Lecture Notes in Computer Science 5045

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Peter Hertling   Christoph M. Hoffmann
Wolfram Luther   Nathalie Revol (Eds.)

Reliable Implementation of Real Number Algorithms: Theory and Practice

International Seminar
Dagstuhl Castle, Germany, January 8-13, 2006
Revised Papers

Springer

Volume Editors

Peter Hertling
Universität der Bundeswehr München
Fakultät für Informatik
85577 Neubiberg, Germany
E-mail: peter.hertling@unibw.de

Christoph M. Hoffmann
Purdue University
Department of Computer Science
West Lafayette, IN 47907-1398, USA
E-mail: cmh@cs.purdue.edu

Wolfram Luther
Universität Duisburg–Essen
Fakultät für Ingenieurwissenschaften
Abteilung Informatik und Angewandte Kognitionswissenschaft
47048 Duisburg, Germany
E-mail: luther@inf.uni-due.de

Nathalie Revol
École Normale Supérieure de Lyon
Laboratoire de l'Informatique du Parallélisme
Projet Arenaire
46, allée d'Italie, 69364 Lyon CEDEX 07, France
E-mail: Nathalie.Revol@ens-lyon.fr

Library of Congress Control Number: 2008932924

CR Subject Classification (1998): G.1, B.2, B.8, F.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-85520-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-85520-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12513802 06/3180 5 4 3 2 1 0

Preface

A large amount of the capacity of today's computers is used for computations that can be described as computations involving real numbers. In this book, the focus is on a problem arising particularly in real number computations: the problem of verified or reliable computations. Since real numbers are objects containing an infinite amount of information, they cannot be represented precisely on a computer. This leads to the well-known problems caused by unverified implementations of real number algorithms using finite precision. While this is traditionally seen to be a problem in numerical mathematics, there are also several scientific communities in computer science that are dealing with this problem.

This book is a follow-up of the Dagstuhl Seminar 06021 on "Reliable Implementation of Real Number Algorithms: Theory and Practice," which took place January 8–13, 2006. It was intended to stimulate an exchange of ideas between the different communities that deal with the problem of reliable implementation of real number algorithms either from a theoretical or from a practical point of view. Forty-eight researchers from many different countries and many different disciplines gathered in the castle of Dagstuhl to exchange views and ideas, in a relaxed atmosphere. The program consisted of 35 talks of 30 minutes each, and of three evening sessions with additional presentations and discussions. There were also lively discussions about different theoretical models and practical approaches for reliable real number computations. For a short description of the full program of the seminar including a complete list of presentations at the seminar the reader is referred to the Web page of the seminar on the website `http://www.dagstuhl.de/` of Schloss Dagstuhl. During the seminar, theories and results concerning computability over the real numbers or just over the real algebraic numbers were presented and discussed. Topics also included formal proofs to gain an extra level of confidence in the quality of computed results, be they computed with effective real numbers or with floating-point arithmetic. Other discussions concerned software libraries, systems, and platforms: exploiting the well-defined specifications of floating-point arithmetic to get accurate and provable results, using interval arithmetic to obtain reliable results, or assessing the quality of floating-point results. Another theme was computational geometry and solid modeling, mainly as far as robustness is concerned, along with proposed solutions. Note that 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.

We will now introduce the topics and results presented in the 12 articles in this book. They are written by participants of the seminar, and most of them are

based on presentations at the Dagstuhl Seminar. They represent a cross-section through the topics of the seminar. First, we describe two papers presenting computability notions specially suited for geometric computations and results concerning effective computations involving real algebraic numbers. These two papers lead to computational geometry and solid modelling, which is treated in five papers, one of them containing an overview and a critical discussion, and one of them considering scientific visualization. Then we discuss three papers dealing with software systems for reliable computations and with implementation issues. They involve floating-point arithmetic or interval arithmetic or both. Finally, two papers are concerned with applications. They present methods and algorithms for solving real-world problems.

Computability and Complexity over Real Numbers and Real Algebraic Numbers

One approach in computable analysis is to represent objects by bit streams and to perform computations by Turing machines on such bit streams. Thus, real numbers are represented in an approximating way. This leads to a notion of computation which is quite closely related to constructive mathematics and corresponds to reliable computation on a digital computer. An entirely different, algebraic theory of computability and complexity over the real numbers is based on the real random access machine model. The fundamental assumption in this model is that elementary arithmetic operations as well as comparisons over the real numbers can be performed in one step. In his contribution, C. Yap presents a variant of the first approach in which it is possible to decide whether a given number is equal to zero or not. This approach is based on the so-called Exact Geometric Computation mode, which is encoded in several well-known libraries for geometric computation such as LEDA, CGAL, and the Core library. C. Yap's paper addresses the problem to provide a theoretical foundation for the Exact Geometric Computation mode of computation.

In the implementation of the Exact Geometric Computation mode and in the implementation of algorithms in computational geometry in general, effective operations on real algebraic numbers play an important role. I.Z. Emiris, B. Mourrain, and E.P. Tsigaridas analyze the bit complexity, that is, the complexity in the Turing machine model, of several operations involving real algebraic numbers. In particular, they consider real root isolation of univariate integer polynomials, sign evaluation and comparison of real algebraic numbers, and the problem of simultaneous inequalities. They give an overview of existing approaches and unify, simplify and improve them. In addition, they present results of numerical experimentations using these algorithms.

Computational Geometry and Solid Modeling, Robustness Problems

D. Michelucci, J.M. Moreau, and S. Foufou give a critical survey of a number of approaches for dealing with the robustness and inaccuracy problem in computational geometry. Specifically, they compare the approach based on the idea

of Exact Geometric Computation, as explained in the paper by C. Yap, the use of interval arithmetic, and either arithmetic or geometric probabilistic methods. Their conclusion is that geometric probabilistic approaches, such as the sampling of a surface or of a configuration space (for the motion planning problem) or Monte-Carlo and ray-tracing methods prove to be simple, tractable with the increasing power of computers, and robust.

Computational geometry is closely related to solid modeling. Geometric objects are often described by geometric data and by combinatorial data. Due to imprecise geometric embeddings these data may become inconsistent. V. Shapiro uses “inflated” representations, e.g., a segment is “thickened.” He argues that usually it is assumed that the intended exact set is homotopy equivalent to a set corresponding to the given approximate geometric data. He shows how sufficient conditions for such a homotopy equivalence can be derived systematically from the Nerve Theorem.

When the problem deals with boundaries of patches of a trimmed surface, the inconsistency usually resides in the disjointness of the boundaries of the representations of theoretically adjacent patches. Then, one aims at changing the inconsistent boundaries to close and consistent boundaries. This is called transfinite interpolation. N.F. Stewart and M. Zidani show how in the case of sets defined by combined subdivision surfaces, one can use the Whitney extension theorem for transfinite interpolation. They also give a bound on the deviation of the normal vectors of the newly defined surface from the corresponding normal vectors of patches contained in the original description of the surface.

The last two papers dealing with geometric or solid modeling address reliability and accuracy issues. E. Dyllong uses a representation of objects by octrees, using intervals for reliability. She presents an algorithm for the determination of polyhedral convex enclosures of such objects, which is both fast and accurate. Accuracy is obtained via the use of the accurate dot-product for the tests of orientation and visibility.

The difficulties caused by imprecise computations due to finite precision floating-point arithmetic increase when one switches from geometric modeling to animations and scientific visualizations. L.E. Miller, E.L.F. Moore, T.J. Peters, and A. Russell consider the problem of computing the minimal distance between two curves. They propose an algorithm that is fast enough for interactive visualization purposes. Their algorithm consists in sampling couples of points on these two curves, in discarding quickly points that are not good candidates, and in performing Newton’s method on the remaining pairs. For reliability reasons, they also provide an algorithm that determines a lower bound on this minimal distance, along with a guarantee on the quality of this lower bound.

Software Systems for Reliable Computations, Implementation Issues

The finite precision of floating-point arithmetic has been described repeatedly as one of the reasons for the difficulties in numerical and geometric computation. At least one wishes to have an implementation of floating-point arithmetic with

correct rounding. In fact, in the IEEE 754-1985 standard for binary floating-point arithmetic it is required that all four arithmetic operations and the square root function are correctly rounded. In order to be able to round correctly $f(x)$ where f is an elementary function and x is a floating-point value, one needs to evaluate $f(x)$ with extra precision. V. Lefèvre, D. Stehlé, and P. Zimmermann looked for the worst cases for correct rounding of the exponential function in the IEEE 754r decimal64 format. In their contribution they describe this search and their findings.

The subtleties of implementations of floating-point arithmetic are also important in the contribution of B. Lambov. He is the author of the package RealLib for fast computations over the real numbers with arbitrary precision. In his contribution he presents an implementation of double precision interval arithmetic, which is part of this package. The implementation makes efficient use of the single-instruction-multiple-data SSE-2 instruction and register set extensions. He describes the ideas needed to fit interval arithmetic to this set of instructions, compares the performance of this implementation with the performance of other interval arithmetic packages, and discusses possible hardware extensions that might significantly increase the performance of interval arithmetic.

G.F. Corliss, R.B. Kearfott, N. Nedialkov, J.D. Pryce, and S. Smith have given themselves the mission to "gather, organize and make available" interval software and libraries for developers that are not experts in interval arithmetic. Their goal is to offer a library of interval tools that are seamlessly integrated and that can solve a large variety of problems. Their paper gives an overview of different aspects of this project. They discuss 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. We wish them success!

Applications

The last two papers present methods and algorithms which have been developed in order to solve real-life problems.

E. Auer, A. Rauh, E. P. Hofer, and W. Luther present a new method for the integration of ODEs with initial value conditions, which yields a reliable enclosure of the trajectory. This method is implemented in the ValEncIA-IVP solver. This solver has been integrated into MOBILE, yielding the template-based tool SmartMobile. The tool MOBILE enables the user to model mechanical systems by building them out of predefined components such as joints, balls, etc. directly as executable programs that can simulate the behavior of these systems. This behavior is governed by ODEs. The paper also discusses possible strategies for reducing the overestimation which is due to the wrapping effect in interval arithmetic.

The problem addressed in the paper by S. Kempken and W. Luther is the modeling of traffic in queuing and service networks. Usually a stochastic modeling

is used, based on semi-Markov processes. The goal is to determine characteristic values for the considered network: probabilities and autocorrelation parameters. The authors express the autocorrelation parameters as a sum of exponential terms and then propose a method to obtain reliable and tight enclosures of the sought parameters, via linear programming techniques to obtain the lower and upper endpoints of intervals, when applicable. They use these enclosures to study the transient and steady states and to compute the time required to reach the steady state, i.e., the time where the transient state intersects the steady state.

The articles show that there are already many connections between the various disciplines concerned with the reliable implementation of real number algorithms. We believe that further cooperation and discussions between the different communities will be fruitful for the reliable solution of numerical and geometric problems.

Finally, we thank the participants of the Dagstuhl seminar for their talks and discussions, the authors of the papers in this book for their contributions and their cooperation in making this volume as accurate and readable as possible, and the referees for their careful work. We thank also Alfred Hofmann, the editor of the LNCS series of Springer, for making it possible to publish this post-seminar volume in the LNCS series, and the people at Schloss Dagstuhl for their hospitality and their great efficiency in organizing this seminar.

March 2008

Peter Hertling
Christoph M. Hoffmann
Wolfram Luther
Nathalie Revol

Table of Contents