GLOBAL OPTIMIZATION USING INTERVAL ANALYSIS

Second Edition, Revised and Expanded

ELDON HANSEN
Consultant
Los Altos, California

G. WILLIAM WALSTER
Sun Microsystems Laboratories
Mountain View, California, U.S.A.

MARCEL DEKKER

MARCEL DEKKER, INC. NEW YORK • BASEL

The first edition was *Global Optimization Using Interval Analysis*, Eldon Hansen, ed. (Marcel Dekker, 1992).

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress.

ISBN: 0-8247-4059-9

This book is printed on acid-free paper.

Headquarters
Marcel Dekker, Inc., 270 Madison Avenue, New York, NY 10016, U.S.A.
tel: 212-696-9000; fax: 212-685-4540

Distribution and Customer Service
Marcel Dekker, Inc., Cimarron Road, Monticello, New York 12701, U.S.A.
tel: 800-228-1160; fax: 845-796-1772

Eastern Hemisphere Distribution
Marcel Dekker AG, Hutgasse 4, Postfach 812, CH–4001 Basel, Switzerland
tel: 41-61-260-6300; fax: 41-61-260-6333

World Wide Web: http://www.dekker.com

The publisher offers discounts on this book when ordered in bulk quantities. For more information, write to Special Sales/Professional Marketing at the headquarters address above.

Current printing (last digit):

CR/SH 10 9 8 7 6 5 4 3 2 1

PRINTED IN THE UNITED STATES OF AMERICA

Chapter 1

INTRODUCTION

1.1 AN OVERVIEW

In mathematics, there are real numbers, a real arithmetic for combining them, and a real analysis for studying the properties of the numbers and the arithmetic. Interval mathematics is a generalization in which interval numbers replace real numbers, interval arithmetic replaces real arithmetic, and interval analysis replaces real analysis.

Numerical analysis is the study of computing with real (and other kinds of) numbers. Theoretical numerical analysis considers exact numbers and exact arithmetic, while practical numerical analysis considers finite precision numbers in which rounding errors occur. This book is concerned with both theoretical and practical interval analysis for computing with interval numbers.

In this book we limit our attention almost exclusively to real interval analysis. However, an analysis of complex intervals has been defined and used, beginning with Boche (1966). A complex "interval" can be a rectangle, a circle; or a more complicated set. Intervals of magnitude and phase can also be used. Some early publications discussing complex intervals are Alefeld (1968), Alefeld and Herzberger (1974), Gargantini (1975, 1976, 1978), Gargantini and Henrici (1972), Glatz (1975), Henrici (1975), Krier and Spellucci (1975), and Nickel (1969).

1.2 THE ORIGIN OF INTERVAL ANALYSIS

There are several types of mathematical computing errors. Data often contain measurement errors, or are otherwise uncertain because rounding errors generally occur, and approximations are made, etc. The purpose of interval analysis is to provide upper and lower bounds on the effect all such errors and uncertainties have on a computed quantity.

It is desirable to make interval bounds as narrow as possible. A major focus of interval analysis is to develop practical interval algorithms that produce sharp[1] (or nearly sharp) bounds on the solution of numerical computing problems. However, in practical problems with interval inputs, it is often sufficient to simply compute reasonably narrow interval bounds.

Several people independently had the idea of bounding rounding errors by computing with intervals; e.g., see Dwyer (1951), Sunaga (1958), Warmus (1956), (1960) and Wilkinson (1980). However, interval mathematics and analysis can be said to have begun with the appearance of R. E. Moore's book *Interval Analysis* in 1966. Moore's work transformed this simple idea into a viable tool for error analysis. In addition to treating rounding errors, Moore extended the use of interval analysis to bound the effect of errors from all sources, including approximation errors and errors in data.

1.3 THE SCOPE OF THIS BOOK

In this book we focus on a rather narrow part of interval mathematics. One of our goals is to describe algorithms that use interval analysis to solve the global (unconstrained or constrained) nonlinear optimization problem. We show that such problems can be solved with a guarantee that the computed bounds on the location and value of a solution are numerically correct. If there are multiple solutions, all will be found and correctly bounded. It is also guaranteed that the solution(s) are global and not just local.

Our optimization algorithms use interval linear algebra and interval Newton algorithms that solve systems of nonlinear equations. Consequently, we discuss these topics in some detail. Our discussion includes

[1] An interval bound is said to be *sharp* if it is as narrow as possible.

some historical information but is not intended to be exhaustive in this regard.

We also describe and use an extended interval arithmetic. In the past, it has been customary to exclude certain arithmetic operations in both real and interval arithmetic. Hanson (1968) and Kahan (1968) each described incomplete extensions of interval arithmetic in which endpoints of intervals are allowed to be infinite. The foundation for complete interval arithmetic extensions is described in Chapter 4. Alefeld (1968) (See also Hansen (1978b)) described a practical interval Newton algorithm in which division by an interval containing zero is allowed.

The extension of interval arithmetic that we describe is a closed[2] system with no exclusions of any arithmetic operations or values of operands. It includes division by zero and indeterminate forms such as $\frac{0}{0}$, $\infty - \infty$, $0 \times \infty$, and $\frac{\infty}{\infty}$, etc., that are normally excluded from real and extended (i.e., including infinities) real arithmetic systems. It is remarkable that interval analysis allows closure of systems containing such indeterminate forms and infinite values of variables. All the algorithms in this book can be implemented using these closed interval systems. The resulting benefits are increased generality and simpler code.

1.4 VIRTUES AND DRAWBACKS OF INTERVAL MATHEMATICS

The history of floating-point computing and resulting rounding errors are described in Section 4.11 of Hennessy and Patterson (1994). Interval analysis began as a tool for bounding rounding errors. Nevertheless, the belief persists that rounding errors can be easily detected in another way. The contention is that one need only compute a given result using, say single and double precision. If the two results agree to some number of digits, then these digits are correct.

[2]A closed system is one in which there are no undefined arithmetic operand-operator combinations.

1.4.1 Rump's Example

An example of Rump (1988) shows that this argument is not necessarily valid. Using IEEE-754 computers, the following form (from Loh and Walster (2002)) of Rump's expression with $x_0 = 77617$ and $y_0 = 33096$ replicates his original IBM S/370 results.

$$f(x, y) = (333.75 - x^2)y^6 + x^2(11x^2y^2 - 121y^4 - 2) + 5.5y^8 + \frac{x}{2y} \tag{1.4.1}$$

With round-to-nearest (the usual default) IEEE-754 arithmetic, the expression in (1.4.1) produces:

32-bit: $f(x_0, y_0) = 1.172604$
64-bit: $f(x_0, y_0) = 1.1726039400531786$
128-bit: $f(x_0, y_0) = 1.172603940053178631858834904520183 8$

All three results agree in the first seven decimal digits and thirteen digits agree in the last two results. Nevertheless, they are all completely incorrect. Even their sign is wrong.

Loh and Walster (2001) show that both Rump's original and the expression for $f(x, y)$ in (1.4.1) reduce to:

$$f(x_0, y_0) = \frac{x_0}{2y_0} - 2, \tag{1.4.2}$$

from which

$$f(x_0, y_0) = -0.827396059946821368141165095479816... \tag{1.4.3}$$

with the above values for x_0 and y_0.

Evaluating $f(x_0, y_0)$ in its unstable forms using interval arithmetic of moderate accuracy produces a wide interval (containing the correct value of

$f(x_0, y_0)$). The fact that the interval is wide even though the argument values are machine-representable is a warning that roundoff and catastrophic cancellation have probably occurred; and therefore higher-accuracy arithmetic is needed to get an accurate answer. In some cases, as is seen in the above example, rearranging the expression can reduce or eliminate the catastrophic cancellation.

1.4.2 Real Examples

Rump's example is contrived. However, rounding errors and the effects of cancellation impact computed results from important real world problems, as documented in:

```
www.math.psu.edu/dna/disasters/
```

and by Daumas (2002). For example, the failure of the Patriot Missile battery at Daharan was directly attributable to accumulation of roundoff errors; and the explosion of the Ariane 5 was caused by overflow. The Endeavour US Space Shuttle maiden flight suffered a software failure in its Intelsat satellite rendezvous maneuver and the Columbia US Space Shuttle maiden flight had to be postponed because of a clock synchronization algorithm failure.

Use of standard interval analysis could presumably have detected the roundoff difficulty in the first example. The extended interval arithmetic discussed in Chapter 4 and used in this book would have produced a correct interval result in the second example, even in the presence of overflow. See Walster (2003b) for an extended interval arithmetic implementation standard in which underflow and overflow are respectively distinguished from zero and infinity. The third failure was traced to an input-dependent software error that was not detected in spite of extensive testing. Intervals can be used to perform exhaustive testing that is otherwise impractical. Finally, the fourth failure occurred after the algorithm in question had been subjected to a three year review process and formally *proved* to be correct. Unfortunately, the proof was flawed. Although it is impossible to know, we believe that all of these and similar errors would have been detected if interval rather than floating-point algorithms had been used.

are successful if the result of the Newton step is contained in the input box. See Proposition 11.15.5.

Again let $\mathbf{x}^{I'}$ denote the smallest box containing the output box(es) from the first phase of our minimization algorithm applied to a perturbed problem.

Let $\phi(\mathbf{t}, \mathbf{c})$ denote the John function given by (13.5.1) and discussed in Section 17.5. Recall that

$$\mathbf{t} = \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \\ \mathbf{v} \end{pmatrix}$$

where $\mathbf{u}$ and $\mathbf{v}$ are vectors of Lagrange multipliers. The variable $\mathbf{t}$ takes the place of the variable $\mathbf{x}$ used when discussing Corollary 17.6.2. Let $\mathbf{J}(\mathbf{t}, \mathbf{c})$ denote the Jacobian of $\mathbf{f}(\mathbf{t}, \mathbf{c})$ as a function of $\mathbf{t}$. We can use this Jacobian as described above to prove that any John point in $\mathbf{x}^{I'}$ is a global minimum for some $\mathbf{c} \in \mathbf{c}^{I}$. Proof is obtained only if the result of the Newton step is contained in the input box. See Proposition 11.15.5.

Note that a subroutine is available for evaluating the Jacobian (and multiplying by an approximate inverse of its center) when doing the first stage of the algorithm in Section 17.5. Therefore, only a small amount of coding is needed to test the hypothesis of Corollary 17.6.2.

17.7 FIRST NUMERICAL EXAMPLE

In this and the next two section, we discuss examples that illustrate our method for solving perturbed problems.

As a first example, we consider the unconstrained minimization problem with objective function

$$f(x_1, x_2) = 12x_1^2 - 6.3x_1^4 + \mathbf{c}x_1^6 + 6x_1x_2 + 6x_2^2.$$

For $c = 1$, this is (the negative of) the so-called three hump camel function. See (for example) Dixon and Szegö (1975).

Let the coefficient (i.e., parameter) c vary over the interval [0.9, 1]. For all c for which $0.945 < c \leq 1$, the global minimum is the single point

at the origin; and $f^* = 0$ at this point. For $c = 0.945$, there are three global minima. One is at the origin. The others are at $\pm(a, -a/2)$ where $a = (10/3)^{1/2}$.

For $c < 0.945$, there are two global minima at $\pm(b, -b/2)$ where

$$b = \left[\frac{4.2 + (17.64 - 14c)^{1/2}}{2c}\right]^{1/2}. \tag{17.7.1}$$

At these points, f takes the negative (minimal) value

$$f^* = \frac{22.05c - 18.522 + (3.5c - 4.41)(17.64 - 14c)^{1/2}}{c^2}. \tag{17.7.2}$$

The smallest value of f^* for $c \in [0.9, 1]$ occurs for $c = 0.9$ for which $f^* = -1.8589$, approximately.

Consider perturbing the problem continuously by letting c increase from an initial value of 0.9. Initially, there are two global solution points that move along (separate) straight lines in the x_1, x_2 plane until $c = 0.945$. As c passes through this value, the global minimum jumps to the origin and remains there for all $c \in [0.945, 1]$.

A traditional perturbation analysis can detect the changes in the global minimum as c increases slightly from $c = 0.9$ (say). However, an expansion about this point cannot reveal the nature of the discontinuous change in position of the global minimum as c varies at and near the value 0.945.

The algorithm in Section 12.14 solves this problem without difficulty. Unfortunately, if termination tolerances are small, the output consists of an undesirably large number of boxes.

For $\mathbf{c}^{\mathbf{I}} = [0.9, 1]$, the set $\mathbf{x}^*(\mathbf{c}^{\mathbf{I}})$ of solution points consists of three parts. One part is the origin. Another is the line segment joining the two points of the form $(y, -y/2)$ where $y = (10/3)^{1/2}$ at one endpoint and $y = \{[21 + (126)^{1/2}]/9\}^{1/2}$ at the other endpoint. The third part of the set is the reflection of this line segment in the origin.

The interval algorithm does not reveal the value of c at which the global point jumps discontinuously. However, it bounds the solution set for all $c \in [0.9, 1]$ as closely as prescribed by the tolerances.

We ran this problem with c replaced by [0.9, 1]. The initial box had components $X_1^{(0)} = X_2^{(0)} = [-2, 4]$. The function width tolerance ε_f was chosen to be large so that termination was determined by the box size tolerance ε_X. Thus, we chose $\varepsilon_f = 10^5$.

We chose ε_X to have the relatively large value 0.1 so that only a few boxes were needed to cover the solution set. As is generally desired for the first stage of the two-stage process described in Section 17.5, we obtained rather crude bounds on the solution set.

The data for this example were computed using an algorithm similar to but less sophisticated than the one in Section 12.14. The algorithm produced a set of 11 boxes covering the solution set. The smallest box covering one subset of the exact solution is

$$\begin{bmatrix} [1.825, 1.893] \\ 0.9128, 0.9462] \end{bmatrix}. \tag{17.7.3}$$

This solution set was covered by five "solution" boxes. The smallest box containing these five boxes is

$$\begin{bmatrix} [1.739, 1.896] \\ 0.856, 0.968] \end{bmatrix}.$$

The interval components of this box are too large by roughly the size of the tolerance ε_X.

The reflection (in the origin) of the exact solution set (17.7.3) is also a solution set. It was covered by five "solution" boxes in much the same way.

The third subset of the exact solution is the origin. It remains an isolated solution point as c varies. It is a global solution for all $c \in [0.945.1]$. Its location was computed precisely and is guaranteed to be exact because the bounding interval components are degenerate.

The bounds on the set of values of f^* were computed to be the interval $[-4.781, 0]$. The correct interval is $[-1.859, 0]$. Since we chose $\varepsilon_f = 10^5$, the algorithm was not configured to produce good bounds on the interval $f^*(\mathbf{c}^{\mathbf{I}})$. The more stringent box size tolerance $\varepsilon_X = 0.1$ kept the bounds on

$f^*(\mathbf{c}^{\mathbf{I}})$ from being even worse. When we choose ε_X smaller, we incidentally bound $f^*(\mathbf{c}^{\mathbf{I}})$ more sharply.

With a smaller value of ε_X, the algorithm produces a larger number of smaller boxes covering the solution set more accurately and produces a narrower bound on $f^*(\mathbf{c}^{\mathbf{I}})$.

Note that even with the loose tolerances used, the algorithm correctly computed three disjoint sets covering the three correct solution sets.

Since the set of output boxes formed strictly disjoint subsets, we must allow for the possibility that the global solution points move discontinuously as c varies. This implies that the John points in the output boxes can correspond to local (nonglobal) minima for some values of $c \in \mathbf{c}^{\mathbf{I}}$. For this example, we know that this is, in fact, the case. Generally, however, we do not know the nature of such a solution.

If we apply the method of Section 17.5 to bound $f^*(\mathbf{c}^{\mathbf{I}})$ for each of the three "solution" regions separately, we obtain the approximate bounding interval $[-1.8589, 5.4359]$. Outwardly rounded to five decimal digits, the correct interval is $[-1.8589, 0]$. To compute a reasonably accurate result, ε_X must be considerably smaller than the 0.1 value used above. Alternatively, ε_f can be chosen smaller.

17.8 SECOND NUMERICAL EXAMPLE

We now consider a second example. It adds little to our understanding of the subject of this chapter. However, it is another easily analyzed example with which to research global optimization algorithms.

Our example is an inequality constrained minimization problem. The objective function is

$$f(x_1, x_2) = 12x_1^2 - 6.3x_1^4 + cx_1^6 + 6x_1x_2 + 6x_2^2$$

as in Section 17.7. We impose the constraints

$$p_1(\mathbf{x}) = 1 - 16x_1^2 - 25x_2^2 \leq 0,$$

$$p_2(\mathbf{x}) = 13x_1^3 - 145x_1 + 85x_2 - 400 \leq 0,$$

$$p_3(\mathbf{x}) = x_1x_2 - 4 \leq 0.$$

As in Section 17.7, we replace c by the interval [0.9, 1].

For this problem, the position of the global minimum again jumps discontinuously as c varies. The jump occurs at $c = c'$ where $c' = 0.95044391$, approximately.

For $c' \leq c < 1$, the global minimum occurs at two points on the boundary of the feasible region where $p_1(\mathbf{x}) = 0$. For $c = c'$, these two points are still global, and there are two other global minima in the interior of the feasible region. For $0.9 \leq c < c'$, only the two minima in the interior are global.

The minima in the interior are at $\pm(b, -b/2)$ where b is given by (17.7.1). The value of the objective function at these points is given by (17.7.2).

It can be shown that the minima on the boundary of the feasible region where $p_1(\mathbf{x}) = 0$ are at $\pm(x_1, x_2)$ where x_1 satisfies

$$(81.6x_1 - 126x_1^3 + 30cx_1^5)(1 - 16x_1^2)^{1/2} - 6 + 192x_1^2 = 0$$

and

$$x_2 = -(1 - 16x_1^2)/5.$$

The value of the minimum at these points depends very little on c. For $c = 0.9$, the global minimum is $f^* = 0.199035280$ and for $c = 1$, $f^* = 0.199035288$ approximately. The value of x_1 for $c = c'$ is 0.06604161 and for $c = 1$ it is 0.066041626 approximately.

The smallest boxes containing the set of points of global minimum as c varies over [0.9, 1] are (when outwardly rounded)

$$\pm\begin{bmatrix} [0.0660416, 0.0660417] \\ [-0.192896, -0.192895] \end{bmatrix}$$

and

$$\pm\begin{bmatrix} [1.81787, 1.89224] \\ [-0.946118, -0.908935] \end{bmatrix}.$$

We solved this problem by an algorithm that differs somewhat from the one given in Section 14.8. For reasons given in Section 17.3, we want ε_f to be large. We chose $\varepsilon_f = 10^5$. We again regard the computations to be the first stage of the two stage process given in Section 17.5. For such a computation, we want ε_X to be relatively large. We chose $\varepsilon_X = 0.05$.

The algorithm produced a set of 92 boxes as the solution. One isolated box is approximately

$$\begin{bmatrix} [0.0625, 0.0750] \\ [-0.2040, -0.1779] \end{bmatrix}.$$

Another isolated box is approximately the negative of this one. They contain the minima on the boundary of the feasible region.

One set of 76 contiguous boxes is isolated from all the other output boxes. The smallest box containing all of them, when outwardly rounded is

$$\mathbf{y}^{\mathbf{I}} = \begin{bmatrix} [1.7472, 1.8923] \\ [-0.9611, -0.8679] \end{bmatrix}.$$

The remaining set of 14 boxes is contained in a single box that is approximately the negative of this one.

Because of the loose tolerances used, the results do not bound the solution sets very tightly. The first output box above bounds a solution that is insensitive to perturbation of c. Therefore the computed result can be greatly improved by making the tolerance smaller.

Let $\mathbf{y}^{\mathbf{I}}$ denote the box containing the 76 contiguous output boxes as given above. This box bounds a solution that is more sensitive to c's perturbation. The width of $\mathbf{y}^{\mathbf{I}}$ is 0.1451, while the correct solution can be bounded by a box of width 0.0743. The individual boxes from which $\mathbf{y}^{\mathbf{I}}$ is determined satisfy the convergence criteria and hence are each of width ≤ 0.05. However, the size of the box covering their union is determined by the problem itself and cannot be changed by choosing a tolerance.

Since the output consists of strictly disjoint subsets, we cannot expect Assumption 17.5.1 of Section 17.5 to be satisfied. As in the example in Section 17.7, we must either be satisfied with poor bounds on $f^*(\mathbf{c}^{\mathbf{I}})$ and

$\mathbf{x}^{I*}(\mathbf{c}^I)$ or else continue with the first phase of our algorithm using a smaller box size tolerance.

17.9 THIRD NUMERICAL EXAMPLE

We now consider an example of a perturbed problem that arose in practice as a chemical mixing problem. It is an equality constrained least squares problem given by

$$\text{Minimize (globally) } f(\mathbf{x}) = \sum_{i=1}^{18} (x_i - c_i)^2$$

$$\begin{aligned}
\text{subject to } x_1x_9 &= x_2x_{14} + x_3x_4 \\
x_1x_{10} &= x_2x_{15} + x_3x_5 \\
x_1x_{11} &= x_2x_{16} + x_3x_6 \\
x_1x_{12} &= x_2x_{17} + x_3x_7 \\
x_1x_{13} &= x_2x_{18} + x_3x_8 \\
x_4 + x_5 + x_6 + x_7 + x_8 &= 1 \\
x_9 + x_{10} + x_{11} + x_{12} + x_{13} &= 1 \\
x_{14} + x_{15} + x_{16} + x_{17} + x_{18} &= 1 \\
x_{14} &= 66.67x_4 \\
x_{15} &= 50x_5 \\
x_{16} &= 0.015x_6 \\
x_{17} &= 100x_7 \\
x_{18} &= 33.33x_8.
\end{aligned}$$

The parameters c_i and their uncertainties ε_i are given in the following table.

i	$c_i \pm \varepsilon_i$	i	$c_i \pm \varepsilon_i$
1	100 ± 1.11	10	0.66 ± 0.017
2	89.73 ± 1.03	11	0.114 ± 0.0046
3	10.27 ± 0.51	12	0.002 ± 0.0001
4	0.0037 ± 0.00018	13	0.004 ± 0.00012
5	0.0147 ± 0.0061	14	0.245 ± 0.0067
6	0.982 ± 0.032	15	0.734 ± 0.02
7	0 ± 0	16	0.0147 ± 0.0061
8	0.0001 ± 0	17	0.0022 ± 0.0001
9	0.22 ± 0.0066	18	0.0044 ± 0.0013

We used the constraints to eliminate variables and write the problem as an unconstrained problem in five variables. We first solved the unperturbed case. The minimum value of f was found to be $f^* = 3.07354796 \times 10^{-7} \pm 2 \times 10^{-15}$. We then solved the perturbed case. and obtained a set of 59 boxes covering the solution set. The smallest single box (call it $\mathbf{x}^{\mathrm{I}'}$) containing the 59 boxes is given in the following table. We obtained the interval $[0, 2.556]$ bounding $f^*(\mathbf{c}^{\mathrm{I}})$.

i	X_i'	i	X_i'
1	[96.2, 103.8]	10	[0.6046, 0.7207]
2	[87.7, 91.7]	11	[0.0603, 0.1638]
3	[8.47, 12.07]	12	[−0.0174, 0.0237]
4	[0.0035, 0.00348]	13	[−0.0109, 0.0205]
5	[0.0142, 0.0152]	14	[0.2338, 0.2559]
6	[0.98142, 0.98147]	15	[0.7122, 0.7556]
7	[−0.000205, 0.000249]	16	[0.0147, 0.0148]
8	[−0.000384, 0.000645]	17	[−0.0205, 0.0249]
9	[0.1983, 0.2440]	18	[−0.0128, 0.0215]

As pointed out above, we used the equality constraints to eliminate variables and obtain an unconstrained optimization problem. Therefore, the Jacobian for the function $\phi(\mathbf{x}, \mathbf{c})$ expressing the John conditions (see Section 17.6) is just the Hessian of the objective function. As described in

Section 17.6, we verified that the Hessian (with $\mathbf{c}$ replaced by $\mathbf{c}^{\mathbf{I}}$) does not contain a singular matrix. Therefore, we know that the method described in Section 17.5 yields sharp bounds on $f^*(\mathbf{c}^{\mathbf{I}})$ and $\mathbf{x}^{\mathbf{I}*}(\mathbf{c}^{\mathbf{I}})$. We did not do the computations.

17.10 AN UPPER BOUND

In Section 12.5, and elsewhere, we discuss how we can use an upper bound $\bar{f}$ on the global minimum to improve the performance of our global optimization algorithm. In this section, we consider an artifice in which we preset $\bar{f}$ to zero in certain examples. We then give an example that shows why this is particularly helpful in the perturbed case.

For simplicity, we consider the unconstrained case. Suppose we wish to minimize $f(\mathbf{x}, \mathbf{c}^{\mathbf{I}})$ where $\mathbf{c}^{\mathbf{I}}$ is a nondegenerate interval vector. We apply the algorithm given in Section 12.14.

Assume we know that $f(\mathbf{x}, \mathbf{c})$ is nonnegative for all values of $\mathbf{x}$ and $\mathbf{c}$ of interest. Suppose we also know that $f(\mathbf{x}^*, \mathbf{c}) = \mathbf{0}$ for all $\mathbf{c} \in \mathbf{c}^{\mathbf{I}}$, where $\mathbf{x}^*$ is the point of global minimum. Then we know that $f^*(\mathbf{c}^{\mathbf{I}}) = \mathbf{0}$. Therefore, we set $\bar{f} = 0$.

Least squares problems are sometimes of this type of problem. It can be known that the squared functions are consistent and, hence, that $f^*(\mathbf{c}^{\mathbf{I}}) = 0$. For example, see Walster (1988).

As described in Section 12.5, our algorithm deletes points of an original box in which $f(\mathbf{x}, \mathbf{c}^{\mathbf{I}}) > \bar{f}$. The smaller $\bar{f}$, the more points that can be deleted in a given application of the procedure. Thus, it is advantageous to know $f^*(\mathbf{c}^{\mathbf{I}})$ when the algorithm is first applied. Often, the first value of $\bar{f}$ computed by the algorithm is much larger than $f^*(\mathbf{c}^{\mathbf{I}})$. Values of $\bar{f}$ closer to $f^*(\mathbf{c}^{\mathbf{I}})$ are computed as the algorithm proceeds.

We know $f^*(\mathbf{c}^{\mathbf{I}}) = 0$. In addition to speeding up the algorithm, this also saves the effort of repeatedly trying to improve $\bar{f}$. But, in the perturbed case, there is another advantage. When we evaluate $f(\mathbf{x}, \mathbf{c}^{\mathbf{I}})$ at some point $\mathbf{x}$, we obtain an interval. The upper endpoint of this interval is an upper bound for $f^*(\mathbf{c}^{\mathbf{I}})$. But even if we evaluate $f(\mathbf{x}, \mathbf{c}^{\mathbf{I}})$ at a global minimum point $\mathbf{x}^*$, the upper endpoint of the interval is not f^*. It is larger because

the uncertainty imposed by the interval $\mathbf{c}^I$ precludes the interval $f(\mathbf{x}^*, \mathbf{c}^I)$ from being degenerate.

Therefore, we can never compute an upper bound $\bar{f}$ equal to $f^*(\mathbf{c}^I)$ by evaluating $f(\mathbf{x}, \mathbf{c}^I)$. Knowing $f^* = 0$ and setting $\bar{f} = 0$ provides an advantage not otherwise obtainable.

We now consider an example. In Section 17.1, we pointed out that the problem of solving the set of equations $\mathbf{A}^I\mathbf{x} = \mathbf{b}^I$ can be recast as the least squares problem of minimizing

$$f(\mathbf{x}, \mathbf{c}^I) = (\mathbf{A}^I\mathbf{x} - \mathbf{b}^I)^T(\mathbf{A}^I\mathbf{x} - \mathbf{b}^I).$$

Here, the parameter vector $\mathbf{c}^I$ is composed of the elements of $\mathbf{A}^I$ and the components of $\mathbf{b}^I$.

Consider the system given in Equation (5.3.1). The solution set is shown in Figure 5.3.1. The smallest box containing the solution set is

$$\begin{bmatrix} [-120, 90] \\ [-60, 240] \end{bmatrix}.$$

When we evaluate $f(\mathbf{x}, \mathbf{c}^I)$ at some point $\mathbf{x}$, we obtain an interval $[\underline{f}(\mathbf{x}, \mathbf{c}^I), \overline{f}(\mathbf{x}, \mathbf{c}^I)]$. It can be shown that the smallest value of $\overline{f}(\mathbf{x}, \mathbf{c}^I)$ for any $\mathbf{x}$ is $2862900/121$, which is approximately 23660.33. Suppose we compute $\bar{f}$ equal to this best possible value we can compute.

Suppose we then delete all points $\mathbf{x}$ where $\underline{f}(\mathbf{x}, \mathbf{c}^I) > \bar{f}$. The smallest box that contains the remaining points can be shown to be

$$\begin{bmatrix} [-291.98, 243.82] \\ [-277.53, 457.53] \end{bmatrix}.$$

If we rely only on the procedure that uses $\bar{f}$ to delete points, the computed solution is much larger than it is possible to compute by including other procedures. The other procedures in the optimization algorithm delete whatever remaining points they can that are outside the solution set.

If we set $\bar{f} = 0$, then deleting points where $\underline{f}(\mathbf{x}, \mathbf{c}^I) > \bar{f}$ can delete all points not in the solution set. That is, the subroutine using $\bar{f}$ can contribute to the progress of the algorithm as long as points remain that are not in the solution set.

17.11 SHARP BOUNDS FOR PERTURBED SYSTEMS OF NONLINEAR EQUATIONS

We discussed perturbed systems of nonlinear equations of one variable in Section 9.10 and the multivariable case in Section 11.17. In this section, we discuss how such problems can be recast as optimization problems. We can then use the methods discussed earlier in this chapter to compute sharp bounds on the smallest box containing the solution set.

Consider a perturbed problem

$$\mathbf{f}(\mathbf{x}, \mathbf{c}^{\mathbf{I}}) = \mathbf{0} \tag{17.11.1}$$

where $\mathbf{f}$ is a vector of nonlinear functions and $\mathbf{x}$ is a vector of the same number of dimensions. The interval $\mathbf{c}^{\mathbf{I}}$ can be a scalar or vector. We replace this problem by

$$\text{Minimize } f(\mathbf{x}, \mathbf{c}^{\mathbf{I}}) \tag{17.11.2}$$

where

$$f(\mathbf{x}, \mathbf{c}^{\mathbf{I}}) = [\mathbf{f}(\mathbf{x}, \mathbf{c}^{\mathbf{I}})]^T \mathbf{f}(\mathbf{x}, \mathbf{c}^{\mathbf{I}}).$$

Pintér (1990) suggests solving unperturbed systems of nonlinear equations by recasting them as global optimization problems. He does not use interval methods. He considers more general norms than least squares.

We can apply the method described in Section 17.5 to solve (17.11.2) and thus compute sharp bounds on the smallest box containing the set of solutions of $\mathbf{f}(\mathbf{x}, \mathbf{c}^{\mathbf{I}}) = \mathbf{0}$.

It is reasonable to assume that (17.11.1) has a solution for all $\mathbf{c} \in \mathbf{c}^{\mathbf{I}}$. However, we need assume only that there exists at least one $\mathbf{c} \in \mathbf{c}^{\mathbf{I}}$ for which a solution exists. Under this assumption, the globally minimum value f^* of f is zero.

In Section 12.5, we discussed how an upper bound $\bar{f}$ on the global minimum f^* can be used in our optimization algorithm. Since we know that $f^* = 0$, we set $\bar{f} = 0$. As pointed out in Section 17.10, this improves the performance of our algorithm.

REFERENCES

1. Alefeld, G. (1968). Intervallrechnung über den Komplexen Zahlen und einige Anwendungen, doctoral dissertation, University of Karlsruhe.

2. Alefeld, G., (1977), Das symmetrische Einzelshrittverfahren bei linearen Gleichungen mit Intervallen als Koeffizienten, Computing, 18, 329–340.

3. Alefeld, G. (1979). Intervallanalytische Methoden bei nichtlinearen Gleichungen, in *Jahrbuch Überlicke Mathematik 1979*, S.D. Chatterji et al. (eds.) Bibliographisches Institut, Mannheim, pp. 63–78.

4. Alefeld, G. (1981). Bounding the slope of polynomial operators and some applications, Computing 26, 227–237.

5. Alefeld, G. (1984). On the convergence of some interval arithmetic modifications of Newton's method, SIAM J. Numer. Anal. 21, 363–372.

6. Alefeld, G. (1999). Interval arithmetic tools for range approximation and inclusion of zeros, in Bulgar and Zenger (1999), pp. 1–21.

7. Alefeld, G. and Herzberger, J. (1970). ALGOL-60 Algorithmen zur Auflösung linearer Gleichungs-systeme mit Fehlerfassung, Computing 6,28–34.

8. Alefeld, G. and Herzberger, J. (1974). *Einführung in die Intervallrechnung,* Bibliographisches Institut, Mannheim.

9. Alefeld, G. and Herzberger, J. (1983). *Introduction to Interval Computations,* Academic Press, Orlando, Florida

10. Alefeld, G. and Rokne, J. (1984). On improving approximate triangular factorizations iteratively, Beitr. Numer. Math. 12, 7–20.

11. Bao, P. G. and Rokne, J. G. (1988). Low complexity k-dimensional Taylor forms, Appl. Math. Comput. 27, 265–280.

12. Bauer, W. and Strassen, V. (1983). The complexity of partial derivatives, Theoret. Comput. Sci. 22, 317–330.

13. Bazaraa, M. S. and Shetty, C. M. (1979). *Nonlinear Programming. Theory and Practice,* Wiley, New York.

14. Bliek, C. (1992). Computer methods for design automation, Ph. D. thesis, Dept. of Ocean Engineering, Massachusetts Inst. of Tech.

15. Bliek, C., Jermann, C., and Neumaier, A. (eds.) (2003). *Global optimization and Constraint Satisfaction: First International Workshop Global Constraint Optimization and Constraint Satisfaction, COCOS 2002 Valbonne-Sophia Antipolis, France, October 2–4, 2002.* Volume number: LNCS 2861.

16. Boche, R. (1966). Complex interval arithmetic with some applications, Lockheed Missiles and Space Co. Report 4-22-66-1.

17. Boggs, P. T., Byrd, R. H., and Schnabel, R. B. (eds.) (1985). *Numerical Optimization* 1984, SIAM Publications.

18. Broyden, C. G. (1971), The convergence of an algorithm for solving sparse nonlinear systems, Math. Comp., 25, 285–294.

19. Bulgar, H. and Zenger, C. (eds.). *Error Control and Adaptivity in Scientific Computing,* Kluwer, Netherlands.

20. Burke, J. (1990). On the identification of active constraints II: The nonconvex case, SIAM J. Numer Anal. 27, 1081–1102.

21. Caprani, O. and Madsen, K. (1980). Mean value forms in interval analysis, Computing 25, 147–154.

22. Casado, L, Garcia, I., and Sergeyev, Y. (2000). Interval branch and bound algorithm for finding the first-zero-crossing-point in one-dimensional functions, Reliable Computing 6, 179–191.

23. Collavizza, H., Delobel, F., and Rueher, M. (1999). Comparing partial consistencies, Reliable Computing, 5, 213–228.

24. Corliss, G. F. (1995). Personal Communication.

25. Corliss, G. F. and Rall, L. B., Bounding Derivative Ranges, in *Encyclopedia of Optimization*, Panos, M. P. and Floudas, C. A. (eds.), Kluwer, Dordrecht, (to appear).

26. Dantzig, G. and Eaves, B. C. (1975). Fourier-Motzkin elimination and its dual with applications to integer programming, in *Combinatorial Programming: Methods and Application,* Proceedings of the NATO Advanced Study Institute, B. Roy (ed.), Reidel, Dordrecht, Netherlands.

27. Deif, A. (1986). *Sensitivity Analysis in Linear Systems,* Springer-Verlag, Berlin.

28. Dennis, J. E., Jr. and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Prentice-Hall, Englewood Cliffs, New Jersey.

29. Dinkel, J. J. and Tretter, M. J. (1987). An interval arithmetic approach to sensitivity analysis in geometric programming, Oper. Res. 35, 859–866.

30. Dinkel, J. J. Tretter, M. J., and Wong, D. (1988). Interval Newton methods and perturbed problems, Appl. Math. Comput. 28, 211–222.

31. Dixon, L. C. W., and Szegö, G. P. (1975). *Towards Global* Optimization, North Holland/American Elsevier, New York.

32. Daumas, M. (2002). Past and future formalizations of the IEEE 754, 854 and 754R standards, Talk presented to the IEEE 754 committee. July 18, 2002 at Cupertino, CA. (See: `grouper.ieee.org/groups/754/meeting-materials/2002-07-18-daumas.pdf`)

33. Dwyer, P. S. (1951). Computation with approximate numbers, in *Linear Computations,* P. S. Dwyer (ed.), Wiley, New York, pp. 11–34.

34. Floudas C. A., and Pardalos, P. M. (eds.) (1992). *Recent Advances in Global Optimization*, Princeton University Press, Princeton, New Jersey.

35. Forte^{TM} Developer 6 Fortran 95. @ `www.sun.com/forte/` by Sun Microsystems Inc., July, 2000.

36. Forte^{TM} Developer 6 Update 1 C++. @ `www.sun.com/forte/` by Sun Microsystems Inc., October, 2000.

37. Forte^{TM} Developer 6 Update 1 Fortran 95. @ `www.sun.com/forte/` by Sun Microsystems Inc., October, 2000.

38. Frommer, A. and Mayer, G. (1990). On the R-order of Newton-like methods for enclosing solutions of nonlinear equations, SIAM J. Numer. Anal. 27,105–116.

39. Gargantini, I. (1975a). Parallel square root iterations, in Nickel (1975), pp. 195–204.

40. Gargantini, I. (1975b). Parallel Laguerre iterations, Numer. Math. 26, 317–323.

41. Gargantini, I. (1978). Further applications of circular arithmetic: Schroeder-like algorithms with error bounds for finding zeros of polynomials, SIAM J. Numer. Anal. 15, 497–510.

42. Gargantini, I. and Henrici, P. (1972). Circular arithmetic and the determination of polynomial zeros, Numer. Math. 18, 305–320.

43. Garloff, J. and Smith, A. P. (2000), Investigation of a subdivision based algorithm for solving systems of polynomial equations, Proc. of the 3rd World Congress of Nonlinear Analysis (WCNA 2000), Catania, Italy.

44. Glatz, G. (1975). Newton-Algorithmen zur Bestimmung von Polynomwurzeln unter Verwendung komplexer Kreisarithmetic, in Nickel (1975), pp. 205–214.

45. Griewank, A. (1989). On automatic differentiation, in *Mathematical Programming 88*, Kluwer Academic Publishers, Boston.

46. Griewank, A. (1991). The chain rule revisited in scientific computing, SIAM News, May.

47. Griewank, A. (2000). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation,* SIAM Publ., Philadelphia.

48. Gustafson, J. L. (1994a). A Paradigm For Grand Challenge Performance Evaluation, (HTML version: `www.scl.ameslab.gov/Publications/Grand Challenge/Paradigm.html`), (PDF version: `www.scl.ameslab.gov/Publications/Paradigm.pdf`) *Proceedings of the Toward Teraflop Computing and New Grand Challenge Applications Mardi Gras '94 Conference*, Baton Rouge, Louisiana, February 1994.

49. Gustafson, J. L. (1994b). Teraflops and other false goals, *Parallel and Distributed Technology*, Summer 1994.

50. Gustafson, J. L. (1995). HINT—A New Way To Measure Computer Performance, (HTML version: `www.scl.ameslab.gov/Publications/HINT/ComputerPerformance.html`) (PDF version: `www.scl.ameslab.gov/Publications/HINT/HINT Acrobat.pdf`), in Gustafson and Snell (1995).

51. Gustafson, J. L. and Snell, Q. O. (1995). Proceedings of the HICSS-28 Conference, Wailela, Maui, Hawaii, January 3–6, 1995.

52. Gustafson, J. L. (1998). Computational verifiability of the ASCI program, *IEEE Computational Science and Engineering*, March-June 1998.

53. Hansen, E. R. (1965). Interval arithmetic in matrix computations, part I, SIAM J. Numer. Anal. 2, 308–320.

54. Hansen, E. R. (1968). On solving systems of equations using interval arithmetic, Math. Comput. 22, 374–384.

55. Hansen, E. R. (1969a). *Topics in Interval Analysis*, Oxford University Press, London.

56. Hansen, E. R. (1969b). On linear algebraic equations with interval coefficients, in Hansen (1969a), pp. 35–46.

57. Hansen, E. R. (1969c). On the centered form, in Hansen (1969a), pp. 102–106.

58. Hansen, E. R. (1975). A generalized interval arithmetic, in Nickel (1975), pp.7–18.

59. Hansen, E. R. (1978a). Interval forms of Newton's method, Computing 20, 153–163.

60. Hansen, E. R. (1978b). A globally convergent interval method for computing and bounding real roots, BIT 18, 415–424.

61. Hansen, E. R. (1979). Global optimization using interval analysis-the one dimensional case, J. Optim. Theory Appl. 29, 331–344.

62. Hansen, E. R. (1980). Global optimization using interval analysis-the multi-dimensional case, Numer. Math. 34, 247–270.

63. Hansen, E. R. (1984). Global optimization with data perturbations, Comput. Ops. Res. 11, 97–104.

64. Hansen, E. R. (1988). An overview of global optimization using interval analysis, in Moore (1988), pp. 289–307.

65. Hansen, E. R. (1991). Bounding the set of solutions of a perturbed global optimization problem, in *Proceedings of the Second Workshop on Global Optimization*, International Institute for Applied Systems Analysis.

66. Hansen, E. R. (1992). Bounding the solution of interval linear equations, SIAM J. Numer. Anal., 29, 1493–1503.

67. Hansen, E. R. (1993). Computing zeros of functions using generalized interval arithmetic, Interval Computations, 3, 3–28.

68. Hansen, E. R. (1997a). Preconditioning linearized equations, Computing, 58, 187–196.

69. Hansen, E. R. (1997b). Sharpness in interval computations, Reliable Computing, 3,17–29.

70. Hansen, E. R. (2000). The hull of preconditioned interval linear equations, Reliable Computing, 6, 95–103.

71. Hansen, E. R. (2002). Sharp solutions for interval linear equations (submitted).

72. Hansen, E. R. and Greenberg, R. I. (1983). An interval Newton method, appl. Math. Comput. 12, 89–98.

73. Hansen, E. R. and Sengupta, S. (1980). Global constrained optimization using interval analysis, in Nickel (1980), pp. 25–47.

74. Hansen, E. R. and Sengupta S. (1981). Bounding solutions of systems of equations using interval analysis, BIT 21, 203–211.

75. Hansen, E. R. and Sengupta, S. (1983). Summary and steps of a global nonlinear constrained optimization algorithm, Lockheed Missiles and Space Co. report no. D889778.

76. Hansen, E. and Smith, R. (1967). Interval arithmetic in matrix computations, Part 2, SIAM J. Numer. Anal., 4, 1–9.

77. Hansen, E. R. and Walster, G. W. (1982). Global optimization in nonlinear mixed integer problems, in *Proc. 10th IMACS World Conference on System Simulation and Scientific Computation*, vol I, pp. 397–381.

78. Hansen, E. R. and Walster, G. W. (1992a). Bounds for Lagrange multipliers and optimal points, in the Second Special Issue on Global Optimization, Control, and Games, Comput. Math. Appl., pp. 59–69.

79. Hansen, E. R. and Walster, G. W. (1992b). Nonlinear equations and optimization, in the Second Special Issue on Global Optimization, Control, and Games, Comput. Math. Appl.

80. Hansen, E. R. and Walster, G. W. (1992c). Equality constrained global optimization, submitted to SIAM Journal On Control and Optimization. Note: The authors finished this paper and submitted it for publication in 1987. It was to be published, but the authors have no record of this fact. The paper will be resubmitted. In the meantime, a preprint can be found at: `www.cs.utep.edu/interval-comp/EqualityConstraints.pdf`

81. Hansen, E. R. and Walster, G. W. (2002). Sharp bounds on interval polynomial roots. Reliable Computing, 8–2, 115–112.

82. Hansen, E. R. and Walster, G. W. (2003). Solving overdetermined systems of linear equations. Submitted to Reliable Computing.

83. Hanson, R. J. (1968). Interval arithmetic as a closed arithmetic system on a computer, Jet Propulsion Laboratory report 197.

84. Hartfiel, D. J. (1980). Concerning the solution of $Ax = b$ when $P \leq A \leq Q$ and $p \leq b \leq q$, Numer. Math. 35, 355–359.

85. Hebgen, M. (1974). Eine scaling Pivotsuche für Intervallmatrizen, Computing 12, 99–106.

86. Heindl, G., Kreinovich, V., and Lakeyev, A. (1998), Solving linear interval systems is NP-hard even if we exclude overflow and underflow, Reliable Computing, 4, 383–388.

87. Hennessy, J. L. and Patterson, D. A. (1994). *Computer Organization and Design*, Morgan Kaufmann, San Mateo, California.

88. Henrici, P. (1975). Einige Anwendungen der Kreisscheibenarithmetic in der Kettenbruchtheorie, in Nickel (1975), pp. 19–30.

89. Hickey, Q. J. and Van Emden M. H. (1999). Interval Arithmetic: From Principles to Implementation, Technical Report DCS-260-IR, Department of Computer Science, Victoria, B.C. Canada.

90. Hock, W. and Schittkowski, K. (1981). *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York.

91. IBM (1986a). IBM high accuracy arithmetic subroutine library (ACRITH). General information manual GC33-6163-02, third edition.

92. Ichida, K., and Fujii, Y. (1990). Multicriterion optimization using interval analysis, Computing 44, 47–57.

93. IEEE (1985). IEEE standard for binary floating-point arithmetic, ANSI/IEEE STD 754–1985 Technical Report, New York.

94. Iri, M. (1984). Simultaneous computation of functions, partial derivatives, and estimates of rounding errors-complexity and practicality, Jpn. J. Appl. Math. 1, 223–252.

95. Jacobs, D. (ed.) (1976). The state of the art in numerical analysis, in *Proc. Conference on the State of the Art in Numerical Analysis*, University of York.

96. Jaulin, L., Kieffer, M., Didrit, O., and Walter, É. (2001) *Applied Interval Analysis*, Springer-Verlag, London.

97. Kahan, W. M. (1968). A more complete interval arithmetic, Lecture notes for a summer course at the University of Michigan.

98. Kearfott, R. (1992). An interval branch and bound algorithm for bound constrained optimization problems, J. Global Optimiz., 2, 259–280.

99. Kearfott, R. (1996). *Rigorous Global Search: Continuous Problems,* Kluwer Academic Publ., Dordrecht.

100. Kearfott, R. B. and Dian, J. (2000). Existence verification for higher-degree singular zeros of complex nonlinear systems, preprint, `interval.louisiana.edu.cplx.0302.pdf`.

101. Kirjner, C. and Polak, E. (1998). On the conversion of optimization problems with max-min constraints to standard optimization problems, SIAM J. Optim., 8, 887–915.

102. Krawczyk, R. (1969). Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlershranken, Computing 4, 187–201.

103. Krawczyk, R. (1983). A remark about the convergence of interval sequences, Computing 31, 255–259.

104. Krawczyk, R. (1986). A class of interval Newton operators, Computing 37, 179–183.

105. Krawczyk, R., and Neumaier, A. (1985). Interval slopes for rational functions and associated centered forms, SIAM J. Numer. Anal. 22, 604–616.

106. Kreier, N. (1974). Komplexe Kreisarithmetic, Z. Angew. Math. Mech. 54, 225–226.

107. Kreier, N., and Spellucci, P. (1975). Einschliessungsmengen von Polynom-Nullstellen, in Nickel (1975), pp.223–229.

108. Kreinovich, V., Lakeyev, A., Rohn, J., and Kahl, P., *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1998.

109. Kulisch, U. (ed.) (1987). *Computer Arithmetic, Scientific Computation and Programming Languages*, Teubner, Stuttgart.

110. Kulisch, U., and Miranker, W. L. (eds.) (1983). *A New Approach to Scientific Computation*, Academic Press, Orlando, Florida.

111. Lemaréchal, C. (1982). Nondifferentiable optimization, in Powell (1982), pp. 85–89.

112. Levy, A. V., and Gomez, S. (1985). The tunneling method applied to global optimization, in Boggs, Byrd, and Schnabel (1985), pp. 213–244.

113. Levy, A. V., Montalvo, A., Gomez, S., and Calderon, A. (1981). *Topics in Global Optimization*, Lecture Notes in Mathematics No.909, Springer-Verlag, New York.

114. Loh, E. and Walster, G. W. (2002). Rump's example revisited. Reliable Computing, 8 (2) 245–248.

115. Mancini, L. J., (1975). Applications of interval arithmetic in signomial programming, Technical report SOL 75–23, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

116. Mancini, L. J., and McCormick, G. P. (1976). Bounding global minima, Math. Oper. Res. 1, 50–53.

117. McAllester, D., Van Hentenryck, P., and Kapur, D. (1995), Three cuts for accelerated interval propagation, Massachusetts Inst. of Tech. Artificial Intelligence Lab. memo no. 1542.

118. Metropolis, N., et al. (eds.) (1980). *A History of Computing in the 20th Century*, Academic Press, New York.

119. Mohd, I. B., (1990). Unconstrained global optimization using strict complementary slackness, Appl. Math. Comput. 36, 75–87.

120. Moore, R. E. (1965). The automatic analysis and control of error in digital computation based on the use of interval numbers, in Rall (1965), pp. 61–130.

121. Moore, R. E. (1966).*Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.

122. Moore, R. E. (1977). A test for existence of solutions of nonlinear systems, SIAM J. Numer. Anal. 14, 611–615.

123. Moore, R. E. (1979).*Methods and Applications of Interval Analysis*, SIAM Publ., Philadelphia, Pennsylvania.

124. Moore, R. E. (1980). Microprogrammed interval arithmetic, ACM SIGNUM Newsletter. 15(2), p. 30.

125. Moore, R. E. (ed.) (1988).*Reliability in Computing*, Academic Press, San Diego.

126. Moore, R. E. (1991). Global optimization to prescribed accuracy, Comput Math. Appl. 21, 25–39.

127. Moore, R. E., Hansen, E. R,, and Leclerc, A. (1992). Rigorous methods for parallel global optimization, in Floudas and Pardalos (1992), pp.321–342.

128. Moré, J. J., Garbow, B. S., and Hillstrom, K. E. (1981). Testing unconstrained optimization software, ACM Trans. Math. Software 7, 17–41.

129. Nataraj, P. S. V. and Sardar, G. (2000). Computation of QFT bounds for robust sensitivity and gain-phase margin specifications, Trans. ASME Journal of Dynamical Systems, Measurement, and Control, Vol. 122, pp. 528–534.

130. Nataraj, P. S. V. (2002a). Computation of QFT bounds for robust tracking specifications, Automatica, Vol. 38, pp. 327–334.

143. Nickel, K. (1977). Die Überschätzung des Wertebereichs einer Function in der Intervallrechnumg mit Anwendungen auf lineare Gleichungssysteme, Computing, 18, 15–36.

144. Nickel, K. L. (ed.) (1980). *Interval Mathematics 1980*, Academic Press, New York.

145. Nickel, K. L. (ed.) (1986). *Interval Mathematics 1985*, Springer Lecture Notes in Computer Science No. 212, Springer-Verlag, New York.

146. Nickel, K. L., and Ritter, K. (1972). Termination criteria and numerical convergence, SIAM J, Numer. Anal. 9, 277–283.

147. Ning, S. and Kearfott, R. (1997). A comparison of some methods for solving linear interval equations, SIAM J. Numer. Anal., 34, 1289–1305.

148. Oettli, H. (1965). On the solution set of a linear system with inaccurate coefficients, SIAM J. Numer. Anal. 2, 115–118.

149. Oettli, H., Prager, H., and Wilkinson, J. H. (1965). Admissible solutions of linear systems with not sharply defined coefficients, SIAM J. Numer. Anal. 2, 291–299.

150. Palmer, J. F. and Morse, S. P. (1984). *The 8087 Primer*, Wiley Press, New York.

151. Pintér, J. (1990). Solving nonlinear equation systems via global partition and search: some experimental results, Computing 43, 309–323.

152. Powell, M. J. D. (ed.) (1982).*Nonlinear Optimization, 1981*, Academic Press, New York.

153. Qi. L. (1982). A note on the Moore test for nonlinear systems, SIAM J. Numer. Anal. 19, 851–857.

154. Rall, L. B. (1965). *Error in Digital Computation*, Vol. I, Wiley, New York.

155. Rall, L. B. (1969). *Computational Solution of Nonlinear Operator Equations*, Wiley, New York.

156. Rall, L. B. (1981).*Automatic Differentiation-Techniques and Applications*, Springer Lecture Notes in Computer Science, Vol. 120, Springer-Verlag, New York.

157. Rall, L. B. (1983). Differentiation and generation of Taylor coefficients in Pascal-SC, in Kulisch and Miranker (1983), pp. 291–309.

158. Rall, L. B. (1984). Differentiation in PASCAL-SC: type gradient, ACM Trans. Math. Software 10, 161–184.

159. Rall, L. B. (1987). Optimal implementation of differentiation arithmetic, in Kulisch (1987).

160. Ratschek, H., and Rokne, J. (1984). *Computer Methods for the Range of Functions*, Halstead Press, New York.

161. Ratschek, H., and Rokne, J. (1988). *New Computer Methods for Global Optimization*, Wiley, New York.

162. Ratz, D. (1994). Box splitting strategies for the interval Gauss-Seidel step in a global optimization method, Computing, 53, 337–353.

163. Ratz, D. (1998). *Automatic Slope Computation and Its Application In Nonsmooth Global Optimization*, Shaker Verlag, Aachen.

164. Ratz, D. (1999). A nonsmooth global optimization technique using slopes — the one-dimensional case. Jour. Global Optim. 14, 365–393

165. Reichmann, K. (1979). Abbruch beim Intervall-Gauss-Algorithmus, Computing 22, 355–362.

166. Rigal, J. L., and Gaches, J. (1967). On the compatibility of a given solution with the data of a linear system, J. Assoc. Comput. Mach. 14, 543–548.

167. Ris, F. N. (1972). Interval analysis and applications to linear algebra, Ph.D. dissertation, Oxford University.

168. Ris, F. N. (1975). Tools for the analysis of interval arithmetic, in Nickel (1975), pp. 75–98.

169. Robinson, S. M. (1973). Computable error bounds for nonlinear programming, Math. Programming 5, 235–242.

170. Rohn, J. (1993). Cheap and tight bounds: The recent result of E. Hansen can be made more efficient, Interval Computations, 4, 13–21.

171. Rokne, J. G. (1986). Low complexity k-dimensional centered forms, computing 37, 247–253.

172. Rokne, J. G., and Bao, P. (1987). Interval Taylor forms, Computing 39, 247–259.

173. Rump, S. M. (1988). Algorithm for verified inclusions-theory and practice, in Moore (1988), pp. 109–126.

174. Rump, S. M. (1990). Rigorous sensitivity analysis for systems of linear and nonlinear equations, Math. Comput. 54, 721–736.

175. Rump, S. M. (1996). Expansion and estimation of the range of nonlinear functions, Math. Comp., 65, 1503–1512.

176. Schwefel, H. (1981).*Numerical Optimization of Computer Models,* Wiley, new York.

177. Sengupta, S. (1981). Global nonlinear constrained optimization, Ph.D. dissertation, Washington State University.

178. Shary, S. (1995), On optimal solution of interval linear equations, SIAM J. Numer. Anal., 32, 610–630.

179. Shary, S. (1999). Outer estimation of generalized solution sets to interval linear systems, Reliable Computing, 5, 323–335.

180. Shearer, J. M., and Wolfe, M. A. (1985a). some computable existence, uniqueness, and convergence tests for nonlinear systems, SIAM J. Numer. Anal. 22, 1200–1207.

181. Shearer, J. M., and Wolfe, M. A. (1985b). An improved form of the Krawczyk-Moore algorithm, Appl. Math. Comput. 17, 229–239.

182. Speelpenning, B. (1980). Compiling fast partial derivatives of functions given by algorithms, Ph.D. dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign.

183. Stewart, G. W. (1973). *Introduction to Matrix Computations*, Academic Press, New York.

184. Sunaga, T. (1958). Theory of interval algebra and its application to numerical analysis, RAAG Memoirs 3, 29–46.

185. Van Hentenryck, P., McAllester, D., and Kapur, D. (1997), Solving Polynomial systems using branch and prune approach, SIAM J. Numer. Anal., 34, 797–827.

186. Van Hentenryck, P., Michel, L., and Deville, Y. (1997), *Numerica. A Modeling Language for Global Optimization,* The MIT Press, Cambridge.

187. Walster, G. W. (1988). Philosophy and practicalities of interval arithmetic, pages 309–323 of Moore (1988).

188. Walster, G. W. (1996). Interval Arithmetic: The New Floating-Point Arithmetic Paradigm, Sun Solaris Technical Conference, June 1996.

189. Walster, G. W. (2000a). The "Simple" closed interval system, Technical report, Sun Microsystems Inc., February, 2000.

190. Walster, G. W. (2000b). Interval arithmetic programming reference: Forte™ workshop 6 Fortran 95. Sun Microsystems Inc., May 2000.

191. Walster, G. W. (2000c). Interval arithmetic programming reference: Forte™ workshop 6 update 1 Fortran 95. Sun Microsystems Inc., Nov., 2000.

192. Walster, G. W. (2002). Implementing the "Simple" Closed Interval System. See:
www.sun.com/software/sundev/whitepapers/simple-implementation.pdf

193. Walster, G. W. (2003a). Exterior interval continuity, to be submitted for publication.

194. Walster, G. W. (2003b). The containment set based interval arithmetic standard, to be submitted for publication.

195. Walster, G. W. and Chiriaev, D. (2000). Interval arithmetic programming reference: Forte™ workshop 6 update 1 C++. Sun Microsystems Inc., Nov, 2000.

196. Walster, G. W. and Hansen, E. R. (1997). Interval algebra, composite functions, and dependence in compilers, Technical Report, Sun Microsystems Inc.

197. Walster, G. W. and Hansen, E. R. (1998). Composite functions in interval mathematics. Preprint available at:
www.mscs.mu.edu/~globsol/readings.html#Walster
March, 1998.

198. Walster, G. W. and Hansen, E. R. (2003). Computing interval parameter bounds from fallible measurements using overdetermined (tall) systems of nonlinear equations, in Bliek, C., *et. al.* (2003)

199. Walster, G. W. and Hansen, E. R. (2004). Using pillow functions to efficiently compute crude range tests, in the SCAN 2000 Proceedings to appear in Num. Alg.

200. Walster, G. W., Hansen, E. R., and Pryce, J.D. (2000). Extended real intervals and the topological closure of extended real relations, Technical Report, Sun Microsystems Inc., February, 2000.

201. Walster, G. W., Hansen, E. R., and Sengupta, S. (1985). Test results for a global optimization algorithm, in Boggs, Byrd, and Schnabel (1985), pp. 272–287.

202. Walster, G. W. and Kreinovich, V. (2003). Computational complexity of optimization and crude range testing: a new approach motivated by fuzzy optimization, Fuzzy Sets and Systems, 9, 179–208.

203. Walster, G. W., Liddiard, L. and Tretter, M. J. (1980). INTLIB User Manual (unpublished).

204. Walster, G., Pryce, J.D., and Hansen, E. R. (2002). Practical, exception-free interval arithmetic on the extended reals. SIAM Journal on Scientific Computing. The paper was provisionally accepted for publication; has since been significantly revised; and will be resubmitted for publication.

205. Warmus, M. (1956). Calculus of approximations, Bull. de l'Academie Polonaise des Sciences, 4, 253–259.

206. Warmus, M. (1960). Approximations and inequalities in the calculus of approximations. Classification of approximate numbers, Bull. de l'Academie Polonaise des Sciences, 9, 241–245.

207. Watson, L. T. (1986). Numerical linear algebra aspects of globally convergent homotopy methods, SIAM Rev. 28, 529–545.

208. Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem*, Oxford University Press, London.

209. Wilkinson, J. H. (1980). Turing's work at the NPL and the Pilot ACE, DEUCE, and ACE, in Metropolis, et al. (1980).

210. Wolfe, M. A.(1999). On discrete minimax problems in R using interval arithmetic, Reliable Computing 5, 371–383.

211. Wolfe, P. (1982). Checking the calculation of gradients, ACM Trans. Math. Software 6, 337–343.

212. Wongwises, P. (1975a). Experimentelle Untersuchungen zur numerische Auflösungen von linearen Gleichungssystemen mit Fehlershranken, dissertation, Instituts für Praktische Mathematik, report 75/1, University of Karlsruhe.

213. Wongwises, P. (1975b). Experimentelle Untersuchungen zur numerische Auflösungen von linearen Gleichungssystemen mit Fehlerschranken, in Nickel (1975), pp. 316–325.

214. Yohe, J. M. (1979). Implementing nonstandard arithmetic, SIAM Rev. 21, 34–56.

215. Zuhe, S., and Wolfe, M. A. (1990). On interval enclosures using slope arithmetic, Appl. Math. Comput. 39, 89–105.