# DISCUSSIONS ON AN INTERVAL ARITHMETIC STANDARD AT DAGSTUHL SEMINAR 08021

R. BAKER KEARFOTT, JOHN PRYCE, AND NATHALIE REVOL

## 1. Background

Efforts have been made to standardize interval arithmetic (IA) for over a decade. The reasons have been to enable more widespread use of the technology, to enable more widespread sharing and collaboration among researchers and developers of the technology, and to enable easier checking that computer codes have been correctly programmed. During the late 1990's, the first author of this report led such a project to introduce an interval data type into the Fortran language. One reason for failure of that effort was the Fortran language standardization committee's lack of familiarity with interval technology and consequent caution. Another was misunderstanding between the Fortran standardization committee's basic tenets on standardizing interline optimization and some views expressed by members of the interval analysis community. A third was confusion over how extended IA (arithmetic dealing with division by intervals that contain zero) should be handled. This was coupled with a heavy committee load associated with other projects, such as standardizing an interface for interoperability with "C" language programs.

Since then, the interval analysis community has studied and gained additional understanding of extended IA. One such study is [9], a systematization of the options. Another, with a particular point of view, is Prof. Kulisch's contribution to this volume. Extended arithmetic remains a controversial part of efforts to standardize the arithmetic, particularly whether the underlying model should consider $-\infty$ and $\infty$ to be numbers in their own right or if $-\infty$ and $\infty$ should just be considered placeholders to describe unbounded sets of finite real numbers. A practical consequence is a difference in the value of $0 \times X$ when $X$ is an unbounded interval. Nonetheless, our understanding and thinking about this issue is clearer than a decade ago. This, coupled with the desire to have a standard, should lead to progress.

A proposal is concurrently being developed to add IA to the C++ standard library. This work is presently slated to become a technical report (something that is generally implemented by compiler developers and is expected to become an integral part of a future standard).

Meanwhile, perhaps the most widely used IA system is that underlying INTLAB [10], although various other systems, such as PROFIL/BIAS [3, 4], FILIB++ [7, 6] and BOOST [8], and the interval facilities in the Sun compilers, e.g. [11], as well as systems within the XSC languages, see [2] and `http://www.xsc.de`, also are in wide use. The above references are biased towards C/C++, but a number of Fortran IA systems are also well used. See also the extensive bibliography in [5].

## 2. Proceedings at the Seminar

The context of this meeting was that

- The IFIP Working Group 2.5 on Numerical Software (WG 2.5) had written in September 2007 to the IEEE 754 Revision committee (IEEE754R) strongly supporting the inclusion of IA in the forthcoming standard (P754).
- IEEE754R had asked WG 2.5 to produce a proposal for inclusion in the standard, which they had submitted in October 2007 with a minor revision in December 2007. This proposal is based on the interval model described in the recent book by Kulisch, [5], see also the paper [1] by Kirchner and Kulisch.
- From then up to and including at the Dagstuhl meeting, there was a division of views among leading workers in intervals, as well as members of IEEE754R. Several expressed broad support for the WG 2.5 proposal but felt it lacked some crucial detail and could not be included in the standard without further work.

*Date*: November 21, 2008.

Others, however, still wished to address perceived defects before the IEEE754R deadline, supporting the Kulisch initiative. They felt that a separate standard for interval arithmetic would lack the impact of one that, being integral with P754, is recognised by the whole floating-point community. Members of this group, therefore, didn't sign the letter.

A first discussion on the topic was held at Dagstuhl on Thurs 10 Jan. The following aims to summarise this discussion. Speakers recorded are:

| | |
|---|---|
| JWvG : Jürgen Wolff von Gudenberg | PM : Peter Markstein |
| RBK : R. Baker Kearfott | JDP : John Pryce |
| UK : Ulrich Kulisch | NR : Nathalie Revol |
| RL : Rudolf Lohner | SR : Siegfried Rump |
| JMM : Jean-Michel Muller | NS : Neil Stewart |

RBK, leading the discussion, began by saying he believed there was a general view that the lack of consensus, and time constraints, made it impossible to include the WG 2.5 proposal in the current P754 revision. Instead, we should ask IEEE's authority for starting a new committee with wide participation from the interval community, to produce an IA standard in a longer timescale.

PM said that IA *had* been in the draft standard but too little of the detail was agreed by everyone so it was dropped. If we could agree on the WG 2.5 proposal or something close, we were in good shape to have it included in IEEE754 in a short time—provided we could muster enough people willing to work on the detail.

JMM said it was important to include people from IEEE754R, to ensure compatibility with that standard.

RBK proposed the motion:

(1) "That the WG 2.5 proposal be supplied to IEEE754R as the basis for an initial draft of a free-standing IA standard, initially independent of P754, to be worked on by a new committee set up for the purpose."

If this were carried, we should write a suitable letter to IEEE with a preliminary list of people who had expressed interest in participating.

UK: He had already essentially done this when submitting his proposal on 10th October.

JDP: Then the purpose of this meeting was to obtain wider consensus.

NS: The new committee needed members with a wide range of expertise—hardware, software engineering, people from industry, numerical analysts, members of research teams, ...

Possible members were suggested, including people from

- Intel, say Herbert Cornelius, since the MK library includes IA.
- Current IEEE754R committee, say David Hough.
- Current WG 2.5 committee, say Van Snyder (who is also on Fortran standard committee).
- ISL (Interval Subroutine Library), say Baker Kearfott and John Pryce.
- INRIA, say Nathalie Revol and Guillaume Melquiond.

RL and JWvG were willing to serve on the committee. UK might do so.

SR raised a more general question about the work. There are several underlying mathematical models on which IA can be based. The WG 2.5 proposal only supported one of these. There were several that he regarded as an acceptable basis for practical interval computation. Should the committee study several models with the aim of (a) choosing the one it thought best and/or (b) creating a standard that supports more than one model? These were not mutually exclusive options.

RBK proposed wording:

(2) "There are several well developed IA philosophies. The committee will study and *integrate* these, so that the resulting standard does not make it too difficult to implement any of them."

JMM: Probably the best IA philosophy to choose is

(3) "The one that is easiest to build other models on top of."

It was agreed that this should be put in the proposed letter.

NR mentioned the Hickey, Ju and van Emden interval model. The authors should be invited.

SR would consider who from the Hamburg group might take part.

RBK mentioned that members of the C++ interval standard group (Brönnimann, Melquiond, Pion) should all be invited.

The motion, as such, was dropped and it was agreed that a letter should be written proposing the new working group, with a remit based on (2, 3), and with a preliminary list of participants.

It was also agreed to start a discussion forum on the Reliable Computing mailing list, to find more participants, gather opinions, and clarify the points of consensus and most crucially the main points of disagreement.

The meeting closed. A letter was drafted by RBK, JDP and NR. Further discussions took place the following day and the letter was revised as a result. It was sent, dated 11th Jan 2008, to Dr Dan Zuras, Chair IEEE 754 Revision Committee, with a copy to Dr. R. F. Boisvert, Chair, IFIP Working Group 2.5. Signatories supporting the letter (listed in order of signing) were:

> R. Baker Kearfott, W. Luther, J.D. Pryce, Nathalie Revol, S.M. Rump, Guillaume Melquiond, Michel Kieffer, Vincent Lefèvre, C. Keil, Andreas Rauh, Jean-Michel Muller, Nicolas Louvet, Jean-Luc Lamotte, Markus Neher, Rudolf Lohner, Peter Markstein.

## 3. Subsequent Developments

Subsequently, the first and third authors of this note collected a list of electronic mail and postal addresses for interested persons. Fortunately, these persons included those who were unable to attend the final meeting, as well as those against formation of the new committee and in favor of trying to get IEEE to adopt the results of IFIP Working Group 2.5.

The three authors of this note worked with Bob Davis, chair of the IEEE Microprocessor Standards Committee, to frame the wording of, and submit, a Project Authorization Request through the IEEE New Standards Committee (NSC), for work on a stand-alone standard (separate from the floating point standard P754) for interval arithmetic. The Scope, Purpose, and Rationale of the project are as follows.

> 1. *Scope.* This standard will specify basic interval arithmetic operations selecting and following one of the commonly used mathematical interval models and at least one floating-point type defined by the IEEE-754/2008 standard. Exception conditions will be defined and standard handling of these conditions will be specified. Consistency with the model will be tempered with practical considerations based on input from representatives of vendors and owners of existing systems.
>
> The standard will provide a layer between the hardware and the programming language levels. It will not mandate that any operations be implemented in hardware. It will not define any realization of the basic operations as functions in a programming language.
>
> 2. *Purpose.* The standard's 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. There are presently many systems for interval arithmetic in use, and lack of a standard inhibits development, portability, and ability to verify correctness of codes.
>
> 3. *Rationale.* There is presently no defined standard, although there are many systems in use today. However, due to the nature of applications of interval arithmetic in providing automatic verification, simplicity, predictability, and portability are doubly important in the underlying computations. The standard will provide the necessary ease of implementation, portability, and ability to check correctness of codes.

Authorization for the project passed unanimously at the NSC on June 11, 2008 and it was given the title *IEEE Working Group P1788, Standardization of Interval Arithmetic.* As the next step, an organizational meeting was held in conjunction with SCAN-2008 in El Paso, Texas, USA, September 29 to October 3, 2008.

The agenda for this meeting included election of officers and agreement on procedures for decision-making. Such procedures need to include all interested parties, and should be such that all will respect the decisions made according to the procedures. Through these procedures, we strive to gain wider and more unambiguous acceptance, at all levels, than of the document produced by IFIP Working Group 2.5.

Although the project is for a stand-alone document, a goal is to make it consistent with the floating point arithmetic standard P754, for eventual incorporation therein. Some (but not all) issues related to this are

- the underlying mathematical model for the real numbers,
- requirements that features be implemented in hardware (or not),
- the set of features that are implemented.

The time frame for the project is four years, although the aim and expectation is to complete the process sooner.

The P1788 membership (currently around 90) has approved the slate of officers as follows. Nathalie Revol, Chair (currently on maternity leave). R. Baker Kearfott and John Pryce, Vice chairs. William Edmonson, Secretary. Jürgen Wolff von Gudenberg, Web master. Guillaume Melquiond, Archivist. David Lester and John Pryce, Technical editors. George Corliss, Vote tabulator.

At the time of writing, the membership is voting on the Policies and Procedures document, which is based on standard IEEE practice. An active discussion of many aspects of the standard is underway on the electronic forum at `stds-1788@LISTSERV.IEEE.ORG`.

## References


[1] Reinhard Kirchner and Ulrich W. Kulisch. Hardware support for interval arithmetic. *Reliable Computing*, pages 225–237, 2006.

[2] Rudi Klatte, Ulrich Kulisch, Christian Lawo, Michael Rauch, and Andreas Wiethoff. *C–XSC: A C++ Class Library for Extended Scientific Computation*. Springer-Verlag, Berlin, 1993.

[3] Olaf Knüppel. PROFIL/BIAS – A fast interval library. *Computing*, 53(3–4):277–287, September 1994.

[4] Olaf Knüppel. PROFIL/BIAS v 2.0. Bericht 99.1, Technische Universität Hamburg-Harburg, Harburg, Germany, February 1999. `http://www.ti3.tu-harburg.de/profil_e`.

[5] Ulrich Kulisch. *Computer Arithmetic and Validity — Theory, Implementation, and Applications*. Number 33 in De Gruyter Studies in Mathematics. Walter de Gruyter, Berlin, Germany, 2008.

[6] Michael Lerch, German Tischler, and Jürgen Wolff von Gudenberg. `filib`—interval library specification and reference manual. Technical Report 279, Universität Würzburg, Germany, 2001. `http://www.math.uni-wuppertal.de/~xsc/software/filib.html`.

[7] Michael Lerch, German Tischler, Jürgen Wolff von Gudenberg, Werner Hofschuster, and Walter Krämer. `filib++`, a fast interval library supporting containment computations. *ACM Transactions on Mathematical Software*, 32(2):299–324, June 2006.

[8] Guillaume Melquiond, Sylvain Pion, and Hervé Brönnimann. Boost interval arithmetic library, 2006. `http://www.boost.org/doc/libs/1_37_0/libs/numeric/interval/doc/interval.htm`.

[9] John Derwent Pryce and George F. Corliss. Interval arithmetic with containment sets. *Computing*, 78(3):251–276, 2006.

[10] Siegfried M. Rump. INTLAB–INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing: Papers presented at the International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics, SCAN-98, in Szeged, Hungary*, volume 5(3) of *Reliable Computing*, pages 77–104, Dordrecht, Netherlands, 1999. Kluwer Academic Publishers. `http://www.ti3.tu-harburg.de/rump/intlab/`.

[11] Sun Microsystems. Interval arithmetic in the Forte[tm] C++ compiler, 2000. `http://www.sun.com/forte/cplusplus/interval/`.