Preface

This book is intended primarily for those not yet familiar with methods for computing with intervals of real numbers and what can be done with these methods.

Using a pair $[a, b]$ of computer numbers to represent an interval of real numbers $a \leq x \leq b$, we define an arithmetic for intervals and interval valued extensions of functions commonly used in computing. In this way, an interval $[a, b]$ has a dual nature. It is a new kind of number pair, and it represents a *set* $[a, b] = \{x : a \leq x \leq b\}$. We combine set operations on intervals with interval function evaluations to get algorithms for computing enclosures of sets of solutions to computational problems. A procedure known as outward rounding guarantees that these enclosures are rigorous, despite the roundoff errors that are inherent in finite machine arithmetic. With interval computation we can program a computer to find intervals that contain—with absolute certainty—the exact answers to various mathematical problems. In effect, interval analysis allows us to compute with sets on the real line. Interval vectors give us sets in higher-dimensional spaces. Using multinomials with interval coefficients, we can compute with sets in function spaces.

In applications, interval analysis provides rigorous enclosures of solutions to model equations. In this way we can at least know for sure what a mathematical model tells us, and, from that, we might determine whether it adequately represents reality. Without rigorous bounds on computational errors, a comparison of numerical results with physical measurements does not tell us how realistic a mathematical model is.

Methods of computational error control, based on order estimates for approximation errors, are not rigorous—nor do they take into account rounding error accumulation. Linear sensitivity analysis is not a rigorous way to determine the effects of uncertainty in initial parameters. Nor are Monte Carlo methods, based on repetitive computation, sampling *assumed* density distributions for uncertain inputs. We will not go into interval statistics here or into the use of interval arithmetic in fuzzy set theory.

By contrast, interval algorithms are designed to automatically provide rigorous bounds on accumulated rounding errors, approximation errors, and propagated uncertainties in initial data during the course of the computation.

Practical application areas include chemical and structural engineering, economics, control circuitry design, beam physics, global optimization, constraint satisfaction, asteroid orbits, robotics, signal processing, computer graphics, and behavioral ecology.

Interval analysis has been used in rigorous computer-assisted proofs, for example, Hales' proof of the Kepler conjecture.

An interval Newton method has been developed for solving systems of nonlinear equations. While inheriting the local quadratic convergence properties of the ordinary Newton method, the interval Newton method can be used in an algorithm that is mathematically guaranteed to find all roots within a given starting interval.

Interval analysis permits us to compute interval enclosures for the exact values of integrals. Interval methods can bound the solutions of linear systems with inexact data. There are rigorous interval branch-and-bound methods for global optimization, constraint satisfaction, and parameter estimation problems.

The book opens with a brief chapter intended to get the reader into a proper mindset for learning interval analysis. Hence its main purpose is to provide a bit of motivation and perspective. Chapter 2 introduces the interval number system and defines the set operations (intersection and union) and arithmetic operations (addition, subtraction, multiplication, and division) needed to work within this system.

The first applications of interval arithmetic appear in Chapter 3. Here we introduce outward rounding and demonstrate how interval computation can automatically handle the propagation of uncertainties all the way through a lengthy numerical calculation. We also introduce INTLAB, a powerful and flexible MATLAB toolbox capable of performing interval calculations.

In Chapter 4, some further properties of interval arithmetic are covered. Here the reader becomes aware that not all the familiar algebraic properties of real arithmetic carry over to interval arithmetic. Interval functions—residing at the heart of interval analysis—are introduced in Chapter 5. Chapter 6 deals with sequences of intervals and interval functions, material needed as preparation for the iterative methods to be treated in Chapter 7 (on matrices) and Chapter 8 (on root finding). Chapter 9 is devoted to integration of interval functions, with an introduction to automatic differentiation, an important tool in its own right. Chapter 10 treats integral and differential equations. Finally, Chapter 11 introduces an array of applications including several of those (optimization, etc.) mentioned above.

Various appendices serve to round out the book. Appendix A offers a brief review of set and function terminology that may prove useful for students of engineering and the sciences. Appendix B, the quick-reference Formulary, provides a convenient handbook-style listing of major definitions, formulas, and results covered in the text. In Appendix C we include hints and answers for most of the exercises that appear throughout the book. Appendix D discusses Internet resources (such as additional reading material and software packages—most of them freely available for download) relevant to interval computation. Finally, Appendix E offers a list of INTLAB commands.

Research, development, and application of interval methods is now taking place in many countries around the world, especially in Germany, but also in Austria, Belgium, Brazil, Bulgaria, Canada, China, Denmark, Finland, France, Hungary, India, Japan, Mexico, Norway, Poland, Spain, Sweden, Russia, the UK, and the USA. There are published works in many languages. However, our references are largely to those in English and German, with which the authors are most familiar. We cannot provide a comprehensive bibliography of publications, but we have attempted to include at least a sampling of works in a broad range of topics.

The assumed background for the first 10 chapters is basic calculus plus some familiarity with the elements of scientific computing. The application topics of Chapter 11 may require a bit more background, but an attempt has been made to keep much of the presentation accessible to the nonspecialist, including senior undergraduates or beginning graduate students in engineering, the sciences (physical, biological, economic, etc.), and mathematics.

Of the various interval-based software packages that are available, we chose INTLAB for several reasons. It is fully integrated into the interactive, programmable, and highly popular MATLAB system. It is carefully written, with all basic interval computations represented. Finally, both MATLAB and INTLAB code can be written in a fashion that is clear and easy to debug.

We wish to cordially thank George Corliss, Andreas Frommer, and Siegfried Rump, as well as the anonymous reviewers, for their many constructive comments. We owe Siegfried Rump additional thanks for developing INTLAB and granting us permission to use it in this book. Edward Rothwell and Mark Thompson provided useful feedback on the manuscript. We are deeply grateful to the staff of SIAM, including Senior Acquisitions Editor Elizabeth Greenspan, Developmental Editor Sara J. Murphy, Managing Editor Kelly Thomas, Production Manager Donna Witzleben, Production Editor Ann Manning Allen, Copy Editor Susan Fleshman, and Graphic Designer Lois Sellers.

The book is dedicated to our wives: Adena, Ruth, and Beth.

Ramon E. Moore

R. Baker Kearfott

Michael J. Cloud