IEEE Working Group P1788: A Standard for Interval Arithmetic

Nathalie Revol
INRIA - LIP (UMR 5668 CNRS - ENS Lyon - INRIA - UCBL)
Université de Lyon

based on a talk given for an EVA-Flo meeting, September 2009
and a talk given by John Pryce (Technical Editor, P1788)
at Dagstuhl Seminar 09471, 15–20 November 2009

Réunion MEA, 3 décembre 2009

Agenda

Agenda

The P1788 Project

- Started by 15 attenders at Dagstuhl, Jan 2008.
- Project Authorisation Request (PAR) to IEEE, spring 2008.
- IEEE authorise it as IEEE-WG-P1788, 12 Jun 2008
- Executive committee elected, El Paso, Sep 2008:

Chair, Vice-chair	: Nathalie Revol, Baker Kearfott
Secretary	: William Edmonson
Technical Editors	: John Pryce, Christian Keil
Archivist	: Guillaume Melquiond
Webmaster	: Jürgen Wolff von Gudenberg
Vote Tabulator	: George Corliss

Purpose

- From PAR: "... to improve the availability of reliable computing in modern hardware and software environments by defining the basic building blocks needed for performing interval arithmetic."
- Our responsibility is below "programming language level", but P1788 will make recommendations on language standards for intervals.

How P1788's work is done

- The bulk of work is carried out by email, with electronic voting.
- Motions are proposed, seconded; three weeks discussion period; three weeks voting period.
- IEEE has given us a four year deadline (not cast in stone).
- One "in person" meeting per year is planned—last was at PPAM 09, Wroclaw, Poland, 13–16 Sept 2009.

IEEE-1788 WG: some facts

Since October 2008: **very active mailing list**
over 140 participants, over 20 nationalities
over 1500 messages

Work already done:
adoption of officers, of procedures and policy
roster of voting members: 74 members, 14 nationalities
vote on 7 motions: passed
2 (3) motions withdrawn
currently: 1 motion in discussion.

IEEE auspices: 1 report + 1 teleconference quarterly

Motions so far

Motion 10 is still in voting period.

#1. (Pryce/Kreinovich) Provisional standard notation for intervals

#2. (Pryce/Corliss) Levels structure for standardisation process

#3. (W. von Gudenberg/Hack) Standard is based on $\mathbf{R}$ not $\mathbf{R}^*$

#4. (Pryce/Zuras) Restrict standard to 754 systems *(withdrawn)*

#5. (Kulisch/Einarsson) Tables for arithmetic operations

#6. (Pryce/Zuras) Multi- & mixed-format interval support

#7. (W. von Gudenberg/Cunha) Have just one NaI *(withdrawn)*

#8. (Hayes/Kreinovich) Decorations: a way to handle exceptions

#9. (Kulisch/Einarsson) Exact dot product

#10. (W. von G./Kreinovich) Supported elementary functions

Agenda

Agenda

Motion 1: Standardized notations

Cf. "Standardized notation in interval analysis" by R.B. Kearfott, M.T. Nakao, A. Neumaier, S.M. Rump, S.P. Shary, and P. van Hentenryck,
http://www.mat.univie.ac.at/~neum/papers.html
(open to amendment after sufficient experience of using it)

A *box* of dimension n is a pair $\boldsymbol{x} = [\underline{x}, \overline{x}]$ consisting of two real column vectors $\underline{x}$ and $\overline{x}$ of length n with $\underline{x} \leq \overline{x}$. The set of all boxes of dimension n is denoted by $\mathbb{IR}^n$.

$f([-1,1])$ denotes the image of $[-1,1]$ under f; f$([-1,1])$ specifies it as the result of applying the operations in f to the interval $[-1,1]$ in its intrinsic arithmetic.

Motion 2: Levels Structure of P1788

Issue: manage complexity.
Following good software engineering practice, we decided to arrange the standard in 4 levels, or layers:

- Level 1: mathematical level
- Level 2: datum level: *IF* the set of machine intervals
- Level 3: representation level ($\emptyset$, NaI,...)
- Level 4: bit strings

Motion 2: Levels structure of P1788

Level 1. Mathematical objects. Intervals are subsets of **R**.
Operation on intervals $\boldsymbol{x}, \boldsymbol{y}$ defined by
$\boldsymbol{x} \bullet \boldsymbol{y} = \text{hull}\{\, x \bullet y \mid x \in \boldsymbol{x},\ y \in \boldsymbol{y},\ x \bullet y \text{ exists} \,\}$.

Level 2. Finite set of *machine intervals* (interval datums).
$\boldsymbol{x} \bullet \boldsymbol{y}$ defined as "machine hull" of ($\boldsymbol{x} \bullet \boldsymbol{y}$ at level 1).
Decorations for exception handling—see later—exist at this level.

Level 3. Representation. Exposes the variables in terms of which interval operations are implemented.

Level 4. Encoding. The level of bit strings and memory locations.

Motion 2: Levels structure of P1788

Relationships between different specication levels: interval version

Level 1	Number system $\mathbb{R}$. Set $\mathbb{IR}$ of allowed intervals over $\mathbb{R}$. Principles of how $+ - * /$ and standard functions are extended to intervals.	Mathematical Model level. No NaN or NaI at this level.
many-to-one	↓ interval hull identity map, except NaI ↑	one-to-one
Level 2	The set $\mathbb{F}$ of "machine intervals" in $\mathbb{IR} \cup \{$NaI, if used $\}$	Interval data level.
one-to-many?	↓ representation specification ↑	many-to-one?
Level 3	Representation of nonempty $[\underline{x}, \bar{x}]$ as two FP numbers $\underline{x}$, $\bar{x}$ or alternative. Representation of $\emptyset$ and NaI.	Representations of interval data.
one-to-many?	↓ encoding specication ↑	many-to-one?
Level 4	0111000 . . .	Bit strings.

Agenda

Motion 3: Intervals are sets of reals

Issue: What is an interval, anyway?

Intervals are connected closed sets of reals
(and not of extended reals, i.e. infinites are not included)
(no reversed interval either).

They comprise the empty set.

Motion 3: Consequences

This **excludes** some alternatives:

- Nonstandard intervals (Kaucher/modal): intervals are not sets, but formal objects, ordered pairs $[\underline{x}, \overline{x}]$ of real numbers, allowing $\underline{x} > \overline{x}$.
  Nonstandard intervals *can* be supported as an add-on. . .
- Interval operations are defined purely algebraically (no limits): containment set (cset) theory, where intervals are subsets of extended reals $\mathbf{R}^*$, and operations defined by limits are quite incompatible with the standard.
- Support for open and half-open intervals.
- Complex intervals.

Motion 5: Arithmetic Operations

mathematical definition
implementation/formulas not concerned

proposed by U. Kulisch.

Division by 0:

$$[a_1, a_2]/[b_1, b_2] = (-\infty, +\infty)$$
when $a_1 < 0 < a_2$ and $b_1 < 0 < b_2$

Motion 5: satisfying definition

Division is **total**: $[1,2]/[-1,2] = \mathbf{R}$

The system is **closed**.
It is desirable that every possible combination of $<$ `operator`, `operands` $>$ yields a result within the system.

Digression about 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)

Motion 5: 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?
- 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)?
- 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 mention it

Remark: arguments outside the domain

Problematic example (Rump, Dagstuhl seminar 09471, Nov. 2009).

$$\begin{aligned} f(x) &= |x-1| \\ g(x) &= (\sqrt{x-1})^2 \text{ I know, it is not the best way of writing it...} \end{aligned}$$

What happens if $\boldsymbol{x} = [0,1]$?
With the adopted definitions of operations,

$$\begin{aligned} f(\boldsymbol{x}) &= [0,1] \\ g(\boldsymbol{x}) &= [0] \end{aligned}$$

The Thou shalt not lie principle is not valid... and this is a problem with so-called validated computations.

Agenda

Motion 4: Restriction to IEEE-754 Conforming Environments

Interval arithmetic IEEE-1788 is defined only for systems that support floating-point arithmetic IEEE 754-2008.
Implementation on non-conforming systems is out of scope.

withdrawn.

Motion 6: Conversion between IEEE-754 Floating-Point Formats

Multi-format and conversions between floating-point formats:
single, double or extended binary formats,
binary / decimal formats.

Remark: it is agreed that the representation of an interval, based on two floating-point numbers (endpoints), uses the same format for both.

passed

Agenda

Motion 9: Exact Dot Product

The standard requires the (hardware?) implementation of an exact dot product for vectors of floating-point numbers.

passed

Motion 9: Exact Dot Product

The standard requires the (hardware?) implementation of an exact dot product for vectors of floating-point numbers.

passed with a small majority.

Motion 9: Exact Dot Product

Discussion:

- hardware or software implementation?
  based on a long accumulator: cost (in surface, bus width...)
  in hardware? Competite algorithms exist (cf. Rump).
- exact dot product
  or faithfully rounded dot product
  or dot product almost always correctly rounded but with a maximal error up to 2 ulps?
- should this be standardized with interval arithmetic? with the BLAS?
- should the dot product only be standardized?
  What about BLAS2 routines? BLAS3 routines?

Motion 10: Elementary Functions

Two sets of functions

- a normative list of required functions: sqr, pown, pow, sqrt, exp, exp2, exp10, log, log2, log10, expm1, exp2m1, exp10m1, logp1, log2p1, log10p1, sin, cos, tan, asin, acos, atan, atan2, sinh, cosh, tanh, asinh, acosh, atanh, abs, rSqrt, hypot, compoundm1;
- a non-normative list of recommended functions: sign, ceil, floor, nint, trunc, rootn, powr, sinPi, cosPi, atanPi, atan2Pi, exp1x, exp2x, cos2, sin3, cosh2, sinh3, gamma, lgamma, erf, erfc.

Motion 10: Elementary Functions

Mandatory functions comprise the usual set of math standard functions provided in languages and libraries for scientific applications.
The selection of functions was taken from IEEE 754-2008 standard, the Vienna proposal, and from the C++ interval extension approach.

Motion 10: still under discussion

Agenda

Motion 7: A Unique NaI

one unique NaI

lively discussion, that led to the withdrawal of motion 7 and the proposal of motion 8

Motion 7 and 8: Exceptions

Issue: How to handle exceptions efficiently.

- Typical examples:
  (a) Invalid interval constructor
  `interval(3,2)` `interval("[2.4,3;5]")`
  —interface between interval world and numbers or text strings.
  (b) Elementary function evaluated partly or wholly outside domain
  `sqrt([-1,4])` `log([-4,-1])` `[1,2]/[0,0]`
- Type (a) can simply cause nonsense if ignored.
- Type (b) are crucial for applications that depend on fixed-point theorems; but can be ignored by others, e.g. some optimisation algorithms.

Motions 7 and 8: Exceptions, cont.

What to do? A complicated issue.

- Risk that (Level 3) code to handle exceptions will slow down interval applications that don't need it.
- One approach to type (a) is to define an NaI "Not an Interval" datum at level 2, encoded at level 3 within the two FP numbers that represent an interval.

Motion 8: Exceptions by Decorations

- Alternative (Motion 8): An extra tag or decoration field (1 byte?) in level 3 representation.
- Divided into subfields that record different kinds of exceptional behaviour.
- Decoration is optional, can be added and dropped.
  – To compute at full speed, use "bare" intervals and corresponding "bare" elementary function library.
  – "Decorated" library records exceptions separate from numbers, hence code has fewer IFs & runs fast too. (We hope!)

Motion 8: Decoration issues

Decorations look promising but many Qs exist:

- Bare (double) interval is 16-byte object. Decoration ups this to 17. Values are trits (three values: OK, pb, undefined). Can compilers efficiently allocate memory for large arrays of such objects?
- Some proposed decoration-subfields record events in the past; others are properties of the current interval. Can semantic inconsistencies arise?
- Can decoration semantics be specified at Level 2 . . .
- . . . such that correctness of code can be proven . . .
- . . . and K.I.S.S. is preserved?

Current Motion 8 is just a start, much work on exceptions remains.

Remark: arguments outside the domain

Problematic example (Rump, Dagstuhl seminar 09471, Nov. 2009).

$$\begin{aligned} f(x) &= |x-1| \\ g(x) &= (\sqrt{x-1})^2 \text{ I know, it is not the best way of writing it...} \end{aligned}$$

What happens if $\boldsymbol{x} = [0,1]$?
With the adopted definitions of operations,

$$\begin{aligned} f(\boldsymbol{x}) &= [0,1] \\ g(\boldsymbol{x}) &= [0] \end{aligned}$$

One has to check whether the decoration mentions a *possibly undefined* operation...

Remark: arguments outside the domain

Problematic example (Rump, Dagstuhl seminar 09471, Nov. 2009).

$$\begin{aligned} f(x) &= |x-1| \\ g(x) &= (\sqrt{x-1})^2 \end{aligned}$$ I know, it is not the best way of writing it...

What happens if $\boldsymbol{x} = [0,1]$?
With the adopted definitions of operations,

$$\begin{aligned} f(\boldsymbol{x}) &= [0,1] \\ g(\boldsymbol{x}) &= [0] \end{aligned}$$

One has to check whether the decoration mentions a *possibly undefined* operation...
but unexperienced programmers will not do it.

Agenda

Other topics

Several position papers (Vienna proposal, Kulisch' proposal, modal intervals)

Many topics of discussion. . . with or without conclusion:

- out of range values
- division by 0
- interval comparison
- . . .

Agenda

Is there a P1788 master plan?

- Yes and no. Any voting member may propose a motion any time. So far the results have been fairly coherent.
- Vienna Proposal for Interval Standardization (December 2008) led by Arnold Neumaier is probably close to P1788's final functionality, so is a good template for our deliberations.
- We are working on draft text of that part of the standard that we have already decided, hopefully out by Christmas 2009.
- This will be voted on, in chunks, and progressively added to.
- As Tech. Ed., John Pryce envisages a few "global revisions", to check the whole text for consistency, simplicity and correct priorities.
- Nothing is cast in stone till after the Sponsor Ballot stage.

Is there a P1788 master plan?

As Chair, I envisage encouraging the next motions to be about basic definitions:

- comparisons
- interval operations (inf, sup, mid, rad...)
- set operations (union, intersection, hull...)
- relations (forward vs backward modes),

Agenda

IEEE 1788 WG: in short

Web site: http://grouper.ieee.org/groups/1788/
(or google "IEEE 1788").

To participate:

- subscribe to the mailing list
- subscribe to the voting roster

To vote:

- be a member of the voting roster
- vote YES or NO: no need to justify a "NO" vote
- no expelled member after failing to vote twice in a row

Standard expected in December 2011...
I hope we will have produced at least a substantial part of it!-)

Future work

The P1788 committee and membership will have to

- complete the list of tasks

Future work

The P1788 committee and membership will have to

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

Future work

The P1788 committee and membership will have to

- complete the list of tasks
  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 P1788 committee and membership will have to

- complete the list of tasks
  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 WG P1788

Send to: Chair Nathalie.Revol@ens-lyon.fr or to Vice-Chair R. Baker Kearfott rbk@louisiana.edu an e-mail with

- your first name and name
- your affiliation
- your complete adress
- your e-mail address
- whether you plan to just subscribe to the mailing list or also to be an active (voting) member of the working group.