$0 \le d(\xi;\zeta) = d(\zeta;\xi) \le d(\xi;\eta) + d(\eta;\zeta)$$

and if $d(\xi;\zeta) = 0$ then $\xi = \zeta$.

The familiar choice $d(x;y) \equiv |x-y|$ is sometimes inconvenient for the extended reals*; instead we can define $d(\xi;\eta)$ by means of our identification of Ω with a circle. Let the circle be drawn on a plane or a sphere and let $d(\xi;\eta)$ be the distance between ξ and η measured perhaps along the circle, or across the surface, or through space. Distance functions $d(\xi;\eta)$ like these, which are continuous as $\xi \to \infty$ and $\eta \to \infty$ independently, are appropriate when convergence to ∞ is at issue. One suitable choice is the *chordal metric*

$$d(x;y) \equiv \frac{|x-y|}{\sqrt{(1+x^2)\,(1+y^2)}} \quad , \quad d(\xi;\infty) \equiv 1/\sqrt{1+\xi^2} \quad ,$$

for which $d(1/\xi;1/\zeta) = d(\xi;\zeta)$; cf. Carathéodory (1958, §86).

---

* Although this d can be imposed upon the extended reals by defining $d(x;\infty) \equiv +\infty$ and $d(\infty;\infty) = 0$, doing so requires that $+\infty$ be distinguished from $-\infty$ and ∞ (cf. sections 3 and 5.*i*) and consequently accentuates the topological distinction between ∞ and the reals. Also, the least inconvenient definition for the width of an interval turns out to be $w([a,b]) \equiv b - a$, which assigns useful negative widths to exterior intervals but useless infinite width to $[x,\infty]$.

We also need a *measure* for the width of an interval; one natural measure is

$$w(Z) \equiv \int d(\zeta;\zeta + d\zeta) \quad \text{over} \quad \zeta \in Z \quad .$$

For example, using the chordal metric we find that

$$w([x,y]) = \arctan y - \arctan x \quad \text{if} \quad x \le y \quad ,$$

$$= \pi - \arctan x + \arctan y \quad \text{if} \quad x > y \quad ,$$

$$w([x,\infty]) = \pi/2 - \arctan x \quad , \quad w([\infty;\infty]) = 0 \quad ,$$

$$w(\Omega) = \pi \quad .$$

And a natural* extension of any chosen $d(\xi;\zeta)$ to cover intervals is Hausdorff's (Dieudonné (1960, p.58, ex.3)) ;

$$d(\Xi;Z) \equiv \max\{ \sup_{\xi\in\Xi} \inf_{\zeta\in Z} d(\xi;\zeta) \quad , \quad \sup_{\zeta\in Z} \inf_{\xi\in\Xi} d(\xi;\zeta)\} \quad ,$$

which is easily computed using only the end-points of Ξ and Z and their types, interior or exterior. Since w and d make no distinction between open and closed intervals (e.g. $d(\Xi;Z) = 0$ implies only that Ξ and Z have the same closure), metric considerations are customarily confined to closed intervals.

Our definitions preserve many familiar theorems. For example:

---

* There are other natural extensions; see Eggleston (1958, p.60) or Rudin (1953, p.195). But Hausdorff's coincides with Moore's (1966, pp.15-16) when $d(x;y) \equiv |x-y|$ is extended to finite interior intervals, and preserves Moore's Lemmas 4.1 and 4.2.

If $A \subseteq B$ then $w(A) \le w(B)$,

and $d(\xi;A) \ge d(\xi;B)$ for all $\xi \notin A$,

and $w(B) \le w(A) + 2e$ implies $d(A;B) \le e$,

and $d(A;B) \le e$ implies $w(B) \le w(A) + \pi e$.

The constant π is appropriate for the chordal metric d ; more generally π should be replaced by

$$2/\inf_{\alpha \neq \beta} \max\{d(\alpha;\beta)/w([\alpha,\beta]) \ , \ d(\alpha;\beta)/w([\beta,\alpha])\} \ .$$

The foregoing definitions provide a terminology with which to discuss how well one interval-number approximates another, and to introduce an Interval Analysis analogous to Real Analysis with continuous or integrable functions. An interval-valued function $\Phi(Z_1;Z_2;\ldots)$ of interval-variables $Z_1, Z_2, \ldots$ is *continuous* at $(X_1;X_2;\ldots)$ in some domain whenever for every $\varepsilon > 0$ there is a $\delta > 0$ such that $d(\Phi(X_1;X_2;\ldots) \ ; \ \Phi(Y_1;Y_2;\ldots)) < \varepsilon$ for all $(Y_1;Y_2;\ldots)$ in that domain which satisfy $d(X_i;Y_i) < \delta$ for $i = 1,2,\ldots$. In particular, a *rational* interval-expression Φ (involving only the arithmetic operators $+$, $-$, $\cdot$, $/$ and n-th powers for integers $n \neq 0$) can fail to be continuous in a domain $(A_1;A_2;\ldots)$ in the chordal metric d only when Ω appears during the evaluation of $\Phi(A_1;A_2;\ldots)$. Consequently, many of the complications associated in §3 with the *extensions* of extended real functions $\phi(\zeta_1;\zeta_2;\ldots)$ to interval-functions $\Phi(Z_1;Z_2;\ldots)$ can be avoided, at least for the purposes of exposition, by limiting attention to the *restrictions* of

rational expressions Φ to rational functions $\phi(\zeta_1;\zeta_2;...) \equiv \Phi(\zeta_1;\zeta_2;...)$. The continuity of Φ is then sufficient (but not necessary) to assure the continuity of ϕ .

Consult the book *Interval Analysis* by R.E. Moore (1966, ch.4 and 6-9) for an extensive treatment of the subject.

5. Implementation Problems

These fall into four areas with which we shall deal in turn;

i) Representation,

ii) Approximation,

iii) Diagnostics,

iv) Compilability.

Instead of solutions to these problems, we offer suggestions and opinions.

i) Representation: The two binary digits required to indicate which of an interval-number's endpoints belong to it may be inconvenient to manipulate on some machines, in which case manipulation can be confined to the subset of closed interval numbers $[\alpha,\beta]$ without much loss of information. Some of this information is recoverable on most machines which represent numbers with a sign-magnitude format, because these machines usually preserve a distinction between $+0$ and -0 and between $+\infty$ and $-\infty$. Consequently, all pairs $\{\alpha,\beta\}$

can be identified with closed interval-numbers $[\alpha,\beta]$ except for the following eight reassignments;

$$[\alpha,+\infty] \equiv [\alpha,\infty] \quad , \quad [\alpha,-\infty] \equiv [\alpha,\infty[\quad , \quad [+\infty,\beta] \equiv]\infty,\beta] \quad ,$$
$$[-\infty,\beta] \equiv [\infty,\beta] \quad , \quad [\alpha,+0] \equiv [\alpha,0[\quad , \quad [\alpha,-0] \equiv [\alpha,0] \quad ,$$
$$[+0,\beta] \equiv [0,\beta] \quad , \quad [-0,\beta] \equiv]0,\beta] \quad .$$

Whenever an arithmetic operation involving one of these eight produces some other unclosed interval-number, that interval-number should be closed to cover its end-points; the consequent loss of information will be no worse than is attributable to roundoff. (See below under *Approximation*.) The eight reassignments sometimes help programmers to suppress Ω .

The symbols Ω and ∞ can be represented on most machines by certain unnormalized floating point zeros or by some other improper floating point numbers. Care should be taken not to represent ∞ in a way which might be confused with an overflow. (See below under *ii* and *iii* .)

Occasionally one may prefer to represent an interval-number $[a,b]$ by some pair of numbers other than a and b ; a plausible choice is $(a+b)/2$ and $(b-a)/2$ (cf. Nickel (1966), Chartres (1966), Dwyer (1941, ch.2)), corresponding to an *approximator* and its *uncertainty* respectively. What motivates such a choice is that the uncertainty is expected to amount at most to a tiny fraction of the approximator and therefore can be represented with a lower relative precision without appreciably degrading the scheme; therefore computer storage can be

saved by assigning shorter words to uncertainties than to the more precise approximators. However, to distinguish between $[10^{-30},10^{30}]$ and $[0,10^{30}]$ both approximator and uncertainty must be represented equally precisely, and there are applications of Interval Arithmetic where that distinction is important. These applications concern the estimation of the range of a function representing, say, some engineering design that is intended to perform correctly in a wide range of environments. The environments are represented by interval-numbers given as data; the performance will be encompassed within the interval-numbers produced by the computation. The widths of the intervals may well be substantial; the ratios *uncertainty/approximator* are of far less concern than that the intervals be not much wider than necessary. My interest in these applications is such that I prefer to represent $[a,b]$ via the pair $\{a,b\}$ rather than via $\{(a+b)/2\ ,\ (b-a)/2\}$.

ii) Approximation: Roundoff need not vitiate the definitions given in §2 of the arithmetic operators $+$, $-$, $\cdot$ and $/$ provided they are approximated in a way which is interpretable as a loss of precision or of information rather than as a source of misinformation. The appropriate way is via what we shall call *outer* approximation.

Just as the real numbers normally representable in a computer constitute a subset of the rationals, so must the interval-numbers normally representable in a computer constitute

a subset; call them the *storable* interval-numbers. We shall call a storable interval-number C' an *outer approximation* to C whenever $C' \supseteq C$ and $d(C';C)$ is sufficiently small. How small is sufficiently small, and which metric is d, are important questions which will not be discussed here. Interval Arithmetic is properly implemented on a computer when, for every arithmetic operator $*$ in $\{+ , - , \cdot , / \}$, every instruction-sequence intended to compute $C = A*B$ produces at worst an outer approximation C'; similar statements should be applicable to all the elementary functions like $R\exp$, $R\sin$, A^B, ... which are provided in that implementation. When properly implemented, Interval Arithmetic will lose information to the extent that its outer approximations are too big, and to that extent may generate excessive pessimism, but cannot generate misinformation.

The associative law is an inevitable casualty of roundoff since, for example, $10^{-35} + (10^{35} - 10^{35})$ produces 10^{-35} whereas $(10^{-35} + 10^{35}) - 10^{35}$ produces 0 in ordinary arithmetic with fewer than 71 decimals. Ideally commutativity, monotonicity and sign-symmetry should be preserved wherever appropriate; this will be so when every operator is approximated ideally, the ideal outer approximation C' to C being the narrowest storable interval of the same type (interior, exterior, open, closed) as C which contains C. Current floating point hardware design does not always help the implementor achieve the ideal. Rarely can he avoid approximating $2 + 2$ by [3.999...99 , 4.000...01] without

programming a host of tedious tests to ensure that $1 + [0,10^{-39}]$ is properly approximated by $[1 , 1.000...01]$ and $1 - [0 , 10^{-39}]$ by $[0.999...99 , 1]$. I think an ideal implementation is worth whatever it costs.

Arithmetic expressions which underflow or overflow can be approximated with the aid of 0 or ∞ respectively. For example, if 10^{-100} and $0.999...99\times10^{99}$ are the smallest and largest real numbers normally representable in the machine, then 10^{-200} might be approximated by $]0 , 10^{-100}[$ and 10^{200} by $]0.999...99\times10^{99} , \infty [$. Thus would overflow join division by zero as the only ways to generate ∞ in our scheme. Some different ways to treat underflow and overflow usefully have been described by Kahan (1966, pp.26-51).

iii) Diagnostics: The appearance of Ω during a calculation is usually but not always symptomatic of a mistake. Every implementation of Interval Arithmetic should permit a program to test whether Ω has appeared recently and to respond in whatever way the programmer has provided. In default of such a provision, the program's execution should be interrupted, if not suspended, as soon as Ω appears, and information should be printed out to help the programmer discover why Ω appeared. The programmer's response to that information will be either to identify and correct a mistake, or to recognize a function whose evaluation requires further analysis at some critical points. Sometimes the simplest way to estimate the range of

a complicated function is to compute several formally equivalent interval expressions of that function and then select the narrowest. Therefore the appearance of Ω is not always a disaster.

Similar considerations apply to overflow, underflow and the appearance of ∞ ; fortunately these events can have serious consequences only if they later cause an Ω to appear, and that Ω will not go unnoticed. The main reason for interrupting a program's execution (only if the programmer has asked for such interruptions) in response to such events is that these events are often followed by Ω's whose causes might otherwise remain obscure.

There are two classes of systems programmers to whom the implementation of Interval Arithmetic should not be entrusted; those whose rigid moralities exclude any tolerance for other men's mistakes, and those who indulgently make provision for every possible vice. The author's *Tao* to Enlightenment through Hindsight, which uses post-execution reminders, simple options, and messages in English or FORTRAN but not Octal, is described in Kahan (1966).

iv) Compilability: Interval Arithmetic is more aptly to be regarded as supplementing than supplanting ordinary real arithmetic. This point of view is supported by the excellent results which Hansen (1968 and references cited therein) has obtained; he uses Interval Arithmetic to refine ordinary

arithmetic calculations and guarantee their validity. K. Nickel, N. Apostolatos et. al. (1967) have gone so far as to propose an extension of ALGOL 60 to cover their brand, called Triplex-ALGOL 60, of Interval Arithmetic. We propose here to outline a comparable extension of ASA standard FORTRAN.

Interval-numbers can be represented by pairs of real numbers (see *i* above) just like complex numbers, so adding a *type* INTERVAL need not complicate the indexing or input/output facilities of the compiler. Scanning INTERVAL-arithmetic statements should be no more complicated than scanning DOUBLE PRECISION or COMPLEX statements since the latter two types involve subroutines for at least some of their elementary arithmetic operations (certainly for COMPLEX multiplication and division) whereas INTERVAL arithmetic uses subroutines for all operations. Mixing REAL and INTERVAL arithmetic is just like mixing REAL and COMPLEX arithmetic. The relational operators .GT. , .GE. , .EQ. , ... (for $>$, $\geq$, $=$,...) will have to call subroutines if they are allowed to appear between INTERVAL expressions*. Transfer functions analogous to REAL, AIMAG and COMPLEX will be needed to facilitate ordinary arithmetic with the end-points of INTERVAL variables, and other

---

* I cannot understand why ASA standard FORTRAN (1964, p.598) forbids .EQ. to appear between two COMPLEX expressions, nor why the assignment COMPLEX = REAL is forbidden (ibid., p.600). Slips like these give FORTRAN a bad reputation.

subroutines will be needed for .INSIDE. , UNION (of two or more overlapping intervals), RABS, REXP, RLOG and similar functions. For each of a few plausible choices $d(A;B)$, it will be necessary to provide DIST and WIDTH functions comparable to CABS.

Interval Arithmetic will remain unknown to most of its potential beneficiaries until it is comfortably embedded in some of the widely used algorithmic languages. Interval Arithmetic's full potential will remain unknown to all of us until it is embedded in a language which, like FORMAC (see Tobey et al. (1967)), offers both symbolic and numerical arithmetic capabilities, because the outstanding problems of Interval Arithmetic are more mathematical (algebra, analysis and geometry) than computational.

6. Applications

Interval Arithmetic's most obvious application is to those numerical problems whose solutions can be implemented in a computer program with no iterations nor oft-repeated loops. Examples include the computation of engineering design parameters and performance from cook-book formulae, the fitting of simple curves to modest numbers of observations, and the transformation of geometrical information from one coordinate system into another. These problems have solutions which may

be identified with the computation of several functions $f_i(x_1;x_2;...;x_n)$ of a modest number n of variables x_j. The people who wish to solve such problems may be expert enough in their chosen fields, but are usually unacquainted with the tricks of error analysis, and therefore unable to assess the accuracy of their computations even when they want to. By using Interval Arithmetic to compute almost any naïve expression for $F_i(X_1;X_2;...;X_n)$ they may be sure that no numerical instability can mislead them. Narrow intervals F_i are acceptable without reservation. If the computed intervals F_i are too wide, there are two possible explanations. First, the width may be due merely to an expression for F_i which is too naïve, corresponding to what might otherwise be called "a numerically unstable calculation"; the remedy here is found by consulting a numerical analyst. Secondly, the width may reflect the fact that some f_i are discontinuous or at least violently varying functions of some x_j; such behaviour is symptomatic of an ill-posed problem. In other words, if wide intervals occur they signify a need for more analysis; if no wide intervals occur then we are all, experts and novices alike, relieved of tedious and superfluous analysis. That is what machines are for.

If the expressions for F_i are aptly chosen (apt choices are sometimes not obvious, sometimes impossible) then they may be used to study the consequences of varying various input parameters x_j across suitable intervals X_j, as was mentioned

in §5.*i*. When combined with interactive computing facilities, this application of Interval Arithmetic can significantly shorten the search for flaws in engineering designs.

Interval Arithmetic is also useful in conjunction with ordinary arithmetic for solving a set of n equations $f_i(x_1;x_2;\ldots;x_n) = 0$, not so much for finding a solution as for proving that a solution has been found. We shall illustrate this point by describing a relatively simple technique; better techniques are described by Hansen (1968).

Let us write $\underline{f}(\underline{x})$ for the column vector whose n components are $f_i(x_1;x_2;\ldots;x_n)$, and $\underline{f}_x$ for the Jacobian matrix of $\underline{f}$'s first partial derivatives. We assume that $\underline{f}(\underline{x})$ and $\underline{f}_x(\underline{x})$ are represented by real expressions to which correspond continuous interval expressions $\underline{F}(\underline{X})$ and $\underline{F}_x(\underline{X})$. We also assume that some approximation $\underline{y}_0$ is given and known to be "fairly close" to the true solution $\underline{z}$ of $\underline{f}(\underline{z}) = \underline{0}$. The computation proceeds in two phases. First is the improvement of $\underline{y}_0$ by Newton's iteration, ideally

$$\underline{y}_{n+1} = \underline{y}_n - \underline{f}_x(\underline{y}_n)^{-1} \cdot \underline{f}(\underline{y}_n) \quad ,$$

in which interval arithmetic is used only to help decide when to stop the iteration. The second phase uses interval arithmetic in an essential way to bound the error in the last iterate. Because Newton's iteration converges so rapidly (quadratically) in the absence of pathology, we shall attempt to approximate $\underline{z}$ as accurately as roundoff permits even though that accuracy may

exceed our needs.

Given $\underline{y}_n$ compute, as accurately as roundoff permits, the values of $\underline{f}(\underline{y}_n)$, $\underline{f}_x(\underline{y}_n)$ and a matrix $\underline{G}_n$ to approximate $\underline{f}_x(\underline{y}_n)^{-1}$; no interval arithmetic is needed here. Also let $\underline{Y}_n$ be an n-vector of intervals obtained from $\underline{y}_n$ by smearing each element of $\underline{y}_n$ two units in its last place: $\underline{Y}_n$ is the narrowest storable interval-vector containing $\underline{y}_n$ in its interior. Compute $\underline{F}(\underline{Y}_n)$ to obtain a bound for the variation of $\underline{f}(\underline{y}_n)$ attributable to uncertainty due to roundoff plus small perturbations in $\underline{y}_n$. Sometimes that bound can be computed more accurately and/or efficiently by means other than Interval Arithmetic; e.g. see Kahan and Farkas (1963), Smith (1967a, pp.70-90, or 1967b), or Adams (1967) if $\underline{f}$'s components are all polynomials. The final result's accuracy depends crucially upon how precisely $\underline{f}(\underline{y}_n)$ and $R\underline{f}(\underline{Y}_n)$ can be estimated.

Normally the iteration would proceed to $\underline{y}_{n+1} \equiv \underline{y}_n - \underline{G}_n \cdot \underline{f}_n$ where $\underline{f}_n$ is the computed approximation to $\underline{f}(\underline{y}_n)$. However, the iteration ought to be stopped when $\underline{y}_n$ is as close to $\underline{z}$ as $\underline{y}_{n+1}$ is likely to be. We choose to stop as soon as $\underline{0} \in \underline{F}(\underline{Y}_n)$; when this criterion is satisfied there is practically no way to distinguish $\underline{y}_n$ from $\underline{z}$. (Note that that criterion is certainly satisfied when $\underline{z} \in \underline{Y}_n$, but does not imply $\underline{z} \in \underline{Y}_n$.) Will the criterion ever be satisfied? In general this is a difficult question to answer precisely; the answer turns out

to be *Yes* provided $\underline{y}_0$ lies within a neighbourhood of $\underline{z}$ wherein $\underline{f}_x(\underline{x})$ varies not too widely and is not too ill-conditioned.

Having stopped at $\underline{y} \equiv \underline{y}_n$, we enter the second phase of the computation to provide a bound for $\underline{z} - \underline{y}$. We do so with the aid of what is intended to be a *contraction mapping* $\underline{h}(\underline{x})$ when $\underline{x}$ lies in the neighbourhood of $\underline{z}$ and $\underline{y}$ (cf. Collatz (1966, p.213)). Let

$$\underline{h}(\underline{x}) \equiv \underline{x} - \underline{G}\cdot\underline{f}(\underline{x})$$

where $\underline{G} \equiv \underline{G}_n$ is the best available approximation to $\underline{f}_x(\underline{y})^{-1}$. We shall verify later (next footnote) whether $\underline{G}$ is nonsingular; if so, each of $\underline{h}$'s fixed points $\underline{z} = \underline{h}(\underline{z})$ is a solution of $\underline{f}(\underline{z}) = \underline{0}$. Since $\underline{h}$ is continuous, any interval-vector $\underline{Z}$ which contains $R\underline{h}(\underline{Z})$ must contain at least one of $\underline{h}$'s fixed points (*ibid.* pp.450-6). Our task now is to exhibit such a $\underline{Z}$, and preferably a narrow one.

Let us first apply the mean value theorem to write

$$\underline{h}(\underline{x}) = \underline{h}(\underline{y}) + (\underline{I} - \underline{Q}(\underline{x})) \cdot (\underline{x} - \underline{y})$$

where $\underline{I}$ is the $n\times n$ identity matrix and $\underline{Q}(\underline{x})$ is the matrix whose elements are

$$q_{ij}(\underline{x}) \equiv \textstyle\sum_k g_{ik} \dfrac{\partial}{\partial x_j} f_k(\underline{y} + (\underline{x} - \underline{y})\theta_k)$$

for some unknown $\theta_k(\underline{x}) \epsilon]0,1[$. As $\underline{x} \to \underline{y}$, $\underline{Q}(\underline{x}) \to \underline{G}\cdot\underline{f}_x(\underline{y})$ and, since $\underline{G} \doteq \underline{f}_x(\underline{y})^{-1}$, we should expect $\underline{I} - \underline{Q}(\underline{x})$ to be small of the order of roundoff plus $O(\underline{x} - \underline{y})$.

This expectation can be put to the test over any interval $\underline{Z}$ containing $\underline{y}$; use Interval Arithmetic to compute the interval matrix

$$\underline{S}(\underline{Z}) \equiv \underline{I} - \underline{G}\cdot\underline{F}_x(\underline{Z}) \supseteq R(I - \underline{Q}(\underline{Z})) \quad .$$

As long as $\|\underline{S}(\underline{Z})\| \le \sigma$ for some small $\sigma < 1$ we may be sure* that $\underline{h}(\underline{x})$ is a contraction mapping over $\underline{x}\in\underline{Z}$; in fact we find that $\underline{h}(\underline{x}_1) - \underline{h}(\underline{x}_2) \in \underline{S}(\underline{Z})\cdot(\underline{x}_1 - \underline{x}_2)$, so $\|\underline{h}(\underline{x}_1) - \underline{h}(\underline{x}_2)\| \le \sigma\|\underline{x}_1 - \underline{x}_2\|$ for every $\underline{x}_1$ and $\underline{x}_2$ in $\underline{Z}$. Provided σ is small enough (depending upon the norm used) the interval-arithmetic analogue of $\underline{h}(\underline{x})$, namely

$$\underline{H}(\underline{X}) \equiv \underline{y} - \left(\underline{G}\cdot\underline{F}(\underline{y}) - \underline{S}(\underline{X})\cdot(\underline{X} - \underline{y})\right) \quad ,$$

will also contract the width of each $\underline{X} \subseteq \underline{Z}$. Unfortunately, $\underline{H}(\underline{X})$ cannot be proved to lie in $\underline{Z}$ without assuming more than has been assumed so far.

---

* Almost any matrix norm, say $\|\{s_{ij}\}\| \equiv \max_i \sum_j |s_{ij}|$, will serve adequately unless the equations $\underline{f}(\underline{x}) = \underline{0}$ are "ill equilibrated", which possibility will not be considered here. Every real matrix $\underline{A}\in\underline{F}_x(\underline{Z})$ satisfies $\|\underline{I} - \underline{G}\cdot\underline{A}\| \le \sigma < 1$ if $\|\underline{S}(\underline{Z})\| \le \sigma < 1$, whence it soon follows that $\|\underline{G}^{-1} - \underline{A}\| \le \sigma\|\underline{G}^{-1}\| \le \sigma\|\underline{A}\|/(1 - \sigma)$ and $\|\underline{G} - \underline{A}^{-1}\| \le \sigma\|\underline{A}^{-1}\| \le \sigma\|\underline{G}\|/(1 - \sigma)$. Therefore $\underline{G}$ is nonsingular, and so $\underline{h}$'s fixed points $\underline{z}$ really do satisfy $\underline{f}(\underline{z}) = \underline{0}$.

After recalling that $\underline{y} \equiv \underline{y}_n \in \underline{Y} \equiv \underline{Y}_n$ and that $\underline{0} \in \underline{F}(\underline{Y}) \supseteq \underline{F}(\underline{y})$, let us choose $\underline{Z}_0 \equiv \underline{y} - \underline{G}\cdot\underline{F}(\underline{Y})$ and compute $\underline{Z}_1 \equiv \underline{H}(\underline{Z}_0)$. Evidently $\underline{y} - \underline{G}\cdot\underline{F}(\underline{y}) \in \underline{Z}_0 \cap \underline{Z}_1$ because $\underline{0} \in \underline{S}(\underline{Z}_0)\cdot(\underline{Z}_0 - \underline{y}) = -\underline{S}(\underline{Z}_0)\cdot\underline{G}\cdot\underline{F}(\underline{Y})$, so $\underline{Z}_1$ has some points in common with $\underline{Z}_0$. Moreover, $\underline{Z}_1$ can extend beyond $\underline{Z}_0$ only when $\underline{S}(\underline{Z}_0)\cdot(\underline{Z}_0 - \underline{y})$, which should be much smaller than $(\underline{Z}_0 - \underline{y})$, is wider than the gap between $\underline{Z}_0 = \underline{y} - \underline{G}\cdot\underline{F}(\underline{Y})$ and $\underline{y} - \underline{G}\cdot\underline{F}(\underline{y})$. Provided $\underline{F}(\underline{y})$ is sufficiently small, no such extension will occur, and we shall have $\underline{Z}_0 \supseteq \underline{Z}_1$, so $\underline{z} \in \underline{Z}_1$. However, if $\underline{Z}_0 \not\supseteq \underline{Z}_1$ then we can replace $\underline{Z}_0$ by $\underline{Z}_0 \cup \underline{Z}_1$ and test again whether $\underline{Z}_1 \equiv \underline{H}(\underline{Z}_0) \subseteq \underline{Z}_0$; this usually works.

Once we find some $\underline{Z} \supseteq \underline{H}(\underline{Z})$, we have proved that $\underline{z} = \underline{h}(\underline{z})$ for some $\underline{z} \in R\underline{h}(\underline{Z}) \subseteq \underline{H}(\underline{Z})$. Unfortunately there is no guarantee that such a $\underline{Z}$ can be found. For example, if $f(x) \equiv \cos(\exp(-1/x^2)) - 1$ but $w(F(x)) \geq 10^{-10}$ for all x, corresponding to calculations with ten decimals, then no mechanical way exists to decide whether $f(x)$ vanishes or not in $[-1/4\ ,\ 1/4]$. Most of the complications in the discussion above are caused by not knowing whether $\underline{f}(\underline{x})$ has a zero near $\underline{y}$ or not. Hansen (1968) assumes that an interval is known which contains a zero of $\underline{f}$, and consequently his arguments are simpler than ours.

Here is an example to illustrate what usually happens. The example is taken from Moore (1966, p.68).

$$\underline{f}(\underline{x}) \equiv \begin{pmatrix} x_1^2 + (x_2^2 - 1) \\ x_1 - x_2 \end{pmatrix}, \quad \underline{f}_{\underline{x}}(\underline{x}) = \begin{pmatrix} 2x_1 & 2x_2 \\ 1 & -1 \end{pmatrix}.$$

Computation is done in four-decimal floating point starting with $\underline{y}_0 = \begin{pmatrix} 0.7 \\ 0.7 \end{pmatrix}$. We find $\underline{f}(\underline{y}_0) = \begin{pmatrix} -.02 \\ 0 \end{pmatrix}$,

$$\underline{0} \notin \underline{F}(\underline{Y}_0) = \begin{pmatrix} [-.0204 & , & -.0196] \\ [-2\times10^{-4} & , & 2\times10^{-4}] \end{pmatrix}, \quad \underline{G}_0 = \begin{pmatrix} .3571 & .5 \\ .3571 & -.5 \end{pmatrix}.$$

$$\underline{y}_1 = \underline{y}_0 - \underline{G}_0 \cdot \underline{f}(\underline{y}_0) = \begin{pmatrix} .7071 \\ .7071 \end{pmatrix}, \quad \underline{0} \in \underline{F}(\underline{Y}_1) = 10^{-4} \begin{pmatrix} [-4 & , & 4] \\ [-2 & , & 2] \end{pmatrix}.$$

$$\underline{y} = \underline{y}_1, \quad \underline{F}(\underline{y}) = 10^{-4} \begin{pmatrix} [-2 & , & 0] \\ & 0 & \end{pmatrix}, \quad \underline{G} = \begin{pmatrix} .3536 & .5 \\ .3536 & -.5 \end{pmatrix},$$

$$\underline{Z}_0 = \begin{pmatrix} [.7068 & , & .7074] \\ [.7068 & , & .7074] \end{pmatrix},$$

$$\underline{S}(\underline{Z}_0) = \underline{I} - \underline{G} \cdot \underline{F}_x(\underline{Z}_0) = 10^{-4} \begin{pmatrix} [-10 & , & 4] & [-4 & , & 4] \\ [-4 & , & 4] & [-10 & , & 4] \end{pmatrix}.$$

$$\underline{Z}_1 = \underline{H}(\underline{Z}_0) = \underline{y} + 10^{-4} \begin{pmatrix} [0 & , & .7072] \\ [0 & , & .7072] \end{pmatrix} + 10^{-8} \begin{pmatrix} [-42 & , & 42] \\ [-42 & , & 42] \end{pmatrix} \subseteq \underline{Z}_0.$$

Therefore $\underline{z}$ exists and lies in $\begin{pmatrix} [.7071 & , & .7072] \\ [.7071 & , & .7072] \end{pmatrix}$.

7. Application to a Differential Equation

We seek to calculate the solution $y(t)$ of

$$\dot{y} \equiv \frac{d}{dt}\, y(t) = f(t;y(t)) \quad , \quad y(t_0) = y_0 \ ,$$

given numerical values for t_0 and y_0 and an expression $f(t;y)$ amenable to symbolic partial differentiation. A problem of long standing has been to compute rigorous bounds for the error in the approximation to $y(t_1)$ when t_1 is substantially greater than t_0 . The outstanding contribution made by Moore (1965a and b, 1966) to the resolution of this problem with the aid of Interval Arithmetic is perhaps the most potent reason for the current interest in both Interval Arithmetic and computable error bounds.

Here we shall attack the problem via a classical differential inequality (see Birkhoff and Rota (1959, §11) or Szarski (1965, p.7));

if $z(t_0) \geq y_0$ and $\dot{z}(t) \geq f(t;z(t))$ for $t \geq t_0$

then $z(t) \geq y(t)$ too.

Our method, chosen for simplicity, is quite different from Moore's.

Given $z_0 \geq y_0$, we attempt to choose storable numbers $\dot{z}_0$, $\ddot{z}_0$ and $\dddot{z}_0$ such that

$$z(t) \equiv z_0 + (t - t_0)\dot{z}_0 + (t - t_0)^2\ddot{z}_0/2 + (t-t_0)^3\dddot{z}_0/6$$

satisfies $\dot{z}(t) \geq f(t;z(t))$ over $t_0 \leq t \leq t_0+h$ for some $h > 0$. Success will yield an upper bound $z(t) \geq y(t)$ for

$t \in [t_0, t_0+h]$. An independent lower bound is to be computed in a similar way. Then t_0+h is renamed t_0 and we continue the advance towards t_1 . If* we reach t_1 , we shall have rigorous upper and lower bounds for $y(t_1)$.

Let us use abbreviations $\tau \equiv t - t_0$ and $g(t) \equiv f(t;z(t))$. Despite that g depends upon values $\dot{z}_0$, $\ddot{z}_0$ and $\dddot{z}_0$ which are not yet known, g is a symbolically differentiable expression;

$$\dot{g} = f_t + f_y \dot{z} \quad , \quad (f_y \equiv \frac{\partial}{\partial y} f(t;y)\Big|_{y=z(t)} \quad , \text{ etc. })$$

$$\ddot{g} = f_{tt} + 2f_{ty} \dot{z} + f_{yy} \dot{z}^2 + f_y \ddot{z} \quad ,$$

$$\dddot{g} = f_{ttt} + 3f_{tty} \dot{z} + 3f_{tyy} \dot{z}^2 + f_{yyy} \dot{z}^3 + 3(f_{ty}+f_{yy} \dot{z}) \ddot{z} + f_y \dddot{z}_0 \quad .$$

We note that numerical values will be computable for $\dot{g}_0 \equiv \dot{g}(t_0)$, $\ddot{g}_0 \equiv \ddot{g}(t_0)$ and the function $\dddot{g}(t)$ as soon as values have been assigned to $\dot{z}_0$, $\ddot{z}_0$ and $\dddot{z}_0$ respectively, but $g_0 \equiv f(t_0;z_0)$ is known now.

Next let us examine $s(t) \equiv z(t) - f(t;z(t))$; we find that

$$s(t) = \dot{z}(t_0 + \tau) - g(t_0 + \tau)$$
$$= (\dot{z}_0 - g_0) + \tau(\ddot{z}_0 - \dot{g}_0) + \tau^2(\dddot{z}_0 - \ddot{g}_0)/2 + \tau^3 \dddot{g}(t_0 + \tau\theta)/6$$

---

* There is some risk that the differential equation may possess a singularity which might intervene before t_1 is reached, but this risk is common to all numerical methods and will not be considered here.

for some unknown $\theta(t) \in]0,1[$. Our task is so to choose $h > 0$, $\dot{z}_0$, $\ddot{z}_0$ and $\dddot{z}_0$ that $Rs([t_0,t_0+h] \geq 0$, in which case we shall have $z(t) \geq y(t)$ in $[t_0,t_0+h]$. And if Rs is not too big, $z(t)$ should not be too much bigger than $y(t)$.

The first step is to choose a tolerance $e > 0$; our intention is to keep $Rs \subseteq [0,2e]$, so e should be chosen small to produce a tight upper bound z . On the other hand, the smaller is e , the smaller must h be kept to keep $Rs \subseteq [0,2e]$, and hence the greater must be the time required for the computation to reach $t = t_1$. If e is too small, roundoff alone may force $h = 0$; therefore e should always substantially exceed the uncertainty in $f(t;z)$ contributed by roundoff alone. Certainly e must exceed $w(F(T_0;Z_0))$, where F is the interval-arithmetic expression for f and T_0 and Z_0 are the narrowest interval numbers containing respectively t_0 and z_0 in their interiors. Increased values of e will permit increased values of h roughly proportional to $e^{1/3}$ while e is still small. There is no simple way to choose e optimally in general, but an adequate choice is hardly ever difficult, and given that h_{min} is the minimal acceptable value for h we may set $e = w(F([t_0,t_0+h_{min}] ; Z_0))$ as a last resort.

The choice of e can affect the precision and the cost of our bounds, but not their validity.

Having chosen e , compute in turn

$$\dot{z}_0 \doteq g_0 + e \quad , \quad \ddot{z}_0 \doteq \dot{g}_0 \quad \text{and} \quad \dddot{z}_0 \doteq \ddot{g}_0 \quad ;$$

the approximations here are due solely to roundoff during ordinary arithmetic evaluations of the formulas above for g , $\dot{g}$ and $\ddot{g}$. The symbols z_0 , $\dot{z}_0$, $\ddot{z}_0$, $\dddot{z}_0$ stand for numbers represented precisely in storage, and they define our upper bound $z(t)$ precisely. Certainly $z(t) \geq y(t)$ in some t- interval $[t_0, t_0+h]$ because $s(t) \geq 0$ in that t-interval provided h is small enough. Our next step is to find out in how wide an interval $[t_0, t_0+h]$ the condition $0 \leq s(t) \leq 2e$ remains valid, though we do so by using Interval Arithmetic to over-estimate $Rs([t_0, t_0+h])$.

Use interval-arithmetic expressions for g and its derivatives to compute the interval-numbers

$$G_0 \equiv G(t_0) \quad , \quad \dot{G}_0 \equiv \dot{G}(t_0) \quad \text{and} \quad \ddot{G}_0 \equiv \ddot{G}(t_0)$$

from the formulae given above. The widths of these interval-numbers should be of the order of roundoff because there is no other reason for their uncertainty, and of course we should find that the interval-numbers

$$\dot{z}_0 - G_0 - e \quad , \quad \ddot{z}_0 - \dot{G}_0 \quad , \quad \dddot{z}_0 - \ddot{G}_0$$

are all very tiny. We shall also need an interval-expression for $\dddot{G}(t_0+H)$ where H is an interval-number of the form $H = [0,h]$. Such an expression is provided by the formula $\dddot{g}$, though not uniquely; sometimes adequately large steps h

can be taken only after an expression for $\dddot{g}$ has been rearranged to yield a $\ddot{G}(t_0+H)$ which is not orders of magnitude wider than $R\ddot{g}(t_0+H)$. Finally assemble

$$S(t_0+H) \equiv \left(\left(H \cdot \ddot{G}(t_0+H)/6 + (\dddot{z}_0 - \ddot{G}_0)/2 \right) \cdot H + (\ddot{z}_0 - \dot{G}_0) \right) \cdot H + (\dot{z}_0 - G_0) \quad ,$$

of which we may be sure that $Rs(t_0+H) \subseteq S(t_0+H)$.

For any $H \doteq [0,h] \geq 0$ there are now three possibilities:

i) $S(t_0+H) \subset [0,2e]$ and $w(S)/e$ is not very tiny; such an H provides an acceptable step from t_0 to t_0+h .

ii) $S(t_0+H) \subseteq [0,2e]$ but $w(S)/e << 1$; such an H is too narrow, and might profitably be replaced by, say,

$$\left(0.7\ e/w(S)\right)^{1/3} \cdot H \quad .$$

iii) $S(t_0+H) \not\subseteq [0,2e]$; such an H is too wide and should be cut down to, say, $\left(0.7e/w(s)\right)^{1/3} \cdot H$.

Provided $e > w(G_0)$, $H = [0,0]$ is too narrow; $H = [0,\infty[$ is too wide except in trivial cases. By virtue of S's inclusion-monotonicity and continuity near $H = [0,0]$, some acceptable H must exist. A plausible first guess at H is whatever step was used to reach t_0 ; another plausible guess is $h \doteq |e/\dddot{g}(t_0)|^{1/3}$. The precise manner by which an acceptable H is found cannot be a vital issue first because H does not have to be chosen accurately (just not too wide!) and second because any unacceptable guess can be improved via ii) and iii) above. *The way H is found may affect the cost of our computed bounds, but not their precision nor validity.*

How precise are the bounds? We consider a hypothetical example drawn from Moore (1966, p.126), who drew it from Henrici (1962, p.85-6). Suppose $f(t;y) \equiv -16ty$, $t_0 \equiv -.75$, $t_1 \equiv +.75$, $y_0 > 0$. We choose a tolerance $e(t;y) \doteq 16\,\varepsilon\,y$ with some ε like 10^{-7} for an IBM 7094, on which each arithmetic operation is accurate to about 8 decimal digits. Because $|t| < 1$ for this problem, that tolerance e substantially exceeds the uncertainty introduced into f by roundoff. We assume too that z_0 is stored to double-precision, and that $z(t_0+h)$ is computed by first calculating $h(\dot{z}_0 + h(\ddot{z}_0 + h\dddot{z}_0/3)/2)$ in single precision with forced upward rounding of each arithmetic operation, and secondly adding that result to z_0 in double precision with an upward rounding. Wherever z_0 has been used above to compute $\dot{z}_0$, $\ddot{z}_0$ and $\dddot{z}_0$, the value of z_0 rounded to single precision may be used instead. Wherever z_0 appears during the interval-arithmetic calculation of $S(t_0+H)$ it should be replaced by the narrowest single-precision interval-number Z_0 containing z_0 . Without this appeal to one double-precise addition, the computed values of $z(t)$ would drift up excessively by one unit in the last place after each t-step h , and several thousand steps could be taken. (Actually, a few ~~hundred~~ thousand steps are sufficient.)

The computed values of $z(t)$ can now be approximated adequately by solving the differential inequality

$$0 \le \dot{z} - f(t;z) \le 2e \quad , \quad z(t_0) = z_0 \quad ,$$

in closed form with $f(t;z) = -16tz$ and $e = 16\,\varepsilon\,z$ and $z_0 = (1+\delta)y_0$ for some positive $\delta < 2\times10^{-8}$. We find that

$$1+\delta \le z(t)/y(t) \le (1+\delta)\exp\left(16\,\varepsilon\,(t-t_0)\right) \quad ,$$

which shows that $z(t)$ approximates $y(t)$ to within a few hundred units in y's last (8th) place. Since $y(t) = c\cdot\exp(-8t^2)$ for some constant $c > 0$, we observe that the width of the interval separating y from z will decrease as t increases toward $t_1 = +.75$. This observation contrasts strongly with the results produced by Moore's first method (1965a, and 1966, ch.10-12) because the widths of his interval-estimates for y cannot decrease (*ibid.* p.132).

The method described above is capable of generalization; we could use other functions $z(t)$ than cubic polynomials, other tolerances e than stepwise constants. But the crucial generalization to systems of differential equations is a step beyond the scope of these notes. Some idea of the rôle played by geometry in such a generalization can be inferred from the work of Moore (1965b, and 1966, ch.13), Guderley and Valentine (1967), and Kahan (1966', 1967). These notes will conclude with some indication of what goes wrong with a naïve approach.

Let us consider two differential equations

$\dot{y}_i = f_i(y_1;y_2)$, and suppose nothing more is known about $y_i(0)$ than that $y_i(0) \in Y_i(0) \equiv [a_i,b_i]$. It seems natural to approximate each $y_i(t)$ by an interval- valued function $Y_i(t) \equiv [x_i(t) , z_i(t)]$. Provided $y_i \in Y_i$ for all $t > 0$ we find that, for example,

$$\min_{\eta_2 \in Y_2} f_1(y_1;\eta_2) \le \dot{y}_1 \le \max_{\eta_2 \in Y_2} f_1(y_1;\eta_2) .$$

These inequalities resemble the classical differential inequality mentioned above, and lead to a natural generalization:

If $y_i(0) \in [x_i(0) , z_i(0)]$ and

$\dot{x}_1 \le R\, f_1(x_1;\ [x_2,z_2])$,

$\dot{z}_1 \ge R\, f_1(z_1;\ [x_2,z_2])$,

$\dot{x}_2 \le R\, f_2([x_1,z_1];x_2)$, and

$\dot{z}_2 \ge R\, f_2([x_1,z_1];x_2)$

for all $t > 0$, then $y_i(t) \in [x_i(t) , z_i(t)]$ too.

This theorem suggests that "the best" bounds for y_i will be obtained from the maximal solutions x_i and minimal solutions z_i of the differential inequalities; the only things wrong with the suggestion are the words "the best".

For example, suppose $f_i(y_1;y_2) \equiv (-1)^i y_{3-i}$. Then the desired, the maximal, and the minimal solutions satisfy

$\dot{y}_1 = -y_2$, $\dot{x}_1 = -z_2$, $\dot{z}_1 = -x_2$,

$\dot{y}_2 = y_1$, $\dot{x}_2 = x_1$, $\dot{z}_2 = z_1$.

The y-equations represent uniform rigid rotation of the $(y_1;y_2)$-plane. If at $t = 0$ the point $(y_1;y_2)$ lies in some

rectangle $(Y_1;Y_2)$, then for all $t > 0$ the point $(y_1;y_2)$ will lie in the image of that rectangle under rotation; the dimensions of the rectangle do not change, only its position and orientation. The theorem says that the rotating rectangle lies in another rectangle $([x_1,z_1]\ ;\ [x_2,z_2])$ for all t ; this larger rectangle's position changes too, but its sides stay parallel to the coordinate axes and lengthen like multiples of $\exp(t)$ as t increases (unless the rectangle started as a point). In fact, we verify immediately that

$$w_i(t) \equiv z_i(t) - x_i(t) = w_i(0)\ \cosh(t) + w_{3-i}(0)\ \sinh(t) \quad .$$

These results are substantially the same as obtained by Moore (1965a, and 1966, p.128) from his first method. The bounds produced by his second method (1965b, and 1966, ch.13) grow less quickly, but still exponentially too fast. Only Kahan (1967) claims to produce bounds which do not grow exponentially too fast, and then only when the bounds are small enough that uncertainties are propogated in a way adequately approximated by the linearized variational equations

$$\delta\dot{y}_i = \sum_j \frac{\partial}{\partial y_j} f_i \quad \delta y_j \quad .$$

Acknowledgement

This work has been supported partly by the National Research Council of Canada.

Departments of Mathematics
and of Computer Science,
University of Toronto,
June, 1968.

References

Adams, D.A. (1967) A Stopping Criterion for Polynomial Root Finding, *Comm. ACM* 10,655-8.

ALGOL 60 (1963) Revised Report on the Algorithmic Language ALGOL 60, Ed. by P. Naur, *Comm.ACM* 6, 1-17.

Apostolatos, N., et al. (1967) The Algorithmic Language Triplex - ALGOL 60, *Numer. Math.* 11, 175-180.

Apostolatos, N., and U. Kulisch (1967) Grundlagen einer Maschinenintevallarithmetik, *Z. Computing* 2, 89-104.

ASA standard FORTRAN (1964) *Comm. ACM*7, 590-625.

Birkhoff, G. (1967) *Lattice Theory* Amer. Math. Soc. Colloquium Publications vol.XXV, Providence, R.I.

Birkhoff, G. and G.-C. Rota (1959) *Ordinary Differential Equations*; Ginn, Boston.

Carathéodory, C. (1958) *Theory of Functions* vol.I, trans. by F. Steinhardt; Chelsea, New York.

Chartres, B.A. (1966) Automatic Controlled Precision Calculations, *J. ACM.*13, 386-403.

Collatz, L. (1966) *Functional Analysis and Numerical Mathematics*; Academic Press, New York, N.Y.

Control Data Corporation (1967) *6400/6500/6600 Computer Systems* Reference Manual 60100000 (Rev.D), St. Paul, Minn.

Coxeter, H.S.M. (1949) *The Real Projective Plane*; McGraw-Hill, N.Y.

Dieudonné, J. (1960) *Foundations of Modern Analysis*; Academic Press, New York.

Dwyer, P.S. (1951) *Linear Computations* ; Wiley, New York.

Eggleston, H.G. (1958) *Convexity*, Cambridge Tracts in Math. and Math. Phys. no.47; Cambridge Univ. Press.

Guderley, K.G. and Marian Valentine (1967) On Error Bounds for the Solution of Systems of Ordinary Differential Equations, pp.45-89 of the *Blanch Anniversary Volume*, Aerospace Research Labs., Wright-Patterson AFB, Ohio.

Hansen, E.R. (1968) On Solving Systems of Equations Using Interval Arithmetic, *Math. of Comp* 22, 374-384.

Henrici, P. (1962) *Discrete Variable Methods in Ordinary Differential Equations* ; Wiley, New York.

Kahan, W.M. (1966) Abstract in *SIAM Rev.* 8, 568-9.

Kahan, W.M. (1967) An Ellipsoidal Error Bound for Linear Systems of Ordinary Differential Equations, *Manuscript*, to appear.

Kahan, W.M. and I. Farkas (1963) Algorithms 168 and 169, *Comm.ACM* 6, 165.

Kahan, W.M. (1966) *7094-II System Support for Numerical Analysis, draft of first half* , distributed as item C-4537 in SHARE SSD no.159 (Dec.1966), reprinted in *Proc. 1967 Army Numer. Anal. Conference* (Wisconsin, May 1967) ARO-D Report 67-3, pp.175-208 & errata.

Moore, Ramon E. (1965a) The Automatic Analysis and Control of Error in Digital Computing Based on the use of Interval Numbers, in *Error in Digital Computation* , vol.I, ed. by L.B. Rall; Wiley, New York.

Moore, Ramon E. (1965b) Automatic Local Coordinate Transformations to Reduce the Growth of Error Bounds in Interval Computation of Solutions of Ordinary Differential Equations, in *Error in Digital Computation* , vol.II, ed. by L.B. Rall; Wiley, New York.

Moore, Ramon E. (1966) *Interval Analysis* ; Prentice-Hall, Englewood Cliffs, N.J.

Nickel, K. (1966) Über die Notwendigkeit einer Fehlerschranken-Arithmetik für Rechenautomaten, *Numer. Math* 9, 69-79.

Rudin, W. (1953) *Principles of Mathematical Analysis* ; McGraw-Hill, New York.

Smith, Brian Terence (1967a) ZERPOL, a Zero Finding Algorithm for Polynomials Using Laguerre's Method, *Proc. 1967 Army Numer. Anal. Conference* , ARO-D Report 67-3, pp.153-174.

Smith, Brian Terence (1967b) *A Zero Finding Algorithm Using Laguerre's Method* ; M.Sc. thesis, Univ. of Toronto.

Szarski, J. (1965) *Differential Inequalities* ; Polish Sci. Publ., Warsaw.

Toby, R.G., et al (1967) *PL/I-FORMAC INTERPRETER User's Manual* ; IBM, Cambridge, Mass. Also distributed by SHARE as SDA no.3538.

ERRATA

W. Kahan
6818

Page	Line	
26	8	After "or input/output" aid "unless we demand outer approximation during binary-decimal conversion".
24	5-1)	A similar expedient is proposed by Richard J. Hanson in "Interval Arithmetic as a Closed Arithmetic System on a Computer", JPL 314 Tech. Memo. 197, 4 June, 1968; however he does not allow for exterior intervals.
33	6	Change each Z to Z_0.