2009 19th IEEE International Symposium on Computer Arithmetic

IEEE Interval Standard Working Group - P1788: Current Status

William Edmonson
Dept. of Electrical & Computer Eng.
NC State University
Raleigh, NC 27695
wwedmons@ncsu.edu

Guillaume Melquiond
INRIA Saclay–Île-de-France
Orsay, F-91893 France
guillaume.melquiond@inria.fr

Abstract

Late 2008, at SCAN 2008 in El Paso, TX, an effort to standardize interval computations was started by a working group of the IEEE Microprocessor Standards Committee, titled the Interval Arithmetic Working Group of the IEEE P1788 Standard. This paper describes the goals of this effort, the history of the working group, and how it relates to the IEEE 754 Standard. It gives a brief overview of the policies and procedures for constructing the standard, and its expected structure. It also presents some of the questions the group may have to solve in the future.

1. Philosophy of Interval Computation

Interval computations [1], known as either *Interval Analysis* or *Interval Arithmetic* (both abbreviated as IA), are a way of extending the usual arithmetic from real numbers to intervals, which typically are closed convex sets of real numbers. The operations on these sets are defined in such a way that the *inclusion property* is respected: For $\mathbf{x}$ and $\mathbf{y}$ intervals, for $\diamond \in \{+, -, \times, \div, \ldots\}$, we have

$$\forall x \in \mathbf{x},\ \forall y \in \mathbf{y}, \quad x \diamond y \in \mathbf{x} \diamond \mathbf{y}.$$

Here $x \diamond y$ is the mathematical exact result while $\mathbf{x} \diamond \mathbf{y}$ is an interval possibly computed with finite-precision arithmetic. This property ensures that an expression evaluated by intervals returns an enclosure of the set of all the possible real results, irrespective of the imprecision on the inputs or of the inaccuracies during intermediate computations. Thus interval computations are one of the main components for reliable computations.

We enumerate a few applications here, while noting that this illustrative list is in no way exhaustive:

- global optimization (*e.g.*, finding optimal solutions of multi-dimensional not-necessarily-convex problems);
- accounting for rounding errors of floating-point computations at run time;
- constraint propagation for solving satisfiability problems;
- solving [systems of] [linear] [differential] equations using interval analysis;
- mathematical proofs (*e.g.*, Hales' recent celebrated proof of Kepler's conjecture).

1.1. Goals of the Standards

The standard's broad aim is to improve the availability of reliable computing in modern hardware and software environments by defining the basic building blocks needed for performing interval arithmetic. It will facilitate the incorporation of a conforming interval data type in languages used for numerical computing. Its specific purpose in support of the above aim is to decide on a specific model, then define, on interval datums, basic operations that are provably consistent with the chosen model.

More precisely, a usable standard for interval arithmetic can cover some of the following points at least:

- definition of the interval formats, including the encoding of special interval values such as *empty*;
- interval constructor operations, including the effect of passing operands that do not define a valid interval in the model;
- operations to return the lower and upper bounds of an interval, and its midpoint and radius, including the effect in special cases such as the empty interval;
- basic arithmetic operations, including division by an interval containing zero as well as square root;
- interval comparison operations that are deemed important, such as interval containment;
- discontinuity flag or other appropriate mechanism for marking discontinuity detection during computation (needed by certain algorithms that rely on fixed point theorems).

There are several well-developed IA systems based on self-consistent but different philosophies and resulting models. All widely used models agree on the definition of arithmetic operations when (a) intervals are closed and bounded and (b) the real operation is defined for all the real values included in the input intervals. The differences are at the "edges", *e.g.*, infinite bounds or division by an interval containing zero. Not all widely used models will necessarily be supported in the standard, but it is important

1063-6889/09 $25.00 © 2009 IEEE
DOI 10.1109/ARITH.2009.36

that all of them be straightforward to implement on top of the standard, to encourage its widespread acceptance.

1.2. Current Implementation

Advances in computer arithmetic since Ramon E. Moore published his book, Interval Analysis [1], have resulted in numerical analysis being performed by interval arithmetic operations to account for bounding the errors in the data or model or taking into account the uncertainties of the system or model. Today, interval analysis has gained acceptance in many fields, such as signal processing [4], robotics control [5], and computer graphics [6], to name just a few. For a more complete list of interval analysis applications and associated references, we recommend that you visit http://www.cs.utep.edu/interval-comp/.

To meet the above needs to perform interval arithmetic and interval analysis, several software solutions have been developed either as complete packages or as libraries for typical programming languages. Several widely used packages are listed in the following table, with more packages listed at http://www.cs.utep.edu/interval-comp/intsoft.html.

Table 1. Interval Arithmetic Software Packages

Interval Packages		
Program	Definition	URL
Profil/BIAS	C++ class library	http://www.ti3.tu-harburg.de/knueppel/profil/index_e.html
boost	Interval C++ package	http://www.boost.org/
filib++	Interval library	http://www.math.uni-wuppertal.de/~xsc/software/filib.html
Intlab	Matlab toolbox	http://www.ti3.tu-harburg.de/rump/intlab/index.html
Sun Studio	C++/Fortran compiler	http://www.sun.com/software

In addition, hardware prototypes that have been investigated by Kirchner and Kulisch [3] and Schulte [9], [10] are floating-point architectures for implementing interval arithmetic operations, where the later has been extended to provide variable precision arithmetic. A fixed-point interval arithmetic ALU was developed by Edmonson and his students [11], [12]. In the work of [12], the interval fixed-point arithmetic ALU was extended to an interval block floating-point ALU for an increased dynamic range. The major impetus for architecting a fixed-point arithmetic version was to develop an interval arithmetic computing engine that can be used in embedded systems which require low power, small area, and real time operations.

1.3. Relation to 754 Standards

The wide acceptance of the IEEE 754 Standard has had a significant impact on the development of reliable environment for computing. It provides portable formats, but the description of the operations and modes do not leave much to imagination or luck. More importantly for interval arithmetic, it mandates the presence of directed rounding modes which apply to all the basic floating-point operations.

The interval working group intends to build its standard upon the foundations laid by the IEEE 754 Standard: Interval formats will be based on floating-point formats, and interval operations will be defined in terms of floating-point operations (while keeping a mathematical model based on the real numbers). This will greatly ease the development of a portable and reliable interval arithmetic.

2. History of Working Group

Late 2007, the IEEE 754 revision committee received a proposal from the IFIP Working Group 2.5 on Numerical Software that strongly supported the inclusion of IA to the floating-point standard. Some missing details and a scope vastly exceeding the one of IEEE 754 prevented this proposal from passing. Early 2008, at a Schloss Dagstuhl seminar where members from both working groups met, decision was taken to propose a new working group dedicated to standardization of IA [13].

At the 13th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations (SCAN 2008) in El Paso, TX, the initial organizational meeting for the creation of the IEEE Interval Arithmetic Working Group was held. A presentation on developing a standards group was given by Baker Kearfott followed by several luncheon meetings. In these meetings, the working group's and IEEE's policies and procedures were discussed, along with obtaining the volunteers for officer's position. These meetings were well attended with lively discussions. Those who volunteered subsequently became the official officers of the working group. These officers are listed in the following section.

2.1. Structure of Committee

The following officers were accepted on November 8, 2008, after being elected by the participants registered to the working group:

Chair
: Nathalie Revol

Vice chairs
: Baker Kearfott and John Pryce

Vote tabulator
: George Corliss

Technical editors
Christian Keil, David Lester, and John Pryce
Archivist
Guillaume Melquiond
Webmaster
Juergen Wolff von Gudenberg

A website tracking the active motions and archiving some informational documents is available at http://grouper.ieee.org/groups/1788/.

2.2. Structure for Constructing the Standard

To better facilitate the IEEE standardization process, the technical editors of the P1788 Working Group have prepared a document titled, "A proposed structure for the process of constructing the P1788 Standard" (found at http://grouper.ieee.org/groups/1788/ under position papers). This structure represents a conceptual design by providing four specification levels and their relationships. The structure mirrors the P754 levels. Level 1 represents the *mathematical* model and offers a framework for numerical experts to choose the number system, *i.e.*, real numbers $\mathbb{R}$ or extended reals $\mathbb{R}^*$, intervals $\mathbb{IR}$ containing these reals, interval arithmetic operations, and standard interval functions. Level 2, *data* level, defines the finite set $\mathbb{IF}$ (a subset of $\mathbb{IR}$ consisting of machine-representable intervals) and how to choose function results in this set. Level 3 defines how pairs of floating-point datums are used to represent machine intervals. Level 4 provides the bit representation of the level 3 format.

This position paper also gives a list of questions at each level for which the editors believe there is a need for debate. A small sample follows.

- Level 1 debates
  1) Should the numbering system be the set $\mathbb{IR}$ or the extended set $\mathbb{IR}^*$?
  2) What support should be given to other interval models, *e.g.*, modal or Kaucher?
  3) Do we follow the principles of the Vienna model, see http://grouper.ieee.org/groups/1788/?
- Level 2 debates
  1) Should a "Not an Interval" be defined?
  2) What flags should interval operations set and which P754 flags can be used?
  3) How accurate should interval functions be?
- Level 3 debates
  1) What should the standard say about invalid interval objects?
  2) How should nonempty intervals be represented as?
  3) Should the standard provide a use for NaN fields?
- Level 4 debates
  1) Should the standard specify storage allocation?
  2) Should it provide representations for modal intervals?

2.3. Accomplishments to Date

The working group is still in its early stages and it has not yet produced a draft of an interval standard. The following three decisions have been voted and agreed upon: policies and procedures of the group, committee officers, and mathematical notations for writing the draft. Motion 1 on Standard Notation has been officially passed by the working group. At the time this is written, voting on Motion 2 on Process Structure is still in progress, but passage appears likely. Motion 3 on Set of Reals is in discussion.

3. Controversies

As mentioned above, there is a wide consensus on how an interval operation behaves when the input intervals represent bounded sets and the corresponding real operation is defined at each point of these sets. The following paragraphs describe some of the less consensual issues the group will have to work on.

3.1. Representation Issues

Intervals typically are represented by their lower and upper bounds. In conjunction with directed floating-point arithmetic, this representation allows for efficient basic arithmetic operations. Another representation, however, can be encountered: intervals as pairs midpoint-radius.

Basic operations then become more complicated to design, but algorithms for interval linear algebra can take advantage of this representation. It also breaks the symmetry between the two components: The radius can use a smaller floating-point precision than the midpoint.

The working group must decide if this alternate representation is out of the scope of a standard, if it "should" be provided by compliant interval environment, or if it "shall" be provided.

3.2. Inverted Intervals

While typical intervals are just sets of real numbers, some arithmetics give a meaning to interval objects $[a, b]$ with $a > b$ (or negative radius). For instance, they can be used for storing the complementary sets of (interiors of) regular intervals.

Usually these inverted intervals are a way to extend the set of intervals with an algebraic structure [2]. Both interval addition and multiplication admit inverse operations, *e.g.*, $[a, b] + [-a, -b] = [0, 0]$. This property brings many classical algorithms to interval environments. In some cases,

it also supports computation of an under-approximation of the result, that is, a guaranteed subset of the reachable points.

Again, the working group will have to take a position with respect to this extension. Depending on how inverted intervals are handled (*e.g.*, they could be declared "invalid" and cause a *Not-an-Interval* datum to propagate), some extensions may be incompatible with an interval standard.

3.3. Partial Real Operations

Another issue arises when some intervals contain values for which the real operation is not defined, *e.g.*, computing the quotient $[1,1]/[0,1]$. If the divisor was instead $[\varepsilon,1]$, the tightest result would indisputably be $[1,1/\varepsilon]$. Once ε reaches zero however, there are several interpretations of the interval division.

If one is only interested in the set $\{z \mid z = x/y, x \in \mathbf{x}, y \in \mathbf{y}\}$, then $[0,+\infty)$ is the optimal result. If one considers the division as the inverse operation of the multiplication, then the result set becomes $\{z \mid x = y \times z, x \in \mathbf{x}, y \in \mathbf{y}\} = \mathbb{R}$. This division could also be considered as an error that should cause an exceptional behavior instead of continuing silently with an interval value.

The working group will have to decide which versions should be part of a standard (or all of them). Since knowing if the input intervals contained out-of-domain or discontinuity points is critical for interval methods that rely on fixed-point theorems, the group will also have to design a way to get this information to the user.

Independently of the considerations above, fully-fledged inverse versions of the arithmetic operations are needed when propagating and solving interval constraints. So the group will have to discuss the opportunity of including them.

3.4. Comparisons

Comparing intervals is a last example of controversial issues. Depending on the context, $\mathbf{x}$ smaller than $\mathbf{y}$ may mean either "$\mathbf{x} \subsetneq \mathbf{y}$", or "$\forall x \in \mathbf{x},\ \forall y \in \mathbf{y},\ x < y$", or "$\mathrm{lower}(\mathbf{x}) < \mathrm{lower}(\mathbf{y}) \wedge \mathrm{upper}(\mathbf{x}) < \mathrm{upper}(\mathbf{y})$".

The working group will decide if any/some/all of these comparison operators are worth standardizing and they will have to be named.

4. Conclusion

The IEEE Interval Arithmetic Working Group P1788 was formulated to develop a set of standards that is similar in structure and built on the philosophy of the IEEE P754 floating-point standard. The result of this standardization process for interval arithmetic is to have a set of community accepted standards that will increase the availability of reliable computing in modern hardware and software environments. As of March 2009, there are 62 officially registered voting members. Information on the P1788 Standard can be found at http://grouper.ieee.org/groups/1788/.

Acknowledgment

The authors would like to thank all of the P1788 committee for their assistance in the writing of the status report and for putting the time and effort in bringing interval arithmetic to the forefront.

References

[1] R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[2] E. Kaucher. "Interval Analysis in the Extended Interval Space $\mathbb{IR}$," *Computing*, Suppl 2:33–49, 1980.

[3] R. Kirchner and U. Kulisch, "Hardware support for interval arithmetic," *Reliable Computing*, vol. 12, no. 3, pp. 225–237, Jun 2006.

[4] S. Ocloo and W. Edmonson, "IIR Filter Adaptation using Branch-and-Bound: A Novel Approach," *IEEE Trans. on Circuits & Systems*, vol. 55, Issue 11, pp. 3393–3403, 2008.

[5] L. Jaulin, et. al. *Applied Interval Analysis*. Springer, 2001.

[6] K. Suffern and E. Fackerell, "Interval methods in computer graphics," *Computers & Graphics*, vol. 15, no. 3, pp. 331–340, 1991.

[7] R. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic, 1996.

[8] E. Hansen and W. Walster, *Global Optimization using Interval Analysis, 2nd Ed.*, M. Dekker, 2004.

[9] M. J. Schulte and J. E. E. Swartzlander, "A family of variable precision interval arithmetic processors," *IEEE Transactions on Computers*, vol. 49, pp. 387–398, May 2000.

[10] J. E. Stine and M. J. Schulte, "A combined interval and floating-point multiplier," *8th Great Lakes Symposium on VLSI*, pp. 208–213, Feb 1998.

[11] R. Gupte, W. Edmonson, S. Ocloo, and W. Alexander, "Pipelined ALU for signal processing to implement interval arithmetic," *The IEEE 2006 Workshop on Signal Processing Systems (SiPS'06)*, Banff, AB, Canada, pp. 95–100, 2006.

[12] S. Hattangady, W. Edmonson, and W. Alexander, "Block Floating Point Interval ALU for Digital Signal Processing", *13th GAMM-IMACS Inter. Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations*, El Paso, TX, Sep 2008.

[13] R.B. Kearfott, J. Pryce, and N. Revol, "Discussions on an Interval Arithmetic Standard at Dagstuhl Seminar 08021", *Lecture Notes in Computer Science*, vol. 5492, 2009.