CONTENTS

LIST OF FIGURES vii
LIST OF TABLES xi
PREFACE xiii

1 PRELIMINARIES 1
1.1 Interval Arithmetic 3
1.2 Interval Linear Systems 18
1.3 Derivatives and Slopes 26
1.4 Automatic Differentiation and Code Lists 36
1.5 Interval Newton Methods and Interval Fixed Point Theory 50
1.6 The Topological Degree 66

2 SOFTWARE ENVIRONMENTS 71
2.1 INTLIB 71
2.2 Fortran 90 Interval and Code List Support 78
2.3 Other Software Environments 102

3 ON PRECONDITIONING 113
3.1 The Inverse Midpoint Preconditioner 115
3.2 Optimal Linear Programming Preconditioners 120

4 VERIFIED SOLUTION OF NONLINEAR SYSTEMS 145
4.1 An Overall Branch and Bound Algorithm 146
4.2 Approximate Roots and Epsilon-Inflation 150
4.3 Tessellation Schemes 154
4.4 Description of Provided Software 159
4.5 Alternate Algorithms and Improvements 167

5 OPTIMIZATION 169
5.1 Background and Historical Algorithms 170
5.2 Handling Constraints 177
5.3 Description of Provided Software 199

6 NON-DIFFERENTIABLE PROBLEMS 209
6.1 Extensions of Non-Smooth Functions 210
6.2 Use in Interval Newton Methods 218

7 USE OF INTERMEDIATE QUANTITIES IN THE EXPRESSION VALUES 227
7.1 The Basic Approach 227
7.2 An Alternate Viewpoint – Constraint Propagation 230
7.3 Application to Global Optimization 230
7.4 Efficiency and Practicality 232
7.5 Provided Software 233
7.6 Exercises 234

REFERENCES 235

INDEX 255
# LIST OF FIGURES

## Chapter 1

1.1 The united solution set $\Sigma(A, B)$ for the system (1.19)  
1.2 $\Sigma(A, B) \cap X$ can still be bounded when $AX = B$ is underdetermined.  
1.3 The difference between the interval slope $S^i(f, x, \bar{x})$ and the derivative range $f'_0$ for $x = [0, 2]$, $\bar{x} = 1$ and $f(x) = x^2$  
1.4 Two ways of computing the change in $\phi$ between $(\bar{x}_1, \bar{x}_2)$ and $(x_1, x_2)$  
1.5 Sample program to compute a floating point function value from a code list  
1.6 Non-uniqueness with slopes with $f(x) = (x^2 - 1)(x + 2)$, $x = [-2, 3]$ and $\bar{x} = 3$  

## Chapter 2

2.1 A FORTRAN-77 program using INTLIB  
2.2 Illustration of use of the interval data type  
2.3 A univariate interval Newton method program, used to generate the data in Table 1.3  
2.4 Program that generates a code list for $f(x) = x^4 + x^3 + x$  
2.5 Operation lines of the code list produced by the program of Figure 2.4  
2.6 Sample OVERLOAD.CFG file  
2.7 Accumulating a product for a dependent variable  
2.8 Reassignment of a dependent variable  
2.9 The code list operation lines for the program in Figure 2.8  
2.10 (a) Function USR for Example 2.1  
2.10 (b) Function USRD for Example 2.1  
2.11 A program to generate a code list for $f_1(x) = f(x) - 2$, where $f(x)$ is as in Example 2.1
2.12 Generation of a derivative code list corresponding to the code list in file EX1.CDL 92
2.13 Operation lines of the code list for the derivative of \( f(x) = x^4 + x^3 + x \) 92
2.14 Algebraic interpretation of the code list in Figure 2.13 92
2.15 A code list for the function in Example 2.2 93
2.16 The derivative code list corresponding to Figure 2.15 93
2.17 A program to generate a code list for \( \phi(x_1, x_2) = x_1^2 + x_2^2 \), subject to \( x_1 + x_2 - 1 = 0 \) 100
2.18 Evaluation of the constraints in Example 2.3 100

Chapter 3
3.1 Action of a C-preconditioner 125
3.2 Action of an E-preconditioner 125
3.3 Appropriate situation for a left-optimal preconditioner 126
3.4 Action of a magnitude-optimal C-preconditioner: \( \max\{a, b\} \) is minimized. 127
3.5 Action of a magnitude-optimal E-preconditioner: \( \min\{a, b\} \) is maximized. 128

Chapter 4
4.1 Complementation of a box in a box 154
4.2 Complementation of a box in a list of boxes 156
4.3 Bisection of the second coordinate according to maximum smear, for Example 4.1 158
4.4 Program to produce a code list for a counterexample to a method of Branin 161
4.5 Box data file BRANIN.DT1 162
4.6 RUN_ROOTS_DELETE for BRANIN.CDL and BRANIN.DT1 162
4.7 Sample output to RUN_ROOTS_DELETE 163
4.8 Box data file BRANIN.DT2 163
4.9 Use of RUN_ROOTS_DELETE for uniqueness verification 164

Chapter 5
5.1 “Peeling” a box to produce lower-dimensional boundary elements 181
List of Figures

5.2 “Peeling” the box into lower-dimensional boundary elements 183
5.3 Proving that there exists a feasible point of an underdetermined constraint system 186
5.4 Proving existence in a reduced space when the approximate feasible point satisfies bound constraints 186
5.5 A common degenerate case, when $\tilde{X}$ must be perturbed 187
5.6 Geometry of a feasibility proof with LP preconditioners 190
5.7 Box data file GOULD.DT1 202

Chapter 6

6.1 Program that generates a code list for $f(x) = x^2$ if $x < 1$, $f(x) = 2x - 1$ if $x \geq 1$ 211
6.2 Operation lines of the code list produced by the program of Figure 6.1 212
6.3 Computation of slope bounds of a discontinuous function 214
x RIGOROUS GLOBAL SEARCH: CONTINUOUS PROBLEMS
LIST OF TABLES

Chapter 1

1.1 Some example operations and corresponding numerical codes 44
1.2 Tabular representation of the output to the program (1.36) 48
1.3 Illustration of quadratic convergence of the univariate interval
   Newton method for \( f(x) = x^2 - 4 \) 54
1.4 Illustration of quadratic convergence of the Krawczyk method
   for \( F \) as in Example 1.5 58

Chapter 2

2.1 Elementary arithmetic routines in INTLIB 74
2.2 Standard function routines in INTLIB 75
2.3 Utility functions in INTLIB 75
2.4 Special interval functions in module INTERVAL_ARITHMETIC 80
2.5 Logical and interval operators in module
   INTERVAL_ARITHMETIC 81
2.6 Operations in module OVERLOAD and program MAKE_GRADIENT 84
2.7 Additional operations supported in module OVERLOAD 85
2.8 Operational complexity – generic routines 98
2.9 Ratio of interval to floating point CPU times CPURAT for
   ACRITH-XSC on an IBM 3090 105
2.10 Ratio of interval to floating point CPU times CPURAT for
    INTERVAL_ARITHMETIC on a Sun SPARC 20 model 51 105

Chapter 5

5.1 Summary attributes of various global optimization algorithms 177
5.2 Summary of handling of constraints in various global opti-
   mization algorithms 178
5.3 Summary of 3 methods of handling constraints for Example 5.2

Chapter 6

6.1 Iterates of the interval Newton method for \( f(x) = |x^3 - x| - 2x + 2 \)
PREFACE

This work grew out of several years of research, graduate seminars and talks on the subject. It was motivated by a desire to make the technology accessible to those who most needed it or could most use it. It is meant to be a self-contained introduction, a reference for the techniques, and a guide to the literature for the underlying theory. It contains pointers to fertile areas for future research. It also serves as introductory documentation for a Fortran 90 software package for nonlinear systems and global optimization.

The subject of the monograph is deterministic, automatically verified or rigorous methods. In such methods, directed rounding and computational fixed-point theory are combined with exhaustive search (branch and bound) techniques. Completion of such an algorithm with a list of solutions constitutes a rigorous mathematical proof that all of the solutions within the original search region are within the output list.

The monograph is appropriate as an introduction to research and technology in the area, as a desk reference, or as a graduate-level course reference. Knowledge of calculus, linear algebra, and elementary numerical analysis is assumed. Interval computations are presented from the beginning, although the more advanced material is mainly to support development of ideas in nonlinear systems and global optimization. The style is meant to balance the need to inform newcomers with the need to provide concise treatment for experts. The emphasis in the advanced topics is on the author's own contributions to the field. The book contains numerous references and cross-references, as well as an extensive index. A logical thread is provided for easy initial reading, with gateways for subsequent in-depth study.

Although the major goal of the book is to explain techniques and software for rigorous solution of nonlinear systems and rigorous, deterministic global optimization, the extensive introduction in Chapter 1 is suitable as a general introduction or reference for interval arithmetic. Thus, this introduction can also be useful in studying verified quadrature or verified solutions of differential equations, for example.
Similarly, the software described in Chapter 2 includes general-use interval arithmetic software. This software includes a Fortran-90 module for interval arithmetic and Fortran 90 modules for automatic differentiation (termed INTLIB90), freely available from the author (email: rbg@usl.edu), and also includes more special-purpose nonlinear equations and global optimization codes (termed INTOPT90). Chapter 2 also contains a review of alternate programming languages and packages for interval computations, with a discussion of the advantages and disadvantages of each.

In Chapter 3, previously unpublished material on optimal preconditioners for interval linear systems is presented. Illustrations and examples are meant to point the reader to the contexts in which such preconditioners can be advantageous, as well as to give necessary explanation for implementation. (The author's software contains routines for one type of optimal preconditioner.)

Chapter 4 contains an introduction to algorithmic constructs for global search methods for solutions to nonlinear systems of equations. These constructs include overall algorithm structure, as well as methods for subdividing the region and taking advantage of accurate approximate solutions. Some of these techniques and algorithmic constructs are shared by global optimization algorithms, described in the next chapter. Chapter 4 also contains a description of the author’s software for nonlinear equations, in an overall package (termed INTOPT90).

Chapter 5 deals with actual global optimization techniques and algorithms. Beginning with an overall view, historical algorithms and recent, more sophisticated algorithms are reviewed. This is followed by an extensive review of how inequality and equality constraints are handled, including some of the author’s proposed techniques and experience. The chapter concludes with a description of the routines available in INTOPT90.

Chapter 6 deals with special techniques for non-differentiable problems, such as $l_1$ and $l_{\infty}$ optimization. In particular, formulas are listed for interval extensions of non-smooth and discontinuous functions, as well as representation of derivative information for non-smooth and discontinuous functions. Surprisingly, such interval extensions, even for discontinuous functions, can be effective in interval Newton methods; examples are given, and a convergence analysis is reviewed. Support for the techniques is available in the author’s software, INTLIB90.

Chapter 7 deals with using the interval values of subexpressions, obtained while evaluating objective functions and gradients. These techniques are tied to the
preconditioner techniques of Chapter 3, and are also related to the established field of constraint propagation.

Throughout the book, techniques and software are presented not only to explain available work, but also to guide interested readers to future improvements or to ways of using individual techniques in different contexts.

Elements of the theory deemed most relevant are presented. However, theorems appear within the context of the meaning and practical impact of their results. Where proofs are presented, it is mainly for the purpose of giving additional insight. Where proofs are not presented, references to the original research are given.

Exercises after many sections amplify the preceding presentation.

The references are available electronically in BuTεX form, via anonymous FTP from

\texttt{interval.usl.edu}

in the directory

\texttt{pub/interval\_math/bibliographies/optimization_book.bib}

I wish to thank my former student Dr. Xiaofa Shi, and my colleagues George Corliss, Panos Pardalos, Dietmar Ratz, and Siegfried Rump for the careful reading and suggestions that improved this book considerably. I also would like to thank Layne Watson for introducing me to \TeX and for encouraging me, early on, to work hard.