Survey of Proposals for the Standardization of Interval Arithmetic

Nathalie Revol
INRIA
Univ. Lyon, LIP (CNRS-ENS Lyon-INRIA-UCBL), France
Nathalie.Revol@ens-lyon.fr

SWIM, Montpellier, 19-20 June 2008

Context

Definition of interval arithmetic:

- definition of interval arithmetic by Moore: 1966
- modal arithmetic by Gardenes et al.: 1985
- extended interval arithmetic by Ratz: 1996
- definition based on a set point of view: Jaulin et al., 2001
- implementation using floating-point arithmetic: Hickey, Ju and van Emden, 2001
- definition based on limits: cset theory: Walster, Hansen and Pryce, 2002

Context

Definition of interval arithmetic:

- taking into account the existence of complex results: Verdonk et al., 2005
- Fortran: in the 90s
- C++: Brönnimann, Melquiond and Pion, 2006
- hardware support: Kirchner and Kulisch, 2006
- . . .

Context: IEEE P1788 for the standardization of IA

Dagstuhl, January 2008: decision to produce a standard for interval arithmetic.

Also: decision to have a standard under the auspices of IEEE.

Spring 2008: under the sponsorship of the IEEE committee for floating-point arithmetic, proposal of a **working group for the standardization of interval arithmetic**, approved by IEEE the 12 June 2008, under the number P1788.

Context: IEEE P1788 for the standardization of IA

Dagstuhl, January 2008: decision to produce a standard for interval arithmetic.

Also: decision to have a standard under the auspices of IEEE.

Spring 2008: under the sponsorship of the IEEE committee for floating-point arithmetic, proposal of a **working group for the standardization of interval arithmetic**, approved by IEEE the 12 June 2008, under the number P1788.

Do not hesitate to join!

Outline of this talk

Initial definition: Moore 1966

Initial definition by Moore (1962, published in 1966):

- $[a, b] + [c, d] = [a + c, b + d]$;
- $[a, b] - [c, d] = [a - d, b - c]$;
- $[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$;
- $1 \,/\, [c, d] = [1/d, 1/c]$ if $0 \notin [c, d]$;
- $[a, b] \,/\, [c, d] = [a, b] \times (1/[c, d])$ if $0 \notin [c, d]$;
- $f([a, b]) =$ convex hull $(\{f(x) : x \in [a, b]\})$:
  formulas using only the endpoints when f is monotonous,
  more complicated otherwise.

Unsatisfying definition

Division is not **total**: $[1, 2]/[-1, 2]$???

The system is not **closed**.
It is desirable that every possible combination of $<$ `operator`, `operands` $>$ yields a result within the system.

Extended interval arithmetic

Ratz 1996

(or maybe Kahan or Hanson in 1968)
Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two intervals.

$$\boldsymbol{x}/\boldsymbol{y} = \{z \,:\, y \cdot z = x,\ x \in \boldsymbol{x}, y \in \boldsymbol{y}\}.$$

Extended interval arithmetic
Division by an interval containing 0

Main concern: Newton iteration to solve $f(x) = 0$ without losing any solution.

Proposals:

- Jaulin et al.: $1/[-2, 2] = (-\infty, +\infty)$ but $[3, 4]/[0, 0] = \emptyset$;
- $[0, 1]/[0, 1] = [0, +\infty)$ since only nonnegative terms can be produced (Ratschek & Rokne 1988);
- $[1, 2]/[0, 1] = \{-\infty\} \cup [1, +\infty]$ (cset theory)
- $[0, 1]/[0, 1] = (-\infty, +\infty)$ (Ratz)

Remark: arguments outside the domain

More generally, how should $f(\boldsymbol{x})$ be handled when $\boldsymbol{x}$ is not included in the domain of f?

- return NaI (Not an Interval)? Ie. handle exceptional values such as NaI and infinities?
- intersect $\boldsymbol{x}$ with the domain of f prior to the computation, silently?
- intersect $\boldsymbol{x}$ with the domain of f prior to the computation and raise a flag?
- return the set of every possible limits $\lim_{y \to x} f(y)$ for every possible x in the domain of f (but not necessarily y)?

Outline of this talk

Forward and backward

Wording inspired from constraint programming and forward-backward propagation.

Forward and Backward

Forward:

it corresponds to the "natural extension" à la Moore.

$$f(\boldsymbol{x}) = \{f(x) \ : \ x \in \boldsymbol{x}\}$$

or

$$f(\boldsymbol{x}) = \{\lim_{x \to y} f(x) \ : \ y \in \boldsymbol{x}\}$$

or

$$f(\boldsymbol{x}) = \{\lim_{x \to y} f(x) \ : \ x \in \boldsymbol{x}, y \in \boldsymbol{x}\}$$

Forward and Backward

Backward:

it corresponds to the philosophy of Ratz: one does not want to lose any solution.

$$f(\boldsymbol{x}) = \text{convex hull}(\{y : \exists x \in \boldsymbol{x}, f^{-1}(y) = x\})$$

Eg.

$$\sqrt{[1, 2]} = \text{convex hull}([-\sqrt{2}, -1] \cup [1, \sqrt{2}]) = [-\sqrt{2}, \sqrt{2}].$$

I personally prefer the wording **relations** to **backward operations**, since I would also prefer to keep the two separate parts of the answer and thus $\sqrt{\ }$ is no more a function, since it returns two arguments, but it is a relation.

Outline of this talk

With or without the infinities?

Should we work with $IR = (-\infty, +\infty)$ or with $\bar{IR} = [-\infty, +\infty]$?
Should the infinities be first class citizens or outlaws?

With or without the infinities?

Should we work with $IR = (-\infty, +\infty)$ or with $\bar{IR} = [-\infty, +\infty]$?
Should the infinities be first class citizens or outlaws?

If they are first class citizens, $[0, 1]/[0, 1] = \{-\infty\} \cup [0, +\infty]$ (cset theory) becomes natural.

Complex results. . .

Verdonk, Vervloet, Cuyt 2005

Proposal: add flags to indicate whether there could also exist complex results, nonzero complex results. . .

Definition related to floating-point arithmetic

Lozinski 1973, MPFI

Implementation based on IEEE-754 floating-point arithmetic.

Point of view: also based on floating-point arithmetic:

$$f(\boldsymbol{x}) \; = \; \{f(x) \in I\!F \; : \; x \in \boldsymbol{x} \text{ and } x \in I\!F\}.$$

Eg. $\sqrt{[-1,4]} \supset [0,2] \cup \{ \text{ NaN } \}$ and thus $\sqrt{[-1,4]} =$ NaI (Not an Interval).

Hickey, Ju, van Emden 2001

Definition based on the set of reals $IR = (-\infty, +\infty)$.
Interval = closed connected set in IR, ie. one of $\emptyset$, $(-\infty, b]$, $[a, +\infty)$ or $[a, b]$ where $a \in IR$ and $b \in IR$.

Clever implementation using IEEE-754 floating-point arithmetic:

- infinities exist and can be handled;
- use of signed zeroes: $[0, 1]$ is represented as $[+0, 1]$ and thus $[0, 1]/[0, 1]$ naturally yields $[0, +\infty)$.
  Idea: $[0, 1]$ contains only nonnegative numbers and is almost certainly too wide, ie. the exact result may well contain only positive numbers.
  No non-standard analysis (with infinitesimally small numbers between 0 and any positive number).

Outline of this talk

Wraparound intervals

Kulisch: $[3, 4]/[-2, 1] = (-\infty, -2] \cup [3, +\infty)$
To return only one result, return $[3, -2]$.

Markov: $[a, b] + [-a, -b] = [0, 0]$
Algebraic structure (group instead of simply a monoid) is recovered.

Modal arithmetic

Gardenes, Mielgo and Trepat, 1985

Goldsztejn 2005, Shary...

Idea: an improper interval $\boldsymbol{x}$ in an operation is interpreted as $\{\exists x \in \boldsymbol{x} \ : \ \dots\}$.
Restriction: every $\forall$ quantifier must appear before $\exists$ quantifiers in the interpretation.

Outline of this talk

List of operations

- arithmetic operations and functions: $+$, $-$, $\times$, $/$, $\sqrt{}$, power (more tricky), elementary functions, special functions?
- set operations: $\cap$, $\cup$, convex union, $\setminus$
- interval operations: inf, sup, mid, width or radius...

Comparisons

At least three possible definitions:

- **certainly** $<, \leq, >, \geq \ldots$:

$$\boldsymbol{x} < \boldsymbol{y} \Leftrightarrow \forall x \in \boldsymbol{x}, \forall y \in \boldsymbol{y}, x < y$$

- **possibly** $<, \leq, >, \geq \ldots$:

$$\boldsymbol{x} < \boldsymbol{y} \Leftrightarrow \exists x \in \boldsymbol{x}, \exists y \in \boldsymbol{y}, x < y$$

- **Kulisch** $<, \leq, >, \geq \ldots$:

$$\boldsymbol{x} = [\underline{x}, \bar{x}] < \boldsymbol{y} = [\underline{y}, \bar{y}] \Leftrightarrow \underline{x} < \underline{y} \text{ and } \bar{x} < \bar{y}.$$

Algebraic manipulations of expressions

Should we allow the compiler to manipulate the expressions to optimize the computational time?
Forbidden in pure IEEE-floating point mode, because the usual algebraic rules do not apply to floating-point computations.

Ibid. for interval expressions?
What about algebraic manipulations by the user (yielding different results)?

Outline of this talk

Conclusion

Wanted: a standard where

- the system is closed, ie. any operation between any operands results in an element of the system;
- its implementation, using floating-point arithmetic, is closed;
- everything is mathematically sound:
- *Thou shalt not lie*: the inclusion property is valid;
- the implementation is easy and efficient (even if hardware implementation is not required, furthermore some points are language-dependent);
- it is easy to implement other mathematical models (wraparound intervals, modal arithmetic...).

Future work

The IEEE committee will have to

- complete this list

Future work

The IEEE committee will have to

- complete this list
  and you can help us!
- discuss every point, its pro and cons (using counterexamples)

Future work

The IEEE committee will have to

- complete this list
  and you can help us!
- discuss every point, its pro and cons (using counterexamples)
  and you can help us!
- agree on the most sensible choice...

Future work

The IEEE committee will have to

- complete this list
  and you can help us!
- discuss every point, its pro and cons (using counterexamples)
  and you can help us!
- agree on the most sensible choice. . .
  and then you will vote to tell us if we were right!

See you in 4 (or 6, or 8) years time, to introduce you the new standard!

To join IEEE P1788

Send me: **Nathalie.Revol@ens-lyon.fr** an e-mail with

- your first name and name
- your affiliation
- your complete adress
- your e-mail address
- whether you plan to subscribe to the mailing list or to serve on the committee.

Serving on the committee: 3-4 meetings per year, 3 days each, alternately in Europe and North America (very probably).