How Interval Arithmetic Works

Refer to [1] or [6] for a thorough introduction to interval mathematics. Further references appear in [5], and a list of about 2000 papers on the subject appears in [3] and [4]. The recent proceedings [7] contains assessments of the role of interval mathematics in scientific computation. Here, we will give an elementary explanation of some of the most important concepts.

We will denote interval quantities throughout with boldface.

Interval arithmetic is based on defining the four elementary arithmetic operations on intervals. Let $\mathbf{a} = [a_l, a_u]$ and $\mathbf{b} = [b_l, b_u]$ be intervals. Then, if $op \in \{+, -, *, /\}$, we define

$$\mathbf{a}\ op\ \ \mathbf{b} = \{x\ \ op\ \ y \mid x \in \mathbf{a} \text{ and } y \in \mathbf{b}\}. \tag{2.1}$$

For example, $\mathbf{a} + \mathbf{b} = [a_l + b_l, a_u + b_u]$. In fact, all four operations can be defined in terms of addition, subtraction, multiplication, and division of the endpoints of the intervals, although multiplication and division may require comparison of several results. The result of these operations is an interval except when we compute $\mathbf{a}/\mathbf{b}$ and $0 \in \mathbf{b}$. (See eg. [6], pp. 66-68 for a discussion of the latter case.)

A large part of interval mathematics' power lies in the ability to compute *inclusion monotonic interval extensions* of functions. If f is a continuous function of a real variable, then an inclusion monotonic interval extension $\mathbf{f}$ is defined to be a function from the set of intervals to the set of intervals, such that, if $\mathbf{x}$ is an interval in the domain of $\mathbf{f}$,

$$\{f(x) \mid x \in \mathbf{x}\} \subset \mathbf{f}(\mathbf{x})$$

and such that

$$\mathbf{x} \subset \mathbf{y} \implies \mathbf{f}(\mathbf{x}) \subset \mathbf{f}(\mathbf{y}).$$

Inclusion monotonic interval extensions of a polynomial may be obtained by simply replacing the dependent variable by an interval and by replacing the additions and multiplications by the corresponding interval operations. For example, if $p(x) = x^2 - 4$, then $\mathbf{p}([1, 2])$ may be defined by

$$\mathbf{p}([1, 2]) = ([1, 2])^2 - 4 = ([1, 4]) - [4, 4] = [-3, 0].$$

The result of an elementary interval operation is precisely the range of values that the usual result attains as we let the operands range over the two intervals. However, the value of an interval extension of a function is not precisely the range of the function over its interval operand, but only contains this range, and different (mathematically equivalent) forms of the function give rise to different interval extensions. For example, if we write p above as $p(x) = (x - 2)(x + 2)$, then the corresponding interval extension gives

$$\begin{aligned}\mathbf{p}([1, 2]) &= ([1, 2] - 2)([1, 2] + 2) = [-1, 0][3, 4] \\ &= [-4, 0],\end{aligned}$$

which is not as good as the previous extension.

We may use the mean value theorem or Taylor's theorem with remainder formula to obtain interval extensions of transcendental functions. For example, suppose $\mathbf{x}$ is an interval and $a \in \mathbf{x}$. Then, for any $y \in \mathbf{x}$, we have

$$\sin(y) = \sin(a) + (y-a)\cos(a) - (y-a)^2/2\sin(c)$$

for some c between a and y. If a and y are both within a range where the sine function is non-negative, then we obtain

$$\sin(y) \in \sin(a) + (\mathbf{x}-a)\cos(a) - \frac{(\mathbf{x}-a)^2}{2}.$$

The right side of this relationship gives the value of an interval extension of $\sin(x)$, albeit a somewhat crude one.

See [9] for more techniques of producing interval extensions.

Mathematically rigorous interval extensions can be computed in finite precision arithmetic via the use of *directed roundings*. If x and y are machine-representable numbers and op is one of the four elementary operations $+$, $-$, $*$, or $/$, then, $x\ op\ y$ is not normally representable in the machine's memory. In interval arithmetic with directed rounding, if $\mathbf{x}\ op\ \mathbf{y} = [c, d]$, then we always round the computed value for c down to a machine number less than the actual value of c, and and we always similarly round the value for d up. To be completely rigorous, we also first apply directed rounding to the initial data while storing it.

If interval arithmetic with directed rounding is used to compute an interval extension $\mathbf{f}$ of f, if $[c, d] = \mathbf{f}([a, b])$, and $[c, d]$ does not contain zero, then this is a rigorous proof (regardless of the machine wordlength, etc.) that there is no root of f in $[a, b]$. This concept is also valid if $\mathbf{F}$ is an interval vector-valued function of an interval vector $\mathbf{X}$, and complex interval arithmetic can also be defined. These facts should enable us to computationally but rigorously check the stability of systems. (See Section 4 below.)

If available, the language Fortran-SC is a convenient way of programming interval arithmetic computations. This precompiler is available on IBM mainframe equipment, and requires the ACRITH subroutine package; see [2] or [10]. There is also Pascal-SC for personal computers; see [8]. For a discussion of other programming languages and packages for interval arithmetic, see [13].

References

[1] G. ALEFELD and J. HERZBERGER, *Introduction to Interval Computations*, Academic Press, New York, etc., 1983.

[2] J. H. BLEHER, S. M. RUMP, U. KULISCH, M. METZGER, and W. WALTER, "Fortran-SC – A Study of a Fortran Extension for Engineering Scientific Computations with Access to ACRITH", *Computing*, v. 39, 1987, pp. 93–110.

[3] J. GARLOFF, "Interval Mathematics: A Bibliography", preprint, Institut für Angewandte Mathematik der Universität Freiburg, *Freiburger Intervall-Berichte*, v. 85, 1985, pp. 1–222.

[4] J. GARLOFF, "Bibliography on Interval Mathematics, Continuation" preprint, Institut für Angewandte Mathematik der Universität Freiburg, *Freiburger Intervall-Berichte*, v. 87, 1987, pp. 1–50.

[5] R. B. KEARFOTT, "Interval Arithmetic Techniques in the Computational Solution of Nonlinear Systems of Equations: Introduction, Examples, and Comparisons", to appear in the proceedings of the 1988 AMS-SIAM Summer Seminar in Applied Mathematics, Colorado State University, July 18-29, 1988.

[6] R. E. MOORE, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[7] R. E. MOORE, ed., *Reliability in Computing*, Academic Press, New York, etc., 1988.

[8] L. B. RALL, "An Introduction to the Scientific Computing Language Pascal-SC", *Comput. Math. Appl.*, v. 14, 1987, pp. 53–69.

[9] H. RATSCHEK and J. G. ROKNE, *Computer Methods for the Range of Functions*, Horwood, Chichester, England, 1984.

[10] W. WALTER and M. METZGER, "Fortran-SC, A Fortran Extension for Engineering/Scientific Computation with Access to ACRITH", in *Reliability in Computing*, R. E. Moore, ed., Academic Press, New York, etc., 1988.