

Reliable Implementation of Real Number Algorithms: Theory and Practice

Summary of Dagstuhl Seminar 06021
January 8 – 13, 2006

Peter Hertling¹, Christoph M. Hoffmann², Wolfram Luther³, and Nathalie Revol⁴

¹ Universität der Bundeswehr München, Neubiberg, Germany

² Purdue University, West Lafayette, Indiana, USA

³ Universität Duisburg-Essen, Duisburg, Germany

⁴ École Normale Supérieure de Lyon, Lyon, France

Abstract. The seminar brought together researchers from many different disciplines concerned with the reliable implementation of real number algorithms either from a theoretical or from a practical point of view. In this summary we describe the topics, the goals, and the contributions of the seminar.

Key words: Real number computability, real number algorithms, reliable computing, algorithms with result verification, interval arithmetic, geometric computing, robustness, solid modeling.

1 Motivation

Real numbers are objects containing an infinite amount of information. Therefore, they cannot be represented precisely in a computer. This leads to well known problems caused by unverified finite precision implementations of real number algorithms.

There are several scientific communities, not only in mathematics but also in computer science, that are concerned with reliable real number algorithms. Computable analysis is a fast growing subdiscipline of theoretical computer science that analyses real number computation problems in the context of the Turing machine model. Another theoretical approach is domain theory. Here one of the goals is to lay foundations for a programming language for exact real arithmetic. There are many approaches that deal with the reliable implementation of real number algorithms from a practical point of view. The basic idea of interval arithmetic, to start with, is to compute with intervals that are known to contain the real number in question. It is striking that the space of intervals is a special case of a domain. Other, in many cases related approaches are Taylor models, high precision software, exact arithmetic, algorithms using result verification, symbolic representations of part of the data, algebraic computation schemes,

perturbation schemes, and many more. These approaches are being applied in order to solve numerical as well as geometric problems. In computational geometry the special problem of implementing real number algorithms reliably is complicated by the interplay of numerical predicates and hidden dependencies between them that arise from geometry theorems that may not be known. This creates opportunities for inconsistent decisions that lead to faulty data structures and, ultimately, to failure of the computation.

It was the goal of the seminar to bring together people who are dealing with the reliable implementation of real number algorithms either from a theoretical or from a practical point of view and to stimulate an exchange of ideas between the different communities that will bring an advance for the reliable solution of numerical and geometric problems. Some particular goals of the seminar were:

- to point out the most urgent current practical problems in the implementation of real number algorithms, to analyze them using the tools and notions from topology, computability theory and complexity theory, and hopefully to understand them better and make progress towards a solution,
- to explore the practical aspects of the computability and complexity notions for various types of continuous objects, based on various topologies and the resulting types of information and of representation methods that are used in order to describe the objects in an approximating way,
- to analyze and compare the various software tools for reliable implementation of real number algorithms, to analyze and compare their advantages and limits, and to explore the need and the possibilities to develop further software tools specially suited for developing and implementing reliable algorithms over the real numbers,
- to integrate reliable functions and algorithms into computer algebra systems as well as recent modeling and simulation software.

2 The Seminar

Forty eight researchers from many different disciplines attended the seminar: people concerned with constructive mathematics or logic, with computability theory or complexity theory over the real numbers, with interval arithmetic, with robustness problems in computational geometry or solid modeling, with computer arithmetic, and with software for fast, arbitrarily high precision computations. The program consisted of 35 talks of 30 minutes each, and of three evening sessions with additional presentations and discussions. Many presentations showed that there are already strong interconnections between various disciplines. There were also lively discussions about different theoretical models and practical approaches for reliable real number computations. In the following, we summarize the topics of the seminar under a few headings. Many talks listed under one of the following headings could as well be listed under one or more of the other headings.

Constructive Mathematics and Computability Theory over the Real Numbers

One approach in Computable Analysis is to represent objects by bit streams and to perform computations by Turing machines on such bit streams. This leads to an approximative notion of computation which is quite closely related to constructive mathematics and corresponds to reliable computation on a digital computer. In an additional evening session, Klaus Weihrauch gave an introduction to this theory. Vasco Brattka showed that via this approach important results of functional analysis can be effectivized, thus making an analysis possible whether certain partial differential equations, e.g., elliptic boundary value problems, can be solved on a Turing machine resp. a digital computer or not. Andrej Bauer demonstrated how via realizability a constructive mathematical first-order theory, e.g., the constructive theory of real numbers, can be translated into a specification required for an implementation, e.g., of exact real number arithmetic. Paul Taylor explained Abstract Stone Duality, a new kind of general topology, based on logical principles, and demonstrated that it can be used to develop a syntactic calculus of the real line that can also be translated into interval analysis. Chee K. Yap presented an alternative model for real number computation based on the principle of Exact Geometric Computation that is realized in several software systems for Geometric Computation. His talk was continued in an evening session and followed by a lively discussion about models for real number computation.

Complexity Theory over the Real Numbers and Software for Fast, Arbitrarily High Precision Computation

Kurt Mehlhorn presented a fast and practically usable algorithm for the isolation of the roots of a polynomial if the coefficients of the polynomial are given as bit streams as above. Norbert Müller analysed the well-know quadrature schemes Newton-Cotes and Gauss-Legendre for numerical integration with respect to the number of bit operations needed in order to compute the integral with precision 2^{-n} . Paul Zimmermann presented a fast algorithm for computation of the sine and cosine functions based on Brent's algorithm for fast computation of the exponential function. Both, Norbert Müller and Paul Zimmermann, also showed experiments concerning their results in their respective software systems for fast computation with arbitrarily high precision. Joris van der Hoeven presented another such system in the form of a C++ library for computation with effective real numbers. Branimir Lambov, who has developed a similar, very fast system, explained that part of it is based on Interval Arithmetic and how he has implemented that, making efficient use of hardware components in order to speed up computations. In a different setting, Margarita Korovina showed new lower and upper bounds for finite bisimulations of hybrid dynamical systems defined by Pfaffian maps, thus clarifying how costly it is to simulate such dynamical systems in practice.

Computational Geometry and Solid Modeling, Robustness Problems

Thomas J. Peters presented new ideas based on topological considerations for dealing with the difficulties that are caused by floating point computations specifically in animations and scientific visualizations. Vadim Shapiro discussed methods for valid modeling of solids in the presence of errors in the geometric data and/or algorithms. He argued that usually it is assumed that the intended exact set is homotopy equivalent to a set corresponding to the given approximate geometric data, and showed how one can obtain a guarantee that such a homotopy equivalence exists. Neil Stewart showed how the Whitney Extension Theorem can be used to associate a well-defined solid with a possibly inconsistent set of data that are supposed to describe a solid. He also gave a bound on the deviation of the normal vectors of the newly defined solid from the corresponding normal vectors of patches contained in the description. Eva Dyllong also dealt with solid modeling. She discussed several approaches for constructing polyhedral enclosures of interval-based hierarchical object representations. Bernard Mourrain discussed semi-algebraic models of geometric shapes, explaining how both computations with algebraic numbers and subdivision methods come into play, and paying special attention to the computation of the topology of curves and surfaces. Dominique Michelucci reviewed a number of approaches for dealing with the robustness problem in computational geometry. Specifically, he compared approaches based on the idea of Exact Geometric Computation (see above: Yap), on interval analysis and probabilistic methods. Stefan Schirra illustrated visually the shortcomings of (naive) floating point arithmetic applied to geometric problems.

Interval Arithmetic and Software Systems for Reliable Computations

A basic task in interval arithmetic is the evaluation of a real function over an interval, i.e., the computation of a small interval that contains the range of the function over the given interval. Markus Neher showed how this can be done efficiently for certain complex standard functions and presented a number of implemented examples. Götz Alefeld reported on a method which allows to improve an approximate eigenpair iteratively. The dependence of intermediate results in a computation has often been considered in interval analysis. Vladik Kreinovich explained what one can do if there is already some dependence among the input values of a problem. Ekaterina Auer discussed various approaches to validated modeling of multibody systems and presented a template-based tool for verifying kinematical as well as dynamical properties of different types of multibody systems. Wolfram Luther discussed verified methods in stochastic traffic modeling using, among other things, semi-Markov processes. David Lester presented a formally verified interval arithmetic package with a theorem prover augmented to automatically deduce the validity of (some) real number inequalities. Jean-Marie Chesneaux presented a toolbox, working under Simulink, for evaluating the range of implemented functions over given input data. George F. Corliss, John D. Pryce, Baker Kearfott, and Spencer Smith explained different aspects

of their project to gather and disseminate a portable, easy to use, comprehensive Interval Subroutine Library for reliable computations. They discussed the planned overall structure of the library, the planned mathematical basis of the library (containment sets), how they plan to collect, organize and integrate already existing work and to ask for new contributions, and the software engineering methodologies they want to use in order to ensure and improve the quality of the library. Sylvain Pion reported on his and some colleagues' proposal to include interval arithmetic in the next C++ ISO standard.

Floating Point Arithmetic, Verification of Software

In an additional evening session, Paul Zimmermann reported on the last meeting concerning the IEEE 754 revision. Siegfried M. Rump presented a variety of algorithms for computing sum and dot product of floating-point vectors. Philippe Langlois showed how the Fused Multiply and Add operator that provides $x*y+z$ with only one final rounding error can be used to improve the accuracy and the speed efficiency of the Horner scheme for polynomial evaluation. Sylvie Boldo presented new results concerning the correctness of implementations of elementary functions in floating-point arithmetic. Sylvie Putot presented a static analysis of the errors introduced by the approximation by floating-point arithmetic of real number computations. Her analysis is based on, among other things, affine interval arithmetic. Guillaume Melquiond presented a tool for automatically certifying numerical programs, which relies on interval arithmetic and forward error analysis to bound mathematical expressions involving floating-point rounded operators. Jürgen Wolff von Gudenberg showed how evaluation semantics of numerical expressions can be controlled in C++ using expression templates.