, , 1–5 ()

Review for **Computational Complexity and Feasibility of Data Processing and Interval Computations,** by Vladik Kreinovich, Anatoly Lakeyev, Jiří Rohn, and Patrick Kahl

R. BAKER KEARFOTT rbk@usl.edu
Department of Mathematics, U.S.L. Box 4-1010, Lafayette, LA 70504-1010 USA

Editor:

Abstract.

Keywords: computational complexity, interval computations

1. Introduction

Interval computations are used to account for uncertainties in data, to provide rigor in floating point computations, and to construct algorithms not possible without bounds on the ranges of quantities. Because of the accounting and rigor, the emphasis in assessing the practicality of interval algorithms is shifted from robustness and accuracy to whether such algorithms can complete in a reasonable time. *Computational Complexity and Feasibility of Data Processing and Interval Computations* addresses this issue within the context of formalized complexity theory.

2. Subjects Treated

The main part of the book revolves around these basic problems:

1. the *basic problem in interval computations*, consisting of computing the exact range
$$\{f(x_1, x_2, \ldots, x_n) \mid x_1 \in \boldsymbol{x}_1, \ldots, x_n \in \boldsymbol{x}_n\},$$
of a function f of n variables, where the variables x_i range over the intervals $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i]$;

2. solution of linear systems of equations;

3. constrained and unconstrained optimization;

4. solution of nonlinear systems of equations;

5. differential equations.

6. verification of properties of matrices, such as non-singularity, positive definiteness, Schur stability (eigenvalues with negative real parts), etc.

In addition, five appendices give miscellaneous philosophical perspectives, and cover topics that are not an indivisible part of the main development.

The first chapter gives an informal introduction to both interval computations and to computational complexity, while the second chapter introduces the formalities of NP hardness, NP completeness, feasibility, etc. The remainder of the book gives formal complexity results for each of the problems enumerated above. Results are given and are compared for both point data (e.g. non-interval matrices and non-interval right-hand-sides for linear systems) and interval data. Comparisons are also made with regard to other properties, such as for f a polynomial of increasing degree, number of variables increasing versus fixed number of variables but increasing number of terms, etc. The book appears to be a surprisingly unified treatment of the known complexity results in this area.

3. Significance of Complexity Analysis

The introductory chapters motivate the definitions and procedures of complexity theory from practical considerations. The authors also express the hope that the work will help guide algorithm designers and users concerning what is possible. And, indeed, the book achieves this goal. For example, computing the values where an objective function $f(x_1, x_2, \ldots, x_n)$ attains its maximum is listed as doable in polynomial time for f linear, quadratic, and cubic, but exponential time or worse for f of quartic or higher degree (while finding the maximum itself is simply NP-hard for quartic or higher degree). Actual general global optimization algorithms do exhibit an exponential dependence on the data size, for a variety of problems.

However, as the authors acknowledge, such formal analysis has limitations. For example, some general global optimization algorithms exhibit polynomial-time behavior on certain families of practical problems.

As another example, in Appendix D, it is proven that no algorithm is possible that returns exactly the two roots of a polynomial that has exactly two roots. The proof involves the function

$$f_\alpha(x) = (x - 1 - \alpha^2) \cdot (x - 1 + \alpha^2) \cdot \left((x+1)^2 + \alpha^2\right)$$

(with α a constructive real number, and some additional underlying formal structure). Existence of an algorithm would imply the ability to compute whether α is non-zero, something that is known to be impossible. However, practical algorithms, for α small, will return regions around $x = -1$ and $x = 1$ for moderate range accuracy, as they should. When $\alpha \neq 0$, the region around $x = -1$ disappears when a high-enough range accuracy is used, as it should; the main practical difficulty is that it is not known a priori how high the range accuracy should be.

Additional examples of the pitfalls of formalized analysis come from the early history of modern scientific computation. It is well known that the worst-case error growth of Gaussian elimination with partial pivoting is rapid, relative to the order of the linear system; yet, for many practical problems, Gaussian elimination with partial pivoting works well. Similarly, it is known that the worst-case behavior of the simplex method for linear programming is exponential, yet is polynomial for many families of practical problems. Although these examples deal with particular algorithms, some general results in complexity theory can be viewed as having an analogous relationship to practical problems.

The authors of *Computational Complexity and Feasibility* motivate the formalism well, explaining the underlying philosophy and how the theory is modified to reflect practice.

4. Style and Organization

The book successfully pulls together a catalog of results from various sources. The chapters are organized from the most basic to more involved computational tasks. Within each computational task, various aspects of the complexity theory are analyzed. Thus, Chapters 1 and 2 introduce common underlying concepts and definitions. Chapter 3 deals with the basic problem of interval computations, Chapter 4 considers the complexity of the basic problem with a fixed number of variables, Chapter 5 considers complexity when f is a fixed-order polynomial, Chapter 6 considers the basic problem when f is a polynomial with bounded coefficients, Chapter 7 considers a fixed set of f and arbitrary intervals, Chapter 8 considers arbitrary polynomial f but fixed intervals, Chapter 9 considers f defined with subsets of the usual operations $\{+, -, \times\}$, and Chapter 10 gives complexity results for f a quotient of linear functions. Chapters 11 through 14 give a similar treatment of solution of linear systems of equations.

Chapters 15 and 16 give positive interpretations to some of the seemingly negative complexity results. Namely, since many interval computations are NP-hard, good heuristics can be shared with other NP-hard problems. Also, the average case complexity of interval computations, with small input intervals, is polynomial in many cases.

Chapter 17 deals with interval optimization, beginning with the basic problem of interval computations as an example, while Chapter 18 treats solution of nonlinear systems of equations. Chapter 19 treats approximation of interval functions, and Chapter 20 treats solution of differential equations. Chapters 21 and 22 present complexity results for properties of matrices, such as verifying that a matrix is non-singular or verifying that a matrix has eigenvalues with non-negative real parts.

Chapters 23 and 24 deal with complexity of generalizations of interval computations. In particular, ellipsoids approximate ranges of quadratic functions and normal distributions better than interval vectors; certain generalizations of ellipsoids are also appropriate for other statistical distributions. Also, arithmetic on disjoint sets of intervals and on intervals one or both of whose endpoints is infinite is considered.

The last two chapters of the main text compare interval arithmetic to a certain statistical estimation procedure, and consider complexity if only discrete values from the input intervals are allowed.

The book can actually be read successfully on four levels. The top level is the table of contents. On the second level, an introductory paragraph prefaces each chapter. These paragraphs can be read consecutively to get an overall view. They do a good job of connecting the results together and explaining significance.

The proofs of the main results are separated from the statements, and appear at the end of each chapter or, in one case, in a separate chapter. Thus, the third level of the book consists of the statement and summary of the complexity results, at the beginning of each chapter. The book is highly accessible and readable on this level. Results are presented with formal definitions and theorems, and examples are given. Numerous tables appear. The data in these tables is somewhat redundant, in the sense that, when additional results are presented, data in a previous table sometimes appears again, with the new results represented as a new line. But this redundancy results in easier reading and comprehension. An example of a table appears on page 183:

Objective function	Computing the values $x_1, \ldots, x_n$ for which the maximum is attained	Computing stationary points
Linear	Linear time	Linear time
Quadratic	Polynomial time	Polynomial time
Cubic	Polynomial time	Exponential time (or worse)
Quartic	NP-hard	Exponential time (or worse)
5-th and higher degree	NP-hard	Exponential time (or worse)

The fourth level consists of the technical details and proofs at the end of each chapter. Many of these proofs are constructive in the sense that algorithms or counterexamples are exhibited. The authors indicate which algorithms it actually makes sense to use.

The book is highly cross-referenced, and 436 references appear. Although (as is inevitable), there are several minor typographical errors, the notation, highlighting, and typesetting are uniform throughout, and of a high quality.

5. Who Should Read the Book

Due to its style, *Computational Complexity and Feasibility* should be of interest to a fairly wide audience. It is a good reference to current results in complexity of interval computations, so researchers in interval computations will be interested.

Since open problems are mentioned, researchers in complexity theory of interval computations will also find the book of use. Since connections are made to general complexity theory, the book should be of value within the complexity theory community as a whole.

Although primarily a monograph and a reference, the book contains informal motivation, and also is self-contained. Thus, the book could be of use in a graduate course on interval computations or on complexity theory.