Summary of the seminar 08021

Numerical validation in current hardware architectures - From embedded system to high-end computational grids

Annie Cuyt (University of Antwerp), Walter Krämer (University of Wuppertal), Wolfram Luther (University of Duisburg-Essen) , Peter Markstein (Woodside California)

The major emphasis of the seminar concentrated on numerical validation in current hardware architectures and software environments. The general idea was to bring together experts who are concerned with computer arithmetic in systems with actual processor architectures and scientists who develop, use and need techniques from verified computation in their applications. Topics of the seminar therefore included

- The ongoing revision of the IEEE 754/854 standard for floating-point arithmetic
- Feasible ways to implement multiple precision (multiword) arithmetic and to compute the actual precision at run-time according to the needs of input data
- The achievement of a similar behaviour of fixed-point, floating-point and interval arithmetic across language compliant implementations
- The design of robust and efficient numerical programs portable from diverse computers to those that adhere to the IEEE standard.
- The development and propagation of validated special purpose software in different application areas
- Error analysis in several contexts
- Certification of numerical programs, verification and validation assessment.

Computer arithmetic plays an important role at the hardware and software level, when microprocessors, embedded systems or grids are designed. The reliability of numerical software strongly depends on the compliance with the corresponding floating-point norms. Standard CISC processors follow the 1985 IEEE-754 Standard which is actually under revision, but the new highly performing CELL processor is not fully IEEE compliant. The draft standard IEEE 754r guarantees that "systems perform floating-point computation that yields results independent of whether the processing is done in hardware, software, or a combination of the two. For operations specified in this standard, numerical results and exceptions are uniquely determined by the values of the input data, sequence of operations, and destination formats, all under user control." There was a broad consensus that the standard should include interval arithmetic.

The discussion focused on decimal number formats, faithful rounding, longer mantissa lengths, higher precision standard functions and linking of numerical and symbolic algebraic computation. This work is accompanied by new vector, matrix and elementary or special function libraries (i.e. complex functions and continued fractions) with guaranteed precision within their domains. Functions should be correctly rounded and even last bit accuracy is available for standard functions. Important discussion points were additional features like fast interval arithmetic, staggered correction arithmetic or a fast and accurate multiply and accumulate instruction by pipelining. The latter is the key operation for a fast multiple precision arithmetic. An exact dot product (implemented in pipelined hardware) for floating point vectors would provide operations in vector spaces with accurate results without any time penalty. Correctly rounded results in these vector spaces go hand in hand with correctly rounded elementary functions. The new norm will be based on solid and interesting theoretical studies in integrated or reconfigurable circuit design, mathematics and computer science.

In parallel to the ongoing IEEE committee discussions, the seminar aimed at highlighting design decisions on floating-point computations at runtime over the whole execution process under the silent consensus that there are features defined by the hardware standard, language defined or deferred to the implementation for several reasons.

Hardware and software should support several (user defined) number types, i.e. fixed width, binary or decimal floating-point numbers or interval arithmetic. A serious effort is actually made to standardize the use of intervals especially in the programming language C. Developers are encouraged to write efficient numerical programs that are easily portable with small revisions to other platforms. C-XSC is a C++ Library for Extended Scientific Computing to develop numerical software with result verification. This library is permanently enlarged and enhanced as highlighted in several talks.

However, depending on the requirements on speed, input and output range or precision, special purpose-processors also use other number systems, i.e. fixed-point or logarithmic number systems and non-standard mantissa length. Interesting reported examples were a 16-bit interval arithmetic on the FPGA (Field Programmable Gate Array) based NIOS-II soft processor and an online-arithmetic with rational numbers. Therefore, research on reliable computing includes a wide range of current hardware and software platforms and compilers.

Standardization is also asked for inhomogeneous computer networks. The verification step should validate the partial results coming from the owners of parcels before combining them to the final result.

Our insights and the implemented systems should be used by people with various numerical problems to solve these problems in a comprehensible and reliable way and by people incorporating validated software tools into other systems or providing interfaces to these tools.

So we want to create an increasing awareness of interval tools, the validated modeling and simulation systems and computer-based proofs in science and engineering.

List of participants

Numerical Validation in Current Hardware Architectures

1. Götz Alefeld goetz.alefeld@math.uni-karlsruhe.de Universität Karlsruhe
2. Ekaterina Auer auer@informatik.uni-duisburg.de Universität Duisburg-Essen
3. Gerd Bohlender, bohlender@math.uni-karlsruhe.de University of Karlsruhe
4. Annie Cuyt Annie.Cuyt@ua.ac.be University of Antwerp
5. Joris van Deun, Joris.VanDeun@ua.ac.be University of Antwerp
6. Tom Docx, tom.docx@ua.ac.be University of Antwerp
7. Eva Dyllong dyllong@informatik.uni-duisburg.de Universität Duisburg-Essen
8. Andreas Frommer Andreas.Frommer@math.uni-wuppertal.de BU Wuppertal
9. Markus Grimmer grimmer@math.uni-wuppertal.de BU Wuppertal
10. Werner Hofschuster werner.hofschuster@math.uni-wuppertal.de BU Wuppertal
11. Carlos Hölbig holbig@upf.br Universtity de Passo Fundo
12. Di Jiang jiangdi@iro.umontreal.ca University of Montreal
13. Christian Keil, c.keil@tu-harburg.de TU Hamburg-Harburg
14. Baker Kearfott rbk@louisiana.edu University of Louisiana, Lafayette
15. Michel Kieffer michel.kieffer@lss.supelec.fr U Paris Sud Supélec
16. Rudi Klatte, Rudi.Klatte@math.uni-karlsruhe.de University of Karlsruhe
17. Walter Krämer Walter.Kraemer@math.uni-wuppertal.de BU Wuppertal
18. Ulrich Kulisch, ulrich.kulisch@math.uni-karlsruhe.de University of Karlsruhe
19. Jean-Luc Lamotte Jean-Luc.Lamotte@lip6.fr Université Pierre et Marie Curie Paris
20. Bruno Lang lang@math.uni-wuppertal.de BU Wuppertal
21. Vincent Lefèvre vincent@vinc17.org INRIA Lorraine
22. Rudolf Lohner, Rudolf.Lohner@ kit.edu Karlsruher Institut für Technologie
23. Nicolas Louvet, nicolas.louvet@ens-lyon.fr ENS Lyon
24. Mariana Lüderitz Kolberg, mkolberg@inf.pucrs.br PUCRS -Porto Alegre
25. Wolfram Luther luther@informatik.uni-duisburg.de Universität Duisburg-Essen
26. Peter Markstein peter@markstein.org HP Laboratories
27. Gregorio de Miguel Casado, gmiguel@unizar.es University of Zaragoza
28. Guillaume Melquiond guillaume.melquiond@ens-lyon.fr ENS Lyon
29. Jean-Michel Muller jean-michel.muller@ens-lyon.fr ENS Lyon
30. Markus Neher markus.neher@math.uni-karlsruhe.de Universität Karlsruhe
31. Shinichi Oishi oishi@waseda.jp Waseda University
32. Evgenija D. Popova, epopova@bio.bas.bg Bulgarian Academy of Sciences
33. John D. Pryce, j.d.pryce@ntlworld.com Cranfield University
34. Andreas Rauh, Andreas.Rauh@uni-ulm.de Universität Ulm
35. Nathalie Revol nathalie.revol@ens-lyon.fr ENS Lyon
36. Siegfried M. Rump rump@tu-harburg.de TU Hamburg-Harburg
37. Neil Stewart stewart@iro.umontreal.ca University of Montreal
38. Jürgen Wolff von Gudenberg wolff@informatik.uni-wuerzburg.de U. Würzburg
39. Michael Zimmer, zimmer@math.uni-wuppertal.de Universität Wuppertal
40. Paul Zimmermann Paul.Zimmermann@loria.fr INRIA Lorraine

Seminar 08021 January 6 – 11 2008

Numerical validation in current hardware architectures:
From embedded system to high-end computational grids

List of talks

Monday, January 7

9:30 Opening session A. Cuyt, W. Krämer, W. Luther, P. Markstein

The ongoing revision of the IEEE 754/854 standard for floating-point arithmetic
10:15 P. Markstein: The New IEEE-754 Standard for Floating Point Arithmetic
11:00 U. Kulisch: A proposed standard for interval arithmetic and complete arithmetic

Discussion

Enhanced accuracy, correct rounding, multiple precision
14:00 J. M. Muller: Some algorithmic improvements due to the availability of an FMA
14:45 W. Krämer: A modified staggered correction arithmetic with enhanced accuracy and very wide exponent range

16:00 N. Revol: Automatic adaptation of the computing precision
16:45 P. Zimmermann: Implementation of the inverse square root in MPFR

17.30 Discussion on the ongoing IEEE 754 standard revision

Tuesday, January 8

The design of robust and efficient numerical programs portable from diverse computers to those that adhere to the IEEE standard.
9:00 E. D. Popova: On the Interoperability between Interval Software
9:30 M. Neher: Complex Inclusion Functions in the CoStLy C++ Library
10.20 B. Lang: Verified Computations of 3D Spherical t-Designs – Challenges, Approaches, and Current Limitations
11:00 N. Stewart: Robustness of Boolean operations on subdivision surface models 1
11:40 E. Dyllong: Some Applications of Interval Arithmetic in Hierarchical Solid Modelling

Language support – Reliable libraries – High-precision special functions
14:00 W. Hofschuster: C-XSC: Highlights and new developments
14:35 M. Grimmer: Some Tools and Applications Extending the Range of C-XSC
15:10 M. Zimmer: Fast (parallel) dense linear interval systems solving in C-XSC using error free transformations and BLAS

Reliable floating-point and interval computing in different areas – Verification and validation assessment
16:15 W. Luther: Verification and Validation Assessment in Computational Biomechanics
16:45 E. Auer: SmartMOBILE and Its Applications to Biomechanics
17:20 A. Rauh: Towards the Development of an Interval Arithmetic Environment for Validated Computer-Aided Design and Verification of Systems in Control Engineering
20:00 Software demonstration

Wednesday, January 9

Numerical verification in current hardware architectures and software environments
9:00 M. Kieffer: Efficient 16-bit Floating-Point Interval Processor for Embedded Systems
10:15 Jean-Luc Lamotte: Extended precision on the CELL processor
10:55 Di Jiang: Robustness of Boolean operations on subdivision surface models 2
11.35 G. de Miguel Casado:A Software Library for Reliable Online-Arithmetic with Rational Numbers

13:30 Excursion to Saarbrücken – Guided tour - sight seeing – Picasso exhibition – Dinner (Brewery Zum Stiefel)

Thursday, January 10

Reliable methods – Accurate operations – Language support – Reliable function libraries
9:00 G. Alefeld and Z. Wang: Verification of Solutions of Complementarity Problems with Tridiagonal (nonlinear) Functions
9:40 J. Wolff von Gudenberg: Interval arithmetic and standardization
Discussion on interval arithmetic and standardization
11.00 S. Rump: Accurate summation
11:40 A. Frommer: Error bounds for matrix functions

14:00 R. Lohner: Supercomputing at KIT
14:35 Shin'ichi Oishi: Iterative Refinement for Ill-Conditioned Linear Systems
15:10 G. Melquiond: Certifying numerical programs
16:15 A. Cuyt: Continued fractions for special functions – Handbook and Software with demonstration
17:15 R. B. Kearfott: Issues for General Users of Validated Optimization Software: Recent Experiences with GlobSol
Demonstration in a further (evening) session
20:00 Standardization – Forthcoming discussion

Friday, January 11

Reliable floating-point and interval computing in different environments
9:00 N. Louvet: Compensated Horner scheme in k times the working precision
9:30 V. Lefèvre: Searching for the worst cases for the correct rounding of the power functions in double precision
10:30 J. Pryce: DAETS differential algebraic equation solver

11.15 Closing session – Outcome of the seminar – proceedings – forthcoming conferences – Reliable Computing – common activities

The audience decided to submit papers to the DROPS platform before March 16. Abstracts and extended abstracts are obligatory, papers are welcome.

Seminar proceedings are planned within Springer's conference series Lecture Notes in Computer Science. The deadline is March 30. Main topics are

- Standardization (IA and FPA)
- Accurate operations
- Programming languages and libraries on different platforms
- Multiple precision arithmetic
- Accurate standard and special functions
- Algorithms with result verification
- Robustness in applications
- Application tools with result verification
- Certification, validation and verification of systems in different areas

The organizers would like to encourage all participants to provide the program committee with excellent contributions and to come together in groups of two or three authors to present a coherent description of related research and application.

The participants wrote a letter to the chairman of the IEEE 754 Revision Committee asking for including interval arithmetic into the IEEE 754r standard. A similar initiative was recently undertaken by the GAMM Activity Group on Computer Arithmetic and Scientific Computing (see http://www.math.uni-wuppertal.de/org/WRST/gamm-fa/) and the IFIP Working Group 2.5 - Numerical Software. Both groups asked for completing the IEEE standard for floating-point arithmetic by specifying optional support for interval arithmetic and complete arithmetic (i.e. exact dot product).

All participants want the inclusion of an IA standard into the IEEE 754 in the near future but disagree on how to obtain this goal. If the ongoing IEEE revision will not include IA, a group of experts was nominated to serve on a new committee to standardize IA. There are important reasons to standardize IA. Automatically verified methods constitute mathematical proofs and allow for verifying numerical computations. A standard is needed for validating algorithms on different platforms and for promoting portability and interoperability between interval-based software.

We would also like to express our thanks to the scientific board of the Dagstuhl castle for giving us the opportunity to organize this seminar and the Dagstuhl administration for its excellent job.