A. Sydow (Ed.): 15^{th} IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics. Volume 2, Numerical Mathematics, W&T Verlag, Berlin, 1997.

Tawards Credible Implementation of Inner Interval Operations

Evgenija D. Popova & Svetoslav. M. Markov
Institute of Mathematics and Computer Science
Bulgarian Academy of Sciences
"Acad. G. Bonchev" str., block 8, BG-1113 Sofia, Bulgaria
e-mails: epopova@bio.bas.bg; smarkov@bio.bas.bg

Keywords: Interval Arithmetic, Implementation, Reliability

INTRODUCTION

Interval arithmetic is a widely used technique providing validated numerical results. have been proposed aiming at improving its properties and finding tight bounds to solutions of some problems in an effective way. We consider conventional interval arithmetic [1], extended by four inner interval operations [7]. The obtained extended interval arithmetic structure is suitable for the effective computation of functional ranges reducing their overestimation with ordinary interval arithmetic [2], [4].

Additionally, interval arithmetic, with directed roundings [6], can provide mathematically rigorous results from floating-point operations on computers. Although the arithmetic operations with directed roundings, specified by IEEE floating-point standards [3], are sufficient for the implementation of conventional interval operations with 1 ulp (unit in the last place) accuracy of the interval end-points, some reliability problems may occur implementing supplementary interval operations. This paper briefly outlines interval arithmetic, extended by four supplementary interval operations, discuss a source of numerical errors at the implementation of floating-point inner interval operations and shows different ways for their suppression. The goal is to make computations involving these operations more accurate and credible.

INTERVAL SPACE $(IR, +, +^-, \times, \times^-, \subseteq)$

Consider the set of intervals $IR = \{[a^-, a^+] \mid a^- \leq a^+;\ a^-, a^+ \in R\}$. Denote by $Z = 0 \cup \{[a^-, a^+] \mid a^- < 0 < a^+;\ a^-, a^+ \in R\}$ the set of all intervals involving zero in their interior. We utilize functional "$\pm$" notations a^λ for the interval end-points, where $\lambda \in \Lambda = \{+, -\}$, so that interval formulas can be written in a more closed form. The binary variable λ can be expressed as a "product" of two or more binary variables, $\lambda = \mu\nu$, $\mu, \nu \in \Lambda$, defined by $++ = -- = +$, and $+- = -+ = -$. Several functionals, defined to characterize intervals, are widely used in interval analysis. For an interval $A \in IR \setminus Z$ "sign" $\sigma : IR \setminus Z \rightarrow \Lambda$ is defined by $\sigma(A) = \{+,\ \text{if}\ a^- \geq 0;\ -,\ \text{if}\ a^+ < 0\}$. For every interval $A \in IR$ "width" $\omega : IR \rightarrow R^+$ is defined by $\omega = a^+ - a^-$. Functional $\chi : IR \rightarrow [-1, 1]$ is defined to characterize a symmetry behaviour of intervals by $\chi(A) = -1$ for $A = [0, 0]$ and $\chi(A) = \{a^-/a^+\ \text{if}\ |a^-| \leq |a^+|;\ a^+/a^-,\ \text{otherwise.}\}$. Therefore functional χ admits the geometric interpretation that A is more symmetric than B iff $\chi(A) \leq \chi(B)$.

The set IR together with the ordinary relation for inclusion $\subseteq$ and the basic interval arithmetic operations form the well-known interval arithmetic space $(IR, +, -, \times, /, \subseteq)$ [1]. Addition and multiplication operations from this space can be expressed by

$$\begin{aligned} A + B &= [a^- + b^-,\ a^+ + b^+], \quad \text{for } A, B \in IR; \\ A \times B &= \begin{cases} [a^{-\sigma(B)} b^{-\sigma(A)},\ a^{\sigma(B)} b^{\sigma(A)}], & \text{for } A, B \in IR \setminus Z, \\ [a^{\sigma(A)} b^{-\sigma(A)},\ a^{\sigma(A)} b^{\sigma(A)}], & \text{for } A \in IR \setminus Z,\ B \in Z, \\ [a^{-\sigma(B)} b^{\sigma(B)},\ a^{\sigma(B)} b^{\sigma(B)}], & \text{for } A \in Z,\ B \in IR \setminus Z, \\ [\min\{a^- b^+, a^+ b^-\},\ \max\{a^- b^-, a^+ b^+\}], & \text{for } A, B \in Z. \end{cases} \end{aligned}$$

In [7] the set of conventional interval arithmetic operations was extended by two supplementary operations $+^-, \times^-$:

$$A +^- B = [a^{-\alpha} + b^{\alpha},\ a^{\alpha} + b^{-\alpha}], \quad \text{for } A, B \in IR, \tag{1}$$

$$A \times^- B = \begin{cases} [a^{\sigma(B)\varepsilon} b^{-\sigma(A)\varepsilon},\ a^{-\sigma(B)\varepsilon} b^{\sigma(A)\varepsilon}], & \text{for } A, B \in IR \setminus Z, \\ [a^{-\sigma(A)} b^{-\sigma(A)},\ a^{-\sigma(A)} b^{\sigma(A)}], & \text{for } A \in IR \setminus Z,\ B \in Z, \\ [a^{-\sigma(B)} b^{-\sigma(B)},\ a^{\sigma(B)} b^{-\sigma(B)}], & \text{for } A \in Z,\ B \in IR \setminus Z, \\ [\ \max\{a^- b^+, a^+ b^-\},\ \min\{a^- b^-, a^+ b^+\}], & \text{for } A, B \in Z, \end{cases} \tag{2}$$

where $\alpha = \{+, \text{ if } \omega(A) \geq \omega(B);\ -, \text{ if } \omega(A) \leq \omega(B)\}$ and $\varepsilon = \{+, \text{ if } \chi(A) \geq \chi(B);\ -, \text{ if } \chi(A) \geq \chi(B)\}$.

The elements $-A = [-a^+, -a^-]$ and $1/A = [1/a^+, 1/a^-]$ are inverse with respect to the operations $+^-$ and $\times^-$, resp. Thus the following operations can be obtained as composite:

$$\begin{aligned} A - B &= A + (-B) = [a^- - b^+,\ a^+ - b^-], \quad \text{for } A, B \in IR, \\ A -^- B &= A +^- (-B) = [a^{-\alpha} - b^{-\alpha},\ a^{\alpha} - b^{\alpha}], \quad \text{for } A, B \in IR, \\ A/B &= A \times (1/B) = \begin{cases} [a^{-\sigma(B)}/b^{\sigma(A)},\ a^{\sigma(B)}/b^{-\sigma(A)}], & \text{for } A, B \in IR \setminus Z, \\ [a^{-\sigma(B)}/b^{-\sigma(B)},\ a^{\sigma(B)}/b^{-\sigma(B)}], & \text{for } A \in Z,\ B \in IR \setminus Z, \end{cases} \end{aligned} \tag{3}$$

$$A/^-B = A \times^- (1/B) = \begin{cases} [a^{\sigma(B)\varepsilon}/b^{\sigma(A)\varepsilon},\ a^{-\sigma(B)\varepsilon}/b^{-\sigma(A)\varepsilon}], & \text{for } A, B \in IR \setminus Z, \\ [a^{-\sigma(B)}/b^{\sigma(B)},\ a^{\sigma(B)}/b^{\sigma(B)}], & \text{for } A \in Z,\ B \in IR \setminus Z; \end{cases} \tag{4}$$

α and ε are defined as above.

Every operation $\circ \in \{+, -, \times, /\}$ satisfies the relation:

$$A \circ B = \{a \circ b \mid a \in A, b \in B\}, \quad A, B \in IR\, (B \in IR \setminus Z, \text{ if } \circ = /)$$

and the corresponding supplementary operation satisfies the relation $A \circ^- B \subseteq A \circ B$, due to which $\circ^-$ operations are called *inner* interval operations and the conventional ones — *outer* interval operations.

The extended interval arithmetic structure $(IR, +, +^-, \times, \times^-, \subseteq)$ posesses better algebraic properties than the conventional one:

- For $A, B \in IR$ and $\circ \in \{+, +^-, \times, \times^-\}$, $A \circ B = B \circ A$.
- The outer $+$ and $\times$ operations are associative, while $+^-$ and $\times^-$ are conditionally associative depending on the value of α, resp. ε.
- $X = [0, 0] = 0$ and $Y = [1, 1] = 1$ are the unique neutral elements with respect to all (inner and outer) addition and multiplication operations.

• The elements $-A$ and $1/A$ satisfy $0 \subseteq A+(-A) = A-A$, resp. $1 \subseteq A \times (1/A) = A/A$ and they are the unique inverse elements with respect to $+^-$, $\times^-$ so that $A+^-(-A) = A-^-A = 0$, resp. $A \times^- (1/A) = A/^-A = 1$.

• There exist conditionally distributive relations connecting $+, +^-, \times$ and $\times^-$.

A number of numerical algorithms with result verification have been developed showing the advantages of inner interval operations (see e. g. [2], [4]). Some properties of the extended interval arithmetic structure $(IR, +, +^-, \times, \times^-, \subseteq)$, its relations to other interval arithmetic extensions and a large list of references can be found in [8].

WIDTH AND SYMMETRY BASED IMPLEMENTATION

The theory of computer arithmetic defines computer interval arithmetic by semimorphism [6]. Let F be a floating-point symmetric screen over R and $F = F(b, p, emin, emax)$ is defined by its base b, its precision p, and its minimal and maximal exponent, $emin$ and $emax$. If denote by $IF = \{[a^-, a^+] \in IR \mid a^-, a^+ \in F\}$ the set of computer representable intervals, then $\{IF, \subseteq\}$ is a screen of $\{IR, \subseteq\}$. Rounding $\Box : IR \longrightarrow IF$ is defined as a monotonic function with the properties projection and monotonicity. A third property (inclusion) specifies directed roundings for intervals $A \in IR$:

$$\text{outward rounding} : \Diamond A = [\triangledown a^-, \triangle a^+] \supseteq A; \qquad \text{inward rounding} : \bigcirc A = [\triangle a^-, \triangledown a^+] \subseteq A,$$

where $\triangledown, \triangle : R \longrightarrow F$ are corresponding floating-point directed roundings toward $-\infty$ ($\triangledown$) and toward $+\infty$ ($\triangle$) [3], [6]. If $\circ \in \{+, +^-, -, -^-, \times, \times^-, /, /^-\}$ is an arithmetic operation in IR, the corresponding computer operation $\boxdot$ in IF is defined by

$$A \boxdot B = \Box (A \circ B), \qquad \text{for} \quad A, B \in IF, \ \ \Box \in \{\Diamond, \bigcirc\}.$$

The explicit formulae for computation of the corresponding result are summarized as follows:

$$\Diamond (A \circ B) = [\ \triangledown (A \circ B)^-, \ \ \triangle (A \circ B)^+\ ]; \qquad \bigcirc (A \circ B) = [\ \triangle (A \circ B)^-, \ \ \triangledown (A \circ B)^+\ ].$$

In IEEE floating-point environment, outer interval operations can be realized easily with 1 ulp accuracy by floating-point operations with directed rounding. End-points of the resulting intervals of outer $+$ and $-$ operations are specified explicitly by floating-point expressions. End-points of the resulting intervals of outer $\times$ and $/$ operations are determined by the algebraic signs of the end-points of interval arguments. The algebraic sign of a rounded floating-point number is determined exactly by the IEEE systems.

The implementation of inner interval operations, however, is more difficult. The defining formulae of inner addition and subtraction operations (1), (2) involve comparison of the width of the arguments, and the defining formulae of multiplication and division operations (3), (4) involve comparison of the symmetry functional for the arguments. Computation of each of these functionals require a floating-point operation involving round-off error. Even with $+^-$ and $-^-$ operations, which seem so simple, the round-off error at the computation of ω for each of the interval arguments may lead to a incorrect comparison and as a consequence, to a wrong interval result for certain values of the arguments. Next examples illustrate this effect.

Example 1. *Compute* $\diamond(A +^- B)$ *for* $A = [0.465 * 10^2, 0.746 * 10^3]$ *and* $B = [-0.85 * 10^1, 0.692 * 10^3]$ *in decimal floating-point system* $F = F(10, 3, -6, 6)$. The exact width values $\omega(A) = 699.5 < \omega(B) = 700.5$ are not elements of F, and $\Box\omega(A) = \Box\omega(B) = 0.7 * 10^3$, where $\Box$ denotes *round-to-nearest*. An implementation, based on the defining formula (1) with $\alpha = +$, will produce a wrong result $[\bigtriangledown(a^- + b^+), \bigtriangleup(a^+ + b^-)] = [0.738 * 10^3, 0.738 * 10^3]$, for which $A +^- B \not\subseteq \diamond(A +^- B)$, while the correct result is $[0.737 * 10^3, 0.739 * 10^3]$.

Using one of the directed roundings $\Box \in \{\bigtriangledown, \bigtriangleup\}$ for the computation of $\Box\omega(A)$ and $\Box\omega(B)$ will help to perform a correct comparison of latter and thus to obtain a correct result of $\diamond(A +^- B)$ when A and B are those of Example 1 but a wrong result may be delivered for other values of A and B.

Example 2. *Compute* $\diamond(A +^- B)$ *for* $A = [-0.5 * 10^3, 0.999 * 10^6]$ *and* $B = [-0.85 * 10^4, 0.999 * 10^6]$ *in decimal floating-point system* $F = F(10, 3, -6, 6)$*, comparing* $\bigtriangleup\omega(A)$ *and* $\bigtriangleup\omega(B)$. The exact width values $\omega(A) = 999.5 * 10^3 < \omega(B) = 1000.5 * 10^3$ are not elements of F, and $\bigtriangleup\omega(A) = \bigtriangleup\omega(B) = \infty$. An implementation, based on the defining formula (1) with $\alpha = +$, will produce a wrong result $[\bigtriangledown(a^- + b^+), \bigtriangleup(a^+ + b^-)] = [0.998 * 10^6, 0.998 * 10^6]$, for which $A +^- B \not\subseteq \diamond(A +^- B)$, while the correct result is $[0.997 * 10^6, 0.999 * 10^6]$.

A comparison operation between $\chi(A)$ and $\chi(B)$, for the computation of $\Box(A \circ^- B)$ where $\Box \in \{\bigcirc, \diamond\}$ and $\circ \in \{\times, /\}$, may be influenced by even more round-of errors. The above examples show that an implementation of inner interval operations, based on comparison between rounded values of ω, resp. χ functionals, cannot produce credible results. For a program to be credible, the results it produces must never be misleading.

CREDIBLE IMPLEMENTATION ALGORITHMS

There exist two alternatives for implementation of outer interval multiplication operation. First alternative involves nine cases determined by the algebraic signs of the end-points of the operands. Second alternative involves computation of minimum and maximum of the four possible products of the operands end-points according to the following formula

$$\begin{aligned}\diamond(A \times B) \;=\; & [\min\{\bigtriangledown(a^-b^-), \bigtriangledown(a^-b^+), \bigtriangledown(a^+b^-), \bigtriangledown(a^+b^+)\},\\ & \max\{\bigtriangleup(a^-b^-), \bigtriangleup(a^-b^+), \bigtriangleup(a^+b^-), \bigtriangleup(a^+b^+)\}].\end{aligned}$$

The average number of multiplications required for the first alternative is less than that required by the second one. Implemented in software, the relative efficiencies of both alternatives are architecture dependent, although the first alternative is often preferred in low-level implementations designed for efficiency. Considerations in the previous section showed that for inner interval operations we have to seek implementation algorithms based on rounded expressions only for the result end-points, rather than using *a priori* floating-point computations of ω or χ.

A careful analysis of formulae (1)–(4) and some investigations (see e. g. [8],[9]) of the relations between extended interval arithmetic structure $(IR, +, +^-, \times, \times^-, \subseteq)$ and other algebraic extensions [5] of conventional interval arithmetic, led to a second alternative for the implementation of inner interval operations, based on the following formulae:

$$\diamond(A +^- B) \;=\; [\min\{\bigtriangledown(a^- + b^+), \bigtriangledown(a^+ + b^-)\},\ \max\{\bigtriangleup(a^- + b^+), \bigtriangleup(a^+ + b^-)\}]; \qquad (5)$$

$$\diamond(A -^- B) = [\min\{\bigtriangledown(a^- - b^-), \bigtriangledown(a^+ - b^+)\},\ \max\{\triangle(a^- - b^-), \triangle(a^+ - b^+)\}]; \tag{6}$$

$$\diamond(A \times^- B) = \begin{cases} [\min\{\bigtriangledown(a^-b^+), \bigtriangledown(a^+b^-)\}, \max\{\triangle(a^-b^+), \triangle(a^+b^-)\}], \\ \qquad\qquad \text{if } \sigma(A) = \sigma(B), A, B \in IR \setminus Z; \\ [\min\{\bigtriangledown(a^-b^-), \bigtriangledown(a^+b^+)\}, \max\{\triangle(a^-b^-), \triangle(a^+b^+)\}], \\ \qquad\qquad \text{if } \sigma(A) \neq \sigma(B), A, B \in IR \setminus Z; \\ [\min\{\bigtriangledown(a^{-\sigma(A)}b^{-\sigma(A)}), \bigtriangledown(a^{-\sigma(A)}b^{\sigma(A)})\}, \\ \quad \max\{\triangle(a^{-\sigma(A)}b^{-\sigma(A)}), \triangle(a^{-\sigma(A)}b^{\sigma(A)})\}], \quad \text{if } A \in IR \setminus Z, B \in Z; \\ [\min\{\bigtriangledown(a^{-\sigma(B)}b^{-\sigma(B)}), \bigtriangledown(a^{\sigma(B)}b^{-\sigma(B)})\}, \\ \quad \max\{\triangle(a^{-\sigma(B)}b^{-\sigma(B)}), \triangle(a^{\sigma(B)}b^{-\sigma(B)})\}], \quad \text{if } A \in Z, B \in IR \setminus Z; \\ [\max\{\bigtriangledown(a^-b^+), \bigtriangledown(a^+b^-)\}, \min\{\triangle(a^-b^-), \triangle(a^+b^+)\}], \quad \text{if } A, B \in Z; \end{cases} \tag{7}$$

$$\diamond(A /^- B) = \begin{cases} [\min\{\bigtriangledown(a^-/b^-), \bigtriangledown(a^+/b^+)\}, \max\{\triangle(a^-/b^-), \triangle(a^+/b^+)\}], \\ \qquad\qquad \text{if } \sigma(A) = \sigma(B), A, B \in IR \setminus Z; \\ [\min\{\bigtriangledown(a^-/b^+), \bigtriangledown(a^+/b^-)\}, \max\{\triangle(a^-/b^+), \triangle(a^+/b^-)\}], \\ \qquad\qquad \text{if } \sigma(A) \neq \sigma(B), A, B \in IR \setminus Z; \\ [\min\{\bigtriangledown(a^{-\sigma(B)}/b^{\sigma(B)}), \bigtriangledown(a^{\sigma(B)}/b^{\sigma(B)})\}, \\ \quad \max\{\triangle(a^{-\sigma(B)}/b^{\sigma(B)}), \triangle(a^{\sigma(B)}/b^{\sigma(B)})\}], \quad \text{if } A \in Z, B \in IR \setminus Z. \end{cases} \tag{8}$$

The implementations, based on formulae (5)–(8), provide credible results for inner interval operations. Inner multiplication operation involves seven cases, determined by the algebraic signs of the end-points of the operands. For inner addition and subtraction operations and each sign-dependent case of inner multiplication and division operations, these algorithms involve four arithmetic operations and two comparison operations. Note, however, that in all cases these algorithms compute only two floating-point expressions using directed rounding, rather than four expressions by the corresponding algorithm for outer interval operations. This observation allows us to formulate more simple algorithms for the implementation of inner addition and subtraction operations and each sign-dependent case of inner multiplication and division operations. Denote by $\mathtt{expr}_1$ one of the two floating-point expressions, for the corresponding case, defined by formulae (5)–(8), and by $\mathtt{expr}_2$ the other one.

Algorithm 1. *Outwardly rounded inner interval operations.*

Compute $r^- = \bigtriangledown(\mathtt{expr}_1)$; $r^+ = \bigtriangledown(\mathtt{expr}_2)$;

If $r^- \le r^+$ then $\mathtt{res} = [r^-, \triangle(\mathtt{expr}_2)]$

else $\mathtt{res} = [r^+, \triangle(\mathtt{expr}_1)]$.

This algorithm involves three arithmetic and one comparison operations. Inwardly rounded inner interval operations play important role for computation of functional ranges so that their implementation is also of interest. When both exact end-point values for the result of an inner interval operation fall between two successive machine numbers, no element of IF can represent the inwardly rounded result of the corresponding inner interval operation. That is why a more complicated algorithm should be implemented to provide credible results from inwardly rounded inner interval operations. Algorithm 2 requires three arithmetic and two comparison operations in the worst case.

Algorithm 2. *Inwardly rounded inner interval operations.*

Compute $r^- = \bigtriangledown(\texttt{expr}_1)$; $r^+ = \bigtriangledown(\texttt{expr}_2)$;

If $r^- < r^+$ then $\texttt{res} = [\bigtriangleup(\texttt{expr}_1),\ r^+]$

elseIf $r^- > r^+$ then $\texttt{res} = [\bigtriangleup(\texttt{expr}_2),\ r^-]$

else `Signal INVALID OPERATION`
deliver `[qNaN, qNaN]`.

CONCLUSION

We explored the sources of numerical errors in the implementation of floating-point inner interval operations and showed different ways for their suppression. The proposed implementation algorithms are efficient and can be used for developing credible and accurate programs. We believe that present investigations will help the optimized implementation of interval arithmetic by language compilers and in hardware so that it may be fast and efficient.

Acknowledgements. This work was partially supported by the Bulgarian National Science Fund under grand No. I-507/95.

REFERENCES

[1] Alefeld, G.; Herzberger, J.: *Introduction to Interval Computations.* Academic Press (1983).

[2] Alt, R.; Lamotte, J.-L.: *On the Evaluation of Functional Ranges Using a Random Interval Arithmetic.* Extended Abstracts INTERVAL'96, Int. Conf. on Interval Methods and Computer Aided Proofs in Science and Engineering, Würzburg, pp. 11–12 (1996).

[3] ANSI/IEEE: *IEEE Standard for Binary Floating-Point Arithmetic.* ANSI/IEEE Std 754–1985, New York (1985).

[4] Bartholomew-Biggs, M. C.; Zakovic, S.: *Using Markov's Interval Arithmetic to Evaluate Bessel-Ricatti Functions.* Numerical Algorithms, **10**, pp. 261–287 (1995).

[5] Kaucher, E.: *Interval Analysis in the Extended Interval Space IR.* Computing Suppl. **2**, pp. 33–49 (1980).

[6] Kulisch, U., Miranker, W. L.: *Computer Arithmetic in Theory and Practice.* Academic Press, New York (1981).

[7] Markov, S. M.: *Extended Interval Arithmetic.* Compt. Rend. Acad. Bulg. Sci., **30**, 9, pp. 1239–1242 (1977).

[8] Markov, S. M: *On Directed Interval Arithmetic and its Applications.* Journal of Universal Computer Science, **1**, 7, pp. 510-521 (1995).

[9] Popova, E. D.: *Transition Formulae Between Some Interval Arithmetic Structures.* Manuscript, Inst. of Mathematics & Informatics, BAS, Sofia (1996).