Interval Arithmetic and Standardization

Jürgen Wolff von Gudenberg
University of Würzburg

Extended Abstract
Dagstuhl Seminar 08021, 2008

[1]

1 Introduction

Interval arithmetic is arithmetic for continuous sets. Floating-point intervals are intervals of real numbers with floating-point bounds. Operations for intervals can be efficiently implemented. Hence, the time is ripe for standardization. In this paper we present an interval model that is mathematically sound and closed for the 4 basic operations. The model allows for exception free interval arithmetic, if we carefully distinguish between clean and reliable interval arithmetic on one side and rounded floating-point arithmetic on the other side. Elementary functions for intervals can be defined. In some application areas loose evaluation of functions, i.e. evaluation over an interval which is not completely contained in the function domain, is recommended, In this case, however, a discontinuity flag has to be set to inform that Brouwer's fixed point theorem is no longer applicable.

2 Real Interval Arithmetic

2.1 Real Interval Arithmetic

Real interval arithmetic is defined as arithmetic on continuous (in the sense of complete, not discrete) sets.

Definition 1 *For intervals $A = [a_1, a_2]$ and $B = [b_1, b_2] \in \mathbb{IR}$, arithmetic operations are defined as set operations*

$$A \circ B := \{a \circ b \mid a \in A, b \in B\}$$

for operations $\circ \in \{+, -, \cdot, /\}$, $0 \notin B$ in case of division.

Remark 1 *Since the operations are continuous, the result is an interval, $A \circ B \in \mathbb{IR}$*

[1]Thanks to Ulrich Kulisch, Gerd Bohlender, Rudi Klatte, John Pryce and all other paritcipants who helped in the discussion.

Dagstuhl Seminar Proceedings 08021
Numerical Validation in Current Hardware Architectures
http://drops.dagstuhl.de/opus/volltexte/2008/1434

This definition can be extended to elementary functions.

Definition 2 $f(X) = \{f(x) \mid x \in X\}$ *denotes the range of values of the function* $f : D_f \subseteq \mathbb{R} \to \mathbb{R}$ *over the interval* $X \subseteq D_f$.

Remark 2 *If f is continuous, the range is an interval.*

Remark 3 *The range is independent from the specific expression that describes the function. That is not the case for the interval evaluation.*

Definition 3 *The* ***interval evaluation*** $\mathbf{f} : \mathbb{IR} \to \mathbb{IR}$ *(of a real function* f *over an interval* X*) is defined as the function that is obtained by replacing every operator and every elementary function by its interval arithmetic counterpart under the assumption that all operations are executable without exceptions.*

Remark 4 *The variables in an interval evaluation now denote intervals.*
An obvious generalisation for functions of multiple variables may be defined.

Two basic principles are mandatory for every definition of interval arithmetic.
The containment principle is known as the fundamental theorem of interval arithmetic [1].

Principle 1 *If the interval evaluation is defined, we have*

$$f(X) \subseteq \mathbf{f}(X)$$

The second priciple, the inclusion isotonicity is related.

Principle 2 *If* $X \subseteq Y$*, we have*

$$\mathbf{f}(X) \subseteq \mathbf{f}(Y)$$

3 Floating-point Interval Arithmetic

When we proceed to the set of floating-point intervals, our topic for standardization, we are still talking about continuous sets, only the endpoints are floating-point numbers. For the sake of clarity and to emphasize the difference, we introduce a separate notation for floating-point interval arithmetic.

Definition 4 *Let R denote the set of floating-point numbers, then* $\mathbb{I}R \subset \mathbb{IR}$ *denotes the set of floating-point intervals* $A = [a_1, a_2]$ *where* $a_1, a_2 \in R$ *and* $a_1 \leq a_2$

Remark 5 $A = [a_1, a_2] = \{a \in \mathbb{R} \mid a_1 \leq a \leq a_2\}$

Definition 5 *The* ***floating-point interval evaluation*** $\diamond\mathbf{f} : \mathbb{I}R \to \mathbb{I}R$ *of the function expression* f *is defined as the function that is obtained by replacing every operator and every elementary function by its floating-point interval arithmetic counterpart under the assumption that all operations are executable without exceptions.*

The containment principle guarantees that every real number in the original range of values of a continuous function is contained in the result of the floating-point interval evaluation of the same function over the same argument interval.

Principle 3 *If the floating-point interval evaluation for f is defined, we have*

$$f(x) \in \diamond\mathbf{f}(X), \; \forall x \in X \cap D_f$$

Remark 6 *The floating-point interval evaluation is defined, if f is continuous and $X \subseteq D_f$.*

A clean semantics that respects the two basic principles of containment and inclusion isotonicity is mandatory. It can be obtained when we implement the well-known formulae involving only the endpoints and use directed roundings. In the following we indicate rounding towards $-\infty$ by a $\check{\circ}$ and rounding towards $+\infty$ by a $\hat{\circ}$ over the operator symbol $\circ$.

3.1 Representation

A finite floating-point interval is represented by two floating-point numbers, the first a_1 denotes the lower bound, the second a_2 the upper bound. For a valid interval we have $a_1 \leq a_2$.
An infinite interval has its lower bound set to $-\infty$ or its upper bound set to $+\infty$.
The empty set is denoted by $[+\infty, -\infty]$
All other representations, in particular two valid numbers with $a_1 > a_2$, denote invalid intervals.

3.2 Arithmetic Operations

The operations addition, subtraction, multiplication, and division by an interval which does not contain zero are defined as usual. The division by an interval containing zero raises an exception.

4 Infinite Intervals and Division by Zero

4.1 The Set Approach

In section 3.1 we introduced intervals with one endpoint $+\infty$ or $-\infty$. ∞ is not a valid point in the interval, it just states that the interval is unbounded [6]. We do not allow intervals with lower bound $+\infty$ or upper bound $-\infty$.
We now define division by an interval containing zero. Rewriting the definition, we obtain:

$$A/B := \{a/b \mid a \in A, b \in B\} = \{x \mid bx = a \wedge a \in A \wedge b \in B\}$$

Applying this formula eight distinct cases can be set out. In the following table in column 3 we display the 2 bounds, that are returned by the operation. Since no valid

case	$A = [a_1, a_2]$	$B = [b_1, b_2]$	$result$	A/B
1	$0 \in A$	$0 \in B$	$-\infty, +\infty$	$(-\infty, +\infty)$
2	$0 \notin A$	$B = [0, 0]$	$+\infty, -\infty$	$\emptyset$
3	$a_2 < 0$	$b_1 < b_2 = 0$	$a_2 \check{/} b_1, +\infty$	$[a_2 \check{/} b_1, +\infty)$
4	$a_2 < 0$	$b_1 < 0 < b_2$	$a_2 \check{/} b_1, a_2 \hat{/} b_2$	$(-\infty, a_2 \hat{/} b_2] \cup [a_2 \check{/} b_1, +\infty)$
5	$a_2 < 0$	$0 = b_1 < b_2$	$-\infty, a_2 \hat{/} b_2$	$(-\infty, a_2 \hat{/} b_2]$
6	$a_1 > 0$	$b_1 < b_2 = 0$	$-\infty, a_1 \hat{/} b_1$	$(-\infty, a_1 \hat{/} b_1]$
7	$a_1 > 0$	$b_1 < 0 < b_2$	$a_1 \check{/} b_2, a_1 \hat{/} b_1$	$(-\infty, a_1 \hat{/} b_1] \cup [a_1 \check{/} b_2, +\infty)$
8	$a_1 > 0$	$0 = b_1 < b_2$	$a_1 \check{/} b_2, +\infty$	$[a_1 \check{/} b_2, +\infty)$

Table 1: The eight cases of interval division with $A, B \in IS$, and $0 \in B$.

intervals are returned, if 0 is in the interior of B, we add a 4-th column with a set interpretation.
Since, in case 1, $0 \in A$ and $0 \cdot x = 0, \forall x \in \mathbb{R}$ we have $\mathbb{R} = [-\infty, \infty]$ as the solution set, whereas in case 2 there is no $x \in \mathbb{R}$ with $0 \cdot x = a$ for $a \in A$. The other cases are derived by limit processes, or by the arithmetic conventions for infinities.

Remark 7

- *If 0 is in the interior of B, the solution set consists of 2 infinite intervals.*
- *Alternatively the whole line $\mathbb{R}$ can be returned, but that would loose valuable information.*

4.2 Discussion

Let us further discuss the 2 alternatives. The former is consistent with the definition of interval arithmetic as set arithmetic. The subintervals are used in the interval Newton method as 2 sets possibly containing zeros. The middle part $(a_1 \check{/} b_1, a_1 \hat{/} b_2)$ is cut out, since it cannot contain a zero. Hence, the process proceeds, whereas the whole $\mathbb{R}$, swollows this information and the process stops.
The latter solution has two obvious advantages. The system is closed, i.e. in any case a valid interval is returned, and the containment principle also holds for floating-point results, that are no real numbers, but the symbols $\pm\infty$. In this case we have to replace the empty set in row 2 by the whole set, again a huge oversetimation.
As a conclusion of our discussion, we favor the closed, simple approach.

5 The Closed, Simple Approach

We discard the containment of floating-point symbols in case 2, but we tolerate an overestimation in cases 4 or 7. We can simplify the table.

case	$A = [a_1, a_2]$	$B = [b_1, b_2]$	A/B
1	$0 \in A$	$0 \in B$	$(-\infty, +\infty)$
2	$0 \notin A$	$B = [0, 0]$	$\emptyset$
3	$a_2 < 0$	$b_1 < b_2 = 0$	$[a_2 \check{/} b_1, +\infty)$
4	$a_2 < 0$	$b_1 < 0 < b_2$	$(-\infty, +\infty)$
5	$a_2 < 0$	$0 = b_1 < b_2$	$(-\infty, a_2 \hat{/} b_2]$
6	$a_1 > 0$	$b_1 < b_2 = 0$	$(-\infty, a_1 \hat{/} b_1]$
7	$a_1 > 0$	$b_1 < 0 < b_2$	$(-\infty + \infty)$
8	$a_1 > 0$	$0 = b_1 < b_2$	$[a_1 \check{/} b_2, +\infty)$

Table 2: The eight cases of closed interval division with $A, B \in \mathbb{I}\mathrm{R}$, and $0 \in B$.

Remark 8 *The C++ proposal [2] uses the same division table. The approach replaces the interval evaluation by the so-called range closure.*

5.1 Exception-free Arithmetic

As we stated above a floating-point interval is a set of real numbers where the endpoints are floating-point numbers. A floating-point interval thus is completely different from a floating-point number that usually denotes a more or less crude approximation of a real number. We interpret the bounds of an interval as sharp in the sense that lower or upper bounds are true bounds and do not carry some rounding noise in the relevant direction. Therefore it is not recommended to provide mixed operations between floating-point numbers and intervals. A sophisticated user, however, may define those operations, either by overloading the operators or, preferably, by explicitly invoking a constructor. If we follow these rules, we can show that `NaNs` or signed zeros do not need a special treatment, since they will never occur and the infinity symbols are only used to describe sets, i.e. intervals.

Definition 6 *We consider the system of (extended) floating-point intervals* $\mathbb{I}R := \{[a_1, a_2] \mid a_1 \leq a_2\} \cup \{[a_1, +\infty) \mid a_1 < +\infty\} \cup \{(-\infty, a_2] \mid a_2 > -\infty\} \cup \{(-\infty, +\infty)\} \cup \{\emptyset\}$ *Note that* a_1, a_2 *are floating-point numbers but the set definitions are to be read for all real numbers.*

Theorem 4 *The system* $\mathbb{I}R$ *is closed under the 4 basic operations given by the following tables.*

The proof of the theorem may be picked from the tables, see also [5, 6],

6 Elementary Functions

Interval versions of elementary functions must deliver an enclosure of the real range. Least bit accurate versions have been proposed in [2]. The rely on the same functions as those floating-point functions in the IEEE-754 arithmetic standard.

Definition 7 *A function* $\mathbf{f}$ *is loosely evaluated over an interval* X*, if* $\mathbf{f}(X) := \mathbf{f}(X \cap \overline{D_f})$ *where* D_f *is the domain of* $\mathbf{f}$.

Remark 9 *A discontinuousIntervalFunction exception has to be raised, if a function* $\mathbf{f}$ *is loosely evaluated over an interval* X *with* $X \nsubseteq D_f$*. A corresponding flag [7] has to be set.*

- The default handling in this case should be to terminate.
- There is an instruction to read that flag. Hence, user defined actions can be executed.
- An alternative may be to ignore the exception.

The flag indicates that applications which rely on the continuity of the functions like verification algorithms using Brouwer's fixed-point theorem are not allowed.

7 Conclusion and Further Topics

The closed definition of interval operations is mathematically sound and fulfills the priciples of containment and inclusion isotonicity. Under the assumption that no external (hardware) event changes the data, we can guarantee that all intervals produced are valid intervals.
One may argue that we loose information, when we overstimate the union of 2 infinite intervals by the whole line, but this information can always be explicitly computed by 2 floating-point divisions. The interval newton method needs an a priori test whether the denominator contains zero, and then the finite bounds of the 2 infinite interval can be determined. When, on the other hand, we deliver the 2 quotients as an improper interval, we have to check for this situation after the division and produce the 2 subintervals.
We see that programming the interval Newton method needs specific operation in any case.
In this position paper we, therefore, propose a definition of extended interval arithmetic that is closed and mathematically sound. It should be taken as the core of the coming interval arithmetic standard.
The standard should also specify set operations and comparisons as well as elementary functions. For the latter a discontinuity flag shall be defined that supports the loose evaluation.
Further topics of the standard shall be complete arithmetic including an optimal dot-product.

References

[1] Alefeld, g. and Herzberger J. *Introduction to Interval Computation*, Academic Press 1983

[2] Brönnimann, H., Melquiond, G., Pion, S. *A Proposal to add Interval Arithmetic to the C++ Standard Library (revision 2)* `http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n2137.pdf`

[3] Lerch,M. Hofschuster, W. et al *FILIB++, a Fast Interval Library Supporting Containment Computations*, ACM TOMS Vol 32 No 2, pp.299 - 324, 2006

[4] Kulisch, U: *Advanced Arithmetic for the Digital Computer, Interval Arithmetic Revisited*, Jan 2001

[5] Kulisch, U: *Computer Arithmetic in Theory and Practice*, new edition, to be published, 2008

[6] Kulisch, U: *Letter to the IEEE-754 committee concerning Interval Arithmetic* , Jan 2008

[7] Pryyce, J. *Project Authorisation Request for an Interval Standard Study Group*, Feb 2008

[8] Pryyce, J., Corliss, G. *Interval Arithmetic with Containment Sets* Computing 4, 2005

[9] Walster, G.W. et al.: *The "Simple" Closed Interval System*, Sun Microsystems, Feb 2000