C++ Toolbox for Verified Computing I

R. Hammer M. Hocks
U. Kulisch D. Ratz

C++ Toolbox for Verified Computing I

Basic Numerical Problems

Theory, Algorithms, and Programs

With 29 Figures

Springer

Prof. Dr. Ulrich Kulisch
Dr. Rolf Hammer
Dr. Matthias Hocks
Dr. Dietmar Ratz
Institut für Angewandte Mathematik
Universität Karlsruhe
D-76128 Karlsruhe

Cover figure: Function of Levy (see also page 337)

Mathematics Subject Classification (1991): 65-01, 65-04, 65F, 65G10, 65H, 65K

ISBN-13:978-3-642-79653-1

Library of Congress Cataloging-in-Publication Data
C++ toolbox for verified computing: theory, algorithms, and programs
R.Hammer [et al.]. p.cm. - Includes bibliographical references and index. Contents:
v. 1. Basic numerical problems.
ISBN-13:978-3-642-79653-1 e-ISBN-13:978-3-642-79651-7
DOI: 10.1007/978-3-642-79651-7
1. C++ (Computer program language) I. Hammer, (Rolf), 1961- .
QA76.73.C153C18 1995 519.4'0285'5133--dc20 95-10173 CIP

Typesetting: Camera ready by authors using $\TeX$
SPIN 10492623 41/3142 – 5 4 3 2 1 0 – Printed on acid-free paper

Preface

Our aim in writing this book was to provide an extensive set of C++ programs for solving basic numerical problems with *verification of the results*. This *C++ Toolbox for Verified Computing I* is the C++ edition of the *Numerical Toolbox for Verified Computing I*. The programs of the original edition were written in PASCAL–XSC, a PASCAL eXtension for Scientific Computation. Since we published the first edition we have received many requests from readers and users of our tools for a version in C++.

We take the view that C++ is growing in importance in the field of numerical computing. C++ includes C, but as a typed language and due to its modern concepts, it is superior to C. To obtain the degree of efficiency that PASCAL–XSC provides, we used the C–XSC library. C–XSC is a C++ class library for eXtended Scientific Computing. C++ and the C–XSC library are an adequate alternative to special XSC-languages such as PASCAL–XSC or ACRITH–XSC. A shareware version of the C–XSC library and the sources of the toolbox programs are freely available via anonymous ftp or can be ordered against reimbursement of expenses.

The programs of this book do not require a great deal of insight into the features of C++. Particularly, object oriented programming techniques are not required. However, the reader should be familiar with writing programs in a high-level computer language such as PASCAL, C, or FORTRAN. This book is particularly useful for those programmers who have already worked with the PASCAL–XSC edition but have only little knowledge in the C++ language. For those readers our book may be a source of inspiration when switching from PASCAL to C++.

We want to thank the readers of the original version of this book for their overwhelmingly positive comments. We incorporated some minor modifications in our algorithms which take into account the highly valuable suggestions and comments we received from numerous readers, as well as our own experience while using the toolbox. Some errors and misprints in the original edition have now been corrected. Nevertheless, we encourage readers, especially those of this C++ edition, to keep on communicating error reports to us.

Special thanks to Andreas Wiethoff who supported us in all questions concerning C++ and the C–XSC library. Last but not least, we wish to express again our appreciation to all colleagues whose advice helped produce the original edition, particularly those mentioned in the following preface of the PASCAL–XSC edition.

Karlsruhe, December 1994 *The Authors*

Preface to the PASCAL–XSC Edition

As suggested by the title of this book *Numerical Toolbox for Verified Computing*, we present an extensive set of sophisticated tools to solve basic numerical problems with a *verification of the results*. We use the features of the scientific computer language PASCAL–XSC to offer modules that can be combined by the reader to his/her individual needs. Our overriding concern is reliability – the *automatic verification of the result* a computer returns for a given problem. All algorithms we present are influenced by this central concern. We must point out that there is no relationship between our methods of *numerical result verification* and the methods of *program verification* to prove the correctness of an implementation for a given algorithm.

This book is the first to offer a general discussion on

- arithmetic and computational reliability,
- analytical mathematics and verification techniques,
- algorithms, and
- (most importantly) actual implementations in the form of working computer routines.

Our task has been to find the right balance among these ingredients for each topic. For some topics, we have placed a little more emphasis on the algorithms. For other topics, where the mathematical prerequisites are universally held, we have tended towards more in-depth discussion of the nature of the computational algorithms, or towards practical questions of implementation. For all topics, we present examples, exercises, and numerical results demonstrating the application of the routines presented.

The different chapters of this volume require different levels of knowledge in numerical analysis. Most numerical toolboxes have, after all, tools at varying levels of complexity. Chapters 2, 3, 4, 5, 6, and 10 are suitable for an advanced undergraduate course on numerical computation for science or engineering majors. Other chapters range from the level of a graduate course to that of a professional reference. An attractive feature of this approach is that you can use the book at increasing levels of sophistication as your experience grows. Even inexperienced readers can use our most advanced routines as black boxes. Having done so, these readers can go back and learn what secrets are inside.

The central theme in this book is that practical methods of numerical computation can be simultaneously efficient, clever, clear, and (most importantly) reliable.

We firmly reject the alternative viewpoint that such computational methods must necessarily be so obscure and complex as to be useful only in "black box" form where you have to believe in any calculated result.

This book introduces many computational verification techniques. We want to teach you to take apart these black boxes and to put them back together again, modifying them to suit your specific needs. We assume that you are mathematically literate, i.e. that you have the normal mathematical preparation associated with an undergraduate degree in a mathematical, computational, or physical science, or engineering, or economics, or a quantitative social science. We assume that you know how to program a computer and that you have some knowledge of scientific computation, numerical analysis, or numerical methods. We do not assume that you have any prior formal knowledge of numerical verification or any familiarity with interval analysis. The necessary concepts are introduced.

Volume 1 of *Numerical Toolbox for Verified Computing* provides algorithms and programs to solve basic numerical problems using automatic result verification techniques.

Part I contains two introductory chapters on the features of the scientific computer language PASCAL–XSC and on the basics and terminology of interval arithmetic. Within these chapters, the important correlation between the arithmetic capability and computational accuracy and mathematical fixed-point theory is also discussed.

Part II addresses one-dimensional problems: evaluation of polynomials and general arithmetic expressions, nonlinear root-finding, automatic differentiation, and optimization. Even though only one-dimensional problems treated in this part, the verification methods sometimes require multi-dimensional features like vector or matrix operations.

In Part III, we present routines to solve multi-dimensional problems such as linear and nonlinear systems of equations, linear and global optimization, and automatic differentiation for gradients, Hessians, and Jacobians.

Further volumes of *Numerical Toolbox for Verified Computing* are in preparation covering computational methods in the field of linear systems of equations for complex, interval, and complex interval coefficients, sparse linear systems, eigenvalue problems, matrix exponential, quadrature, automatic differentiation for Taylor series, initial value, boundary value and eigenvalue problems of ordinary differential equations, and integral equations. Editions of the program source code of this volume in the C++ computer language are also in preparation.

Some of the subjects that we cover in detail are not usually found in standard numerical analysis texts. Although this book is intended primarily as a reference text for anyone wishing to apply, modify, or develop routines to obtain mathematically certain and reliable results, it could also be used as a textbook for an advanced course in scientific computation with automatic result verification.

We express our appreciation to all our colleagues whose comments on our book were constructive and encouraging, and we thank our students for their help in testing our routines, modules, and programs. We are very grateful to Prof. Dr. George Corliss (Marquette University, Milwaukee, USA) who helped to polish the text and

the contents. His comments and advice based on his numerical and computational experience greatly improved the presentation of our tools for Verified Computing.

Karlsruhe, September 1993 *The Authors*

The computer programs in this book

and a shareware version of the C–XSC library are available in several machine-readable formats. To purchase diskettes in IBM-PC compatible format, use the order form at the end of the book. The programs and the library are also available by anonymous ftp from

`iamk4515.mathematik.uni-karlsruhe.de` (129.13.129.15)

in subdirectory

`pub/toolbox/cxsc`.

Technical questions, corrections, and requests for information on other available formats and software products should be directed to *Numerical Toolbox Software, Institut für Angewandte Mathematik, Universität Karlsruhe, D-76128 Karlsruhe, Germany, e-mail: toolbox@iampc4.mathematik.uni-karlsruhe.de.*

Table of Contents

Appendix

List of Figures