IMPACTS OF RECENT COMPUTER ADVANCES ON OPERATIONS RESEARCH

Editors:

Ramesh Sharda
College of Business Administration
Oklahoma State University
Stillwater, Oklahoma, U.S.A.

Bruce L. Golden
College of Business and Management
University of Maryland
College Park, Maryland, U.S.A.

Edward Wasil
Kogod College of Business Administration
The American University
Washington, D.C., U.S.A.

Osman Balci
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia, U.S.A.

William Stewart
School of Business
College of William and Mary
Williamsburg, Virginia, U.S.A.

NORTH-HOLLAND
New York • Amsterdam • London

This book is printed on acid-free paper.

Published by:
Elsevier Science Publishing Co., Inc.
655 Avenue of the Americas, New York, New York 10010

Sole distributors outside the United States and Canada:

Elsevier Science Publishers B.V.
P.O. Box 211, 1000 AE Amsterdam, The Netherlands

Library of Congress Cataloging in Publication Data

Impact of recent computer advances on operations research / editors, Ramesh Sharda...[et al.].
p. cm. — (Publications in operations research series; v. 9)
Includes index.
1. Operations research—Data processing. I. Sharda, Ramesh. II. Series.
T57.6.I483 1989
658.4'034—dc20 89-11653
CIP
ISBN 0-444-01492-6

Current Printing
10 9 8 7 6 5 4 3 2 1

Manufactured in the United States of America.

INTERVAL ARITHMETIC METHODS FOR NONLINEAR SYSTEMS AND NONLINEAR OPTIMIZATION: AN OUTLINE AND STATUS

R. BAKER KEARFOTT

Dept. of Mathematics, Univ. of Southwestern Louisiana U.S.L. Box 4-1010, Lafayette, Louisiana 70504

ABSTRACT

In operations research, we often try to to find the global optimum of a nonlinear function subject to constraints on the variables. Where applicable, methods of interval arithmetic can do this with *mathematical certainty*, despite the fact that the algorithms execute on conventional computers. This is because such methods perform a (seemingly contradictory) exhaustive but efficient search, and take account of all rounding errors through *directed roundings*.

We have transportable interval software to find all solutions to a nonlinear system of equations within a given region; minor modifications allow this software to solve the global optimization problem. The impact of this on operations research will be to enable certain such problems to be solved with *total reliability*.

Future development will increase the range of problems for which these methods are practical.

1. INTRODUCTION

An important problem in operations research is to find global optima of nonlinear functions subject to constraints. Recent progress in interval mathematics and interval software can impact operations research by providing solutions to the following specific instances of these problems.

Find, *with certainty*, approximations to all solutions of the nonlinear system

(1.1) $$f_i(x_1, x_2, ..., x_n) = 0, 1 \leq i \leq n,$$

where bounds l_i and u_i are known such that:

$$l_i \leq x_i \leq u_i for 1 \leq i \leq n;$$

and, related to (1.1),

find, *with certainty*, the global minimum of the nonlinear objective function

(1.2) $$\phi(x_1, x_2, ..., x_n),$$

where bounds l_i and u_i are known such that:

$$l_i \leq x_i \leq u_i \text{ for } 1 \leq i \leq n.$$

Impacts of Recent Computer Advances on Operations Research
Ramesh Sharda, Bruce L. Golden, Edward Wasil, Osman Balci, William Stewart, Editors

An interval algorithm will produce a list of solutions whose coordinates x_i are given as small intervals of uncertainty. If the proper algorithm (cf. Section 3 below) is correctly implemented with directed roundings (cf. Section 2 below), completion of this algorithm constitutes a computational but *mathematically rigorous proof* that all solutions of (1.1) are within the intervals given in the list.

We have transportable software which will in practice solve many small to moderately sized polynomial systems of the form (1.1) without computer programming, and which will solve more general systems if we program the objective function [Kearfott and Novoa 1988]. The only way such software can fail is by not completing within a given time. (Nonetheless, on some problems, it completes in much less time than alternate methods.)

In this paper, we review facts about interval arithmetic and a class of algorithms of use to members of the operations research community. Section 2 contains elementary definitions and information on computer implementations of interval arithmetic. In Section 3, we discuss interval algorithms for solving nonlinear algebraic systems and for nonlinear optimization. In Section 4, we summarize strengths and weaknesses of these algorithms.

2. ELEMENTARY FACTS AND COMPUTER IMPLEMENTATIONS

Thorough introductions to interval mathematics are given in the books [Alefeld et al. 1983; Moore 1979]. In particular, one can find details for this section in chapters 1, 2, and 3 of [Moore 1979] or in chapters 1, 2, 3, and 4 of [Alefeld et al. 1983]. Also see [Rall 1981] and [Ratschek 1987] if they are available.

Numerous references facilitate advanced study. The bibliographies [Garloff 1985; Garloff 1987] list approximately 2000 books, journal and conference proceedings articles, and technical reports. In [Moore 1988], general interval and non interval numerical methods are compared. Additional proceedings include [Hansen 1969; Miranker 1986; Nickel 1980; Nickel 1986a].

Here, interval quantities will be denoted by boldface. We will also denote the n-vector whose i-th component is x_i by X and the n-vector whose i-th component is f_i by $F(X)$.

Interval arithmetic is based on defining the four elementary arithmetic operations on intervals. Let $a = [a_l, a_u]$ and $b = [b_l, b_u]$ be intervals. Then, if *op* ∈ [+,-,*,/], we define

(2.1) $$a \ op \ b = [x + y \mid x \in a \ and \ y \in b]$$

For example,

$$a + b = [a_l + b_l, a_u + b_u].$$

In fact, all four operations can be defined in terms of addition, subtraction, multiplication, and division of the endpoints of the intervals, although multiplication and division may require comparison of several results. The result of these operations is an interval except when we compute a/b and $0 \in b$. In that case, we use *extended interval arithmetic* [Moore 1979, pp. 66-68] to get two semi-infinite intervals or else the whole real line.

Much of interval mathematics' power lies in the ability to compute *inclusion monotonic interval extensions* of functions. If f is a continuous function of a real variable, then an inclusion monotonic interval extension f is defined to be a function from the set of intervals to the set of intervals, such that, if x is an interval in the domain of f,

$$[\, f(x) \mid x \in x] \subset f(x)$$

and such that

$$x \subset y \text{ implies } f(x) \subset f(y).$$

Inclusion monotonic interval extensions of a polynomial may be obtained by simply replacing the dependent variables by intervals and replacing the additions and multiplications by the corresponding interval operations [Moore 1979].

We emphasize here that the result of an elementary interval operation is precisely the range of values that the usual result attains as we let the operands range over the two intervals. However, the value of an interval extension of a function is not precisely the range of the function over its interval operand, but only contains this range; it is an art to devise interval extensions whose values differ little from the actual range. For example, the values of an interval extension of a polynomial depend on the form in which the polynomial is written. See Kearfott [1988c] for an example, and see Ratschek and Rokne [1984] for a discussion of efficient ways of formulating interval extensions.

We may use the mean value theorem or Taylor's theorem with remainder formula to obtain interval extensions of transcendental functions. See Kearfott [1988c] for an elementary example. Alternate extensions are also possible; see Ratschek and Rokne [1984].

Several software packages [Bleher et al. 1987; Bundy 1984; Yohe 1979] are available for interval extensions of the elementary functions.

Mathematically rigorous interval extensions can be computed in finite precision arithmetic via the use of *directed roundings*. Let x and y be machine-representable numbers, and assume op is one of the four elementary operations +, -, *, or /. Normally, $x \ op \ y$ is not representable in the machine's memory, and there are various schemes of rounding. For example, we may always *round down* to the nearest *machine number less than x op y*, or we may always *round up* to the *nearest machine number greater than x op y*. In interval arithmetic with directed rounding, if

$$x \ op \ y = [c, d],$$

then we always round the value for c down, and we always round the value for d up.

Machine interval arithmetic with directed rounding does not involve deep concepts, but it can be quite powerful. For example, if interval arithmetic with directed rounding is used to compute an interval extension f of f,

$$[c, d] = f([a, b]),$$

and $[c, d]$ does not contain zero, then this is a rigorous proof (regardless of the machine wordlength, etc.) that there is no root of f in $[a, b]$.

The IEEE floating point standard, and hence hardware on many personal computers and mainframes, include support for directed roundings.

Various precompilers and compilers support the interval datatype. An early, reasonably portable one consisted of interval arithmetic packages of the Mathematics Research Center [Yohe 1977; Yohe 1979] in conjunction with the AUGMENT precompiler [Crary 1976]. ACRITH is a high-quality multiple precision and interval package for IBM mainframe equipment [Bleher et al. 1987], which has been used with AUGMENT in the past. More recently, researchers at IBM-Germany have developed the modern precompiler Fortran-SC [Walter and Metzger 1988] to access the ACRITH routines.

PASCAL-SC supports an interval data type directly in the compiler [Rall 1987]. There is much literature on using PASCAL- SC, even though it has until recently only been available on CPM-based personal computers. (There is now a version of PASCAL- SC for IBM PC-compatibles.)

L. A. Liddiard, E. J. Mundstock, and W. Walster wrote the "M-77" Fortran *compile*r which supports the interval data type. This compiler runs only on CDC equipment, but is available from CDC or from the University of Minnesota.

Clemmesen has described pseudocode outlining implementation of interval arithmetic [1984].

The speed of interval arithmetic operations varies greatly depending on implementation. Our experience is, however, that even the slowest implementations allow certain practical problems to be solved.

3. INTERVAL METHODS IN SOLVING NONLINEAR SYSTEMS OF EQUATIONS AND IN NONLINEAR OPTIMIZATION

Here, we discuss solution of (1.1) and (1.2), done via generalized bisection in conjunction with interval Newton methods. This general technique is described in chapters 19 and 20 of [Alefeld et al. 1983] and in chapters 5 and 6 of [Moore 1979]. An early paper on the technique is [Hansen 1968]. Other papers are (but are not limited to) [Hansen 1978; Kearfott 1987a; Kearfott 1987b; Moore 1977; Moore and Jones 1977; Neumaier 1985; Nickel

1971; Nickel 1986b; Schwandt 1985a].

In what follows we denote by X_0 the box in n-space described by

$$[\, X = (x_1, x_2, ..., x_n) \mid l_i \leq x_i \leq u_i for 1 \leq i \leq n \,].$$

In interval Newton methods, we find a box $\overline{X_k}$ which contains all solutions of the interval linear system

(3.1) $$F'(X_k)(\overline{X_k} - X_k) = -F(X_k),$$

where $F'(X_k)$ is a suitable interval extension of the Jacobian matrix of F over the box X_k, and where X_k is some point in X_k. We then define the next iterate X_{k+1} by

(3.2) $$X_{k+1} = X_k \cap \overline{X_k}.$$

The scheme based on solving (3.1) and performing (3.2) is termed an *interval Newton* method. The *convergence rate* of an interval Newton method is determined by the ratios of the *widths* of the component intervals of X_{k+1} to the corresponding widths of X_k.

If each row of F' contains all possible vector values that that row of the scalar Jacobian matrix takes on as X ranges over all vectors in X_k, then it follows from the mean value theorem that all solutions of (1.1) in X_k must be in X_{k+1}. If the coordinate intervals of X_{k+1} are smaller than those of X_k, then we may iterate (3.1) and (3.2) until we obtain an interval vector the widths of whose components are smaller than a specified tolerance.

If the coordinate intervals of X_{k+1} are not smaller than those of X_k, then we may *bisect* one of these intervals to form two new boxes; we then continue the iteration with one of these boxes, and push the other one onto a *stack* for later consideration. After completion of the current box, we pop a box from the stack, and apply (3.1) and (3.2) to it; we thus continue until the stack is exhausted. As is explained in [Moore and Jones 1977; Kearfott 1987a] and elsewhere, such a composite *generalized bisection* algorithm will reliably compute all solutions to (1.1) to within a specified tolerance.

The efficiency of the generalized bisection algorithm depends on

(1) the sharpness of the interval extension of the Jacobian matrix; and

(2) the way we find the solution $\overline{X_k}$ to (1.2).

In particular, iteration with formulas (3.1) and (3.2) should exhibit the quadratic local convergence properties of Newton's method, but repeated bisections are to be avoided if possible. We are thus interested in arranging the computations so that $\overline{X_k}$ has coordinate intervals which are as narrow as possible.

For many ways of solving (3.1), the following statement is true.

(3.3) if $\overline{X_k}$ is strictly contained in X_k, then the system of equations in (1.1) has a unique solution in X_k, and Newton's method starting from any point in X_k will converge to that solution. Conversely, if $X_k \cap \overline{X_k}$ is empty, then there are no solutions of the system in (1.1) in X_k.

See [Hansen and Walster 1988; Neumaier 1985; Qi 1982].

The Krawczyk method [Moore 1977] and an interval version of the Gauss-Seidel method [Hansen 1983] are popular ways of solving (3.1). See [Kearfott 1988b] for details, and see [Kearfott 1988c] for a detailed example of the interval Gauss- Seidel method. In both of these methods, it is usually necessary to first multiply (3.1) by a *preconditioner matrix* Y_k of real numbers. This matrix is often taken to be the inverse of the matrix whose entries are the midpoints of the entries of $F'(X_k)$.

For large, banded or sparse systems, multiplication by an inverse is impractical. Schwandt uses the interval Gauss-Seidel method without a preconditioner to solve systems like finite difference discretizations of Poisson's equation with a nonlinear forcing term [Schwandt 1984; Schwandt 1985b]. In such cases, an interval generalization of diagonal dominance ensures convergence of repeated iteration of the interval Gauss-Seidel method when Y_k is the identity matrix [Rall 1987].

Techniques for computing the rows of Y_k to minimize the widths of the intervals obtained via the interval Gauss-Seidel method (and thus maximize convergence rate) appear in [Kearfott 1988b]. These techniques involve solving linear programming problems for the elements of Y_k; these linear programming problems express optimality conditions for the width of each component interval of $\overline{X}$. The techniques are applicable to illconditioned and singular systems, and the linear programming problem can be altered to take account of structure or sparsity.

The solutions to the global nonlinear optimization problem (1.2) can be found by solving (1.1), where the f_i are the components of $\nabla \phi$. However, we may use the objective function directly to increase the algorithm's efficiency. If p and q are intervals, we say that $p > q$ if every element of p is greater than every element of q. Suppose that X and Y are interval vectors in the stack described below (3.2), and let ϕ be an interval extension to ϕ. Then, if $\phi(Y) > \phi(X)$, we may discard Y from the stack.

Papers and reviews on solution of the global optimization problem include [Baumann 1986; Hansen 1980; Hansen 1988; Ichida and Fujii 1979; Nickel 1986b; Rall 1985; Ratschek and Rokne 1987]. Walster, Hansen, and Sengupta report performance results on their global optimization algorithm in [Walster et al. 1985]. For performance results of a general interval nonlinear equation algorithm, see [Kearfott 1987b].

4. A SUMMARY OF STRENGTHS, WEAKNESSES, AND OPEN QUESTIONS OF INTERVAL OPTIMIZATION ALGORITHMS

4.1 Strengths and Weaknesses

Roughly, interval techniques work well for small and moderately sized polynomial and transcendental systems of the form (1.1) or (1.2). The performance is in general less predictable when there are larger numbers of variables, and it may not presently be practical to apply interval methods when the objective function is complicated (such as when its evaluation involves integrating systems of differential equations).

Interval methods *cannot* fail by identifying a point other than the global optimum of (1.2), or by terminating without finding all solutions to the system in (1.1). However, due to poor interval extensions or other reasons, they can take an excessive amount of computation time. On the other hand, they take considerably less computation time than alternate techniques (which also can, in theory, fail) for certain problems; see [Kearfott 1987b; Walster et al. 1985]. Regardless of relative efficiency, completion of a correct algorithm which employs interval arithmetic with correct directed rounding gives bounds on the solutions which are valid with the same certainty as a rigorous mathematical proof. This fact may be more important than computation time in some cases.

Interval methods for root-finding or for nonlinear optimization are well suited to constrained problems of the form (1.1) or (1.2), and should be seriously considered when the problems take that form.

4.2 Open Questions

First, apart from studies on certain special systems as in [Garloff 1986; Schwandt 1984; Schwandt 1985a; Schwandt 1985b; Thiel 1986], there is not much literature on the behavior of interval methods for large-scale systems. Indeed, the methods' practicality is less predictable for such systems. Nonetheless, we believe that special methods for structured problems can be developed which are predictably effective on wider classes of problems. These methods may involve variants of the preconditioner technique in [Kearfott 1988b] or of the subspace technique in [Kearfott 1988a].

Second, the question of singular and ill-conditioned systems is not fully resolved. These questions are addressed in [Kearfott 1988a; Kearfott 1988b], but additional work is desirable.

Third, it is not yet fully clear for what class of f_i these methods are presently practical. Often, if more operations are required to evaluate the f_i, then the resulting interval f_i will have a larger width and hence will be less useful. In our opinion, more studies need to be carefully done and reported.

We suggest that operations researchers explore the usefulness of interval techniques with the presently available software (e.g. [Kearfott and Novoa 1988]), and consult interval mathematicians as questions arise.

REFERENCES

Alefeld, G., and Herzberger, J. (1983), *Introduction to Interval Computations*, Academic Press, New York, etc.

Baumann, E. (1986), "Globale Optimierung stetig differenzierbarer Functionen einer Variablen," preprint, *Freiburger Intervall-Berichte 86*, 6, Institut für Angewandte Mathematik der Universität Freiburg, 1-89.

Bleher, J. H., Rump, S. M., Kulisch, U., Metzger, M., and Walter, W. (1987), "Fortran-SC – A Study of a Fortran Extension for Engineering Scientific Computations with Access to ACRITH," *Computing 39*, 2, 93-110.

Bundy, A. (1984), "A Generalized Interval Package and its Use for Semantic Checking," *ACM Transactions on Mathematical Software 10*, 397-409.

Clemmesen, M. (1984), "Interval Arithmetic Implementations Using Floating Point Arithmetic," *SIGNUM News 19*, 4, 2-8.

Crary, F. (1976), "The AUGMENT Precompiler," Technical Report 1470, Mathematics Research Center, University of Wisconsin, Madison, Wis.

Garloff, J. (1985), "Interval Mathematics: A Bibliography," preprint, *Freiburger Intervall-Berichte 85*, 6, Institut für Angewandte Mathematik der Universität Freiburg, 1- 222.

Garloff, J. (1986), "Block Methods for the Solution of Linear Interval Equations," preprint, *Freiburger Intervall Berichte 86*, 3, Institut für Angewandte Mathematik der Universität Freiburg, 1-40.

Garloff, J. (1987), "Bibliography on Interval Mathematics, Continuation," preprint, *Freiburger Intervall Berichte 87*, 2, Institut für Angewandte Mathematik der Universität Freiburg, 1-50.

Hansen, E. R. (1968), "On Solving Systems of Equations Using Interval Arithmetic," *Mathematics of Computation 22*, 374- 384.

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

Hansen, E. R. (1978), "Interval Forms of Newton's Method," *Computing 20*, 153-163.

Hansen, E. R. (1980), "Global Optimization Using Interval Analysis – The Multidimensional Case," *Numerische Mathematik 34*, 3, 247-270.

Hansen, E. R., and Greenberg, R. I. (1983), "An Interval Newton Method,"*Applied Mathematics and Computation 12*, 2-3, 89- 98.

Hansen, E. R. (1988), "An Overview of Global Optimization Using Interval Analysis," In *Reliability in Computing*, R. E. Moore, Ed. Academic Press, New York.

Hansen, E. R., and Walster, G. W. (1988), "Nonlinear Equations and Optimization," accepted for publication in *Mathematical Programming*.

Ichida, K., and Fujii, Y. (1979), "An Interval Arithmetic Method for Global Optimization," *Computing 23*, 1, pp. 85-97.

Kearfott, R. B. (1987a), "Abstract Generalized Bisection and a Cost Bound," *Mathematics of Computation 49*, 179 (July), 187-202.

Kearfott R. B. (1987b), "Some Tests of Generalized Bisection," *ACM Transactions on Mathematical Software 13*, 3 (Sept.), 197-220.

Kearfott, R. B. (1988a), "On Handling Singular Systems with Interval Newton Methods," In *Proceedings of the Twelfth IMACS World Congress on Scientific Computation* (Paris, France, July 18-22), pp. 651-653.

Kearfott, R. B. (1988b) "Preconditioners for the Interval Gauss- Seidel Method," submitted to the *SIAM Journal on Numerical Analysis*.

Kearfott, R. B. (1988c) "Interval Arithmetic Techniques in the Computational Solution of Nonlinear Systems of Equations: Introduction, Examples, and Comparisons," *In Lecture Notes for the AMS-SIAM Summer Seminar in Applied Mathematics*, (Colorado State University, July 18-29).

Kearfott, R. B., and Novoa, M. (1988), "A Program for Generalized Bisection," submitted to the *ACM Transactions on Mathematical Software*.

Miranker, W. L., ed. (1986), *Accurate Scientific Computations*, Lecture Notes in Computer Science no. 235, Springer Verlag, New York, etc.

Moore, R. E. (1977), "A Test for Existence of Solutions to Nonlinear Systems," *SIAM Journal on Numerical Analysis* 14, 4 (Sept.), 611-615.

Moore, R. E., and Jones, S. T. (1977), "Safe Starting Regions for Iterative Methods,"*SIAM Journal on Numerical Analysis* 14, 6 (Dec.), 1051-1065.

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

Moore, R. E., ed. (1988),*Reliability in Computing*, Academic Press, New York, etc.

Neumaier, A. (1985), "Interval Iteration for Zeros of Systems of Equations," *BIT 25*, 1, 256-273.

Nickel, K. (1971), "On the Newton Method in Interval Analysis," Technical Report 1136, Mathematics Research Center, University of Wisconsin, Madison, Wis.

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

Nickel, K., ed. (1986a), *Interval Mathematics 1985*, Lecture Notes in Computer Science volume 212, Springer Verlag, Berlin.

Nickel, K. (1986b), "Optimization Using Interval Mathematics," preprint, *Freiburger Intervall-Berichte 86*, 7, Institut für Angewandte Mathematik der Universität Freiburg, 55- 83.

Qi, L. (1982), "A Note on the Moore Test for Nonlinear Systems," *SIAM Journal on Numerical Analysis 19*, 4 (Aug.), 851-857.

Rall, L. B. (1981), "Interval Analysis: A New Tool for Applied Mathematics," Technical Report 2268, Mathematics Research Center, University of Wisconsin, Madison, Wis.

Rall, L. B. (1985), "Global Optimization Using Automatic Differentiation and Interval Iteration," Technical Report 2832, Mathematics Research Center, University of Wisconsin, Madison, Wis.

Rall, L. B. (1987), "An Introduction to the Scientific Computing Language Pascal-SC," *Computers and Mathematics with Applications 14*, 1, 53-69.

Ratschek, H. (1987), "Interval Mathematics," preprint, *Freiburger Intervall Berichte 87*, 4, Institut für Angewandte Mathematik der Universität Freiburg, 1-44.

Ratschek, H., and Rokne, J. G. (1984), Computer Methods for the *Range of Functions*, Horwood, Chichester, England.

Ratschek, H., and Rokne, J. G. (1987), "Efficiency of a Global Optimization Algorithm," SIAM Journal on Numerical Analysis 24, 5 (Oct.), 1191-1201.

Schwandt, H. (1984), "An Interval Arithmetic Approach for the Construction of an Almost Globally Convergent Method for the Solution of the Nonlinear Poisson Equation," SIAM *Journal on Scientific and Statistical Computing* 5, 2 (June), 427-452.

Schwandt, H. (1985a), "Krawczyk-Like Algorithms for the Solution of Systems of Nonlinear Equations,"*SIAM Journal on Numerical Analysis 22*, 4 (Aug.), 792-810.

Schwandt, H. (1985b), "The Solution of Nonlinear Elliptic Dirichlet Problems on Rectangles by Almost Globally Convergent Interval Methods," *SIAM Journal on Scientific and Statistical Computing 6*, 3 (July), 617-638.

Thiel, S. (1986), "Intervalliterationsverfahren für discretisierte elliptische Differentialgleichungen," preprint,*Freiburger Intervall Berichte 86*, 8, Institut für Angewandte Mathematik der Universität Freiburg, 1-72.

Walster, G. W., Hansen, E. R., and Sengupta, S. (1985), "Test Results for a Global Optimization Algorithm," In *Numerical Optimization 1984*, (Boulder, Colo., June 12- 14). SIAM, Philadelphia, pp. 272-287.

Walter, W., and Metzger, M. (1988), "Fortran-SC, A Fortran Extension for Engineering/Scientific Computation with Access to ACRITH," In *Reliability in Computing*, R. E. Moore Ed. Academic Press, New York, etc.

Yohe, J. M. (1977), "The Interval Arithmetic Package," Technical Report 1755, Mathematics Research Center, University of Wisconsin, Madison, Wis.

Yohe, J. M. (1979), "Software for Interval Arithmetic: A Reasonably Portable Package," ACM Transactions on *Mathematical Software 5*, 1 (Mar.), 50-63.