SIAM
Society for Industrial and Applied Mathematics

Workshop
on
VALIDATED COMPUTING

Toronto, Canada
May 23–25, 2002

EXTENDED ABSTRACTS

edited by
R. Baker Kearfott

published in
El Paso, Texas
April 2002

Contents

Preface

This volume contains peer refereed extended abstracts of the SIAM Workshop "Validated Computing 2002", Toronto, Canada, May 23–25, 2002.

What is validated (reliable) computing? Reliable computing is essential. There is no feasible alternative. Modern societies rely more and more on computer systems. Usually, our systems appear to work successfully, but there are sometimes serious, and often minor, errors. Ever increasing reliance on computer systems brings ever increasing need for reliability.

Validated computing is one essential technology to achieve increased software reliability. Validated computing uses controlled rounding of computer arithmetic to guarantee that hypotheses of suitable mathematical theorems are (or are not) satisfied. Mathematical rigor in the computer arithmetic, in algorithm design, and in program execution allows us to guarantee that the stated problem has (or does not have) a solution in an enclosing interval we compute. If the enclosure is narrow, we are certain that we know the answer reliably and accurately. If the enclosing interval is wide, we have a clear warning that our uncertainty is large, and a closer study is demanded.

Intervals capture uncertainly in modeling and problem formulation, in model parameter estimation, in algorithm truncation, in operation round off, and in model interpretation.

The techniques of validated computing have proven their merits in many scientific and engineering applications. They help answer questions from, "How much irrigation water does a desert golf course return effectively unused to its bordering stream?" to "Will a near earth asteroid hit the earth, possibly ending life as we know it?".

The techniques of validated computing rest on solid and interesting theoretical studies in mathematics and computer science. Contributions from fields including real, complex and functional analysis, semigroups, probability, statistics, fuzzy logic, automatic differentiation, computer hardware, operating systems, compiler construction, parallel processing, and software engineering are all essential.

Applications. The major emphasis of the program is on applications. This volume contains contributions from many people who have used tools from validated computing to attack, and often solve, significant practical problems. Successful applications have included medical diagnosis and treatment, financial

simulation, mechanical design, oil reservoir simulation, aeronautics, high energy particle accelerators, environmental engineering, chemical process simulation and control, computer graphics for motion picture special effects, astrophysics, and many more.

Not all applications are as yet successful. This volume also contains challenging applications to which validated techniques have not yet been successfully applied. Hopefully, by encouraging experts in such applications to lay out their problems, we will foster long-term collaborations leading to significant advances in those fields.

Validated computing and optimization. The workshop follows the SIAM Optimization meeting. The reason we decided to relate these two meetings is that global optimization is a major concern of both the optimization and the validated computing communities. By holding the meetings consecutively, we encourage validated computing researchers to become more involved in the wider optimization community, and we encourage people more interested in standard techniques of optimization to participate in interval discussions.

Ray Moore. Our special emphasis during this meeting is on Ramon E. "Ray" Moore. At the workshop, we will have one special session and a conference banquet to honor Ray Moore. His 1966 book defined the field, he pioneered many applications, and he continues to contribute insights and papers. Most of the ideas in our interval algorithms of today directly trace their ancestry to Ray's 1966 and 1979 (from SIAM) books.

Thanks. We want to use this opportunity to thank all the contributors and participants of the workshop. This volume contains papers authored by researchers from all over the world, from Austria, Belgium, Brazil, Bulgaria, Canada, China, Czech Republic, Denmark, England, France, Germany, Hungary, India, Paraguay, Russia, Spain, and USA. Without their active participation, we could not have succeeded.

We want to also thank SIAM for their enormous help, and NASA Pan-American Center for Earth and Environmental Studies (NASA grant NCC5-209) for publishing these abstracts.

Daniel Berleant, Iowa State University, berleant@iastate.edu
George Corliss, Marquette University, George.Corliss@Marquette.edu
Ken Jackson, University of Toronto, krj@cs.toronto.edu
R. Baker Kearfott, U. of Louisiana at Lafayette, rbk@louisiana.edu
Vladik Kreinovich, University of Texas at El Paso, vladik@cs.utep.edu
Weldon Lodwick, U. of Colorado at Denver, weldon.lodwick@cudenver.edu
Bill Walster, Sun Microsystems, Bill.Walster@eng.sun.com

Interval Analysis – An Application to Solvent Design

Luke E. K. Achenie
Department of Chemical Engineering, Unit 3222
University of Connecticut, Storrs, CT 06269

This presentation discusses the use of an interval analysis based global optimization approach for the systematic design of cleaning solvent blends (commonly referred to as blanket washes) for lithographic printing. The simultaneous consideration of associated process constraints, property requirements, and environmental restrictions makes the blanket wash design a rather difficult problem. To address this, we present a framework for designing cleaning solvent blends that meet thermo-physical property requirements and environmental restrictions. The resulting mathematical program is a mixed-integer optimization problem involving (a) the selection of solvents from a set of pure component solvents (the discrete problem) and (b) finding the blend composition (the continuous problem).

The framework has been used to solve an industrially relevant problem of designing optimal blends for blanket wash applications in the printing industry taking into account solvent power, viscosity and surface tension.

1 Mixture Design Problem Formulation

The computer-aided mixture design is composed of three main steps:

(i) Selection of pure components from a database (for example, for designing a binary mixture from a set of 10 pure components can result in $\binom{10}{2}$ or 45 combinations),

(ii) Determining the mixture composition that satisfies the property targets and

(iii) Ranking the candidate mixtures by some criteria such as overall cost.

The first step is a combinatorial problem; the second step is a continuous problem, which can be non-convex depending on the nature of the property

prediction techniques employed. We propose to use a mixed-integer nonlinear problem formulation that is general enough to handle several types of mixture estimation techniques. Obviously if the number of combinations (binary, ternary, etc) for pure components is small, one can enumerate them all and solve a series of continuous nonlinear programs.

In the proposed formulation, binary variables are used to denote the presence or absence of a pure-component solvent in the mixture, and a set of continuous variables are used to describe the mole fractions of the components in the mixture. Hence the formulation is mixed-integer in nature. First let us introduce the variables:

y_i (binary variable) =1 if pure component i is present in the mixture, and =0 otherwise;

x_i (continuous variable between 0 and 1) = mole fraction of pure component i in the mixture

Other parameters include:

n number of pure component solvents (basis set)

$n_{\max}$ maximum number of pure component solvents in the blend

P_{ij} property j of pure component i.

Constraints are imposed for (a) limiting the number of pure component solvents in the blend; (b) ensuring that the mole fraction of an absent component is 0; and (c) all the mole fractions add to 1.0. These constraints are by no means exhaustive, and several different ones can be added to achieve a specific solvent mixture design objective.

$$P_{mix} : \min_{x,y} f(x,y)$$

subject to :

$$P^L \leq P(x,y) \leq P^U, \quad \sum_i y_i \leq n_{\min}, \quad \sum_i x_i = 1, \quad 0 \leq x_i \leq y_i$$

$$x = [x_1, \ldots, x_n]^T, \quad y = [y_1, \ldots, y_m]^T, \quad x_i \in [0,1] \text{ (real)}, \quad y_i \in \{0,1\} \text{ (binary)}$$

P^L and P^U are lower and upper limits on a vector of target properties P. These properties may be nonlinear and nonconvex with respect to the search variables x_i, y_i.

The last constraint in the above formulation ensures that if component i is not present in the mixture (i.e. $i = 0$), then the corresponding composition x_i is also 0. This however, can lead to cases where the composition of one component is infinitesimally small. To avoid this we replace it by: $u_i \cdot \varepsilon < x_i < u_i(1-\varepsilon)$, where ε is a small number (e.g. 0.01).

2 Interval Analysis Technique for Solving Mixture Problem

Since many property estimation techniques are generally non-convex, we have developed an interval analysis based optimization strategy that can design (globally) optimal mixtures. Interval analysis has emerged as a reliable mathematical tool that can automatically generate lower and upper bounds for a function [2]. It has been used for solving ordinary differential equations, linear systems, and verifying chaos. Interval arithmetic, which is at the heart of interval analysis, was developed by Moore [6].

In essence, interval analysis based optimization continually deletes portions of the search space with the goal of maintaining a final box of desired width that contains the global solution. A number of interval-based optimization procedures have been developed (e.g. [2, 3, 5, 7, 10]). Most of these procedures are tailored to unconstrained optimization problems. In addition, these techniques can only handle continuous variables. In other words they do not handle discrete variables. Notwithstanding the attractive features of interval-based global optimization, they are in general computationally intensive. To address some of these issues, we have developed new acceleration strategies, and extended the capabilities of the algorithm to solve mixed integer problems.

Almost all interval analysis based global optimization algorithms employ a successive domain reduction approach by eliminating portions of search regions, which do not contain the global solution. Consider the continuous optimization model

$$\begin{gathered} \text{globally } \min_x f(x) \\ \text{subject to :} \\ g(x) \leq 0; \quad h(x) = 0 \\ x = [x_1, \ldots, x_n]^T \text{ (real)}; \quad x \in \mathbf{X_0} \end{gathered} \tag{2}$$

Almost all domain reduction algorithms invariably use the following tests to systematically remove portions of the domain that cannot contain the global minimum: (a) Upper Bound Test, (b) Infeasibility Test, (c) Monotonicity Test, (d) Non-convexity Test, and (e) Distrust Region Test [10]. An interval-based global optimization algorithm can be constructed based on the above tests. However, in our experience it is computationally slow especially for problems with a large number of constraints. We propose additional domain reduction tests. These are (a) Upper Bound via SQP local optimization and (b) Local Feasibility Test. In (b) the idea is to relax the optimization model and only consider the convex constraints and determine if this relaxed search space contains a feasible solution. This requires the prior specification of which constraints are convex and which are not. This is not always straightforward. However, linear equality and linear inequality constraints are simple convex constraints. Based

on this reduced set of constraints the feasibility of a sub-region $\mathbf{X}_k$ is checked by solving a feasibility problem.

Mixture design problems have relatively small dimension. For a design with a basis set of m pure components the interval dimension is $2m$. Current interval based global optimization algorithms can only solve continuous optimization problems. An extension of the algorithm is made for solving mixed integer nonlinear programs (MINLP) such as the mixture design problem.

3 Case Study: Design of Environmentally Acceptable Blanket Wash Blends

The Printing Industry of America (PIA) and the USEPA started a major initiative in the early 1990's to search for alternative water-based blanket wash solvents [9]. This case study explores the systematic development of aqueous blends for use as blanket wash solvents.

The EPA report on blanket wash risk assessment [1] lists 40 different formulations (or solvent blends) used as blanket washes by different printing facilities throughout the United States. However, due to propriety reasons their compositions are not reported. Out of these, 21 formulations contain petroleum distillates (hydrocarbons and/or aromatic hydrocarbons), which pose considerable environmental health and safety risks. Two common aromatic hydrocarbons used in blanket washes are 1-2-4 trimethyl benzene (C_9H_{12}) and isomers of xylene (C_8H_{10}). Trimethyl benzene has a flash point of 54.4oC and $\log K_{OW}$ of 3.78. Isomers of xylene have flash point as low as 17oC and $\log K_{OW}$ of 3.15. Thus both are flammable and have high bioaccumulation and toxicity.

The pure component solvents employed in this case study are non-halogenated and non-aromatic water-soluble compounds. Also only those solvents, which have relatively small environmental and health impact are selected. The desired attributes for optimal blanket wash formulation target the solvent power, its flow characteristics, surface contacting and environmental impact.

An MINLP model of the blanket wash mixture design problem was formulated and solved two ways. In Case 1, the model was solved by fixing the binary variables resulting in an NLP model. Specifically each binary mixture was constructed by fixing the binary variables for water and one of the other pure component solvents to 1; the remaining binary variables were set at 0. In Case 2 the MINLP model was solved rigorously, i.e. without fixing the binary variables. This is a relatively more difficult problem with 17 variables and 8 binary variables. Also in this case we only considered 2-component blends (i.e. binary blend, not to be confused with binary variable). Thus the solution approach not only picks the best combination of 2-component solvents (discrete problem) but also finds the optimal composition (continuous problem).

The two models were solved to obtain 7 different binary mixtures. Among

all solutions, the lowest objective value was achieved by a γ-butyrolactone and water blend.

References

[1] Design for the Environment Program, *Cleaner Technologies Substitute Assessment: Lithographic Blanket Washes*, EPA 744-R-97-006, USEPA, 1997.

[2] E. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., 1992.

[3] R. J. V. Iwaarden, *An Improved Unconstrained Global Optimization Algorithm*, University of Colorado at Denver, Denver, 1996.

[4] E. M. Krishner, "Environment, Health Concerns Force Shift In Use Of Organic Solvents", *Chemical and Engineering News*, 1995, June 20, pp. 13–20.

[5] R. Moore, E. R. Hansen, and A. Leclerc, "Rigorous Methods for Global Optimization", In: C. A. Floudas and M. Pardalos (eds.), *Recent Advances in Global Optimization*, Princeton University Press, 1992.

[6] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[7] H. Ratscheck and J. Rokne, "Interval Tools for Global Optimization", *Computers Math. Applic.*, 1991, Vol. 21, No. 6/7, pp. 41–50.

[8] Toxics Reduction Institute, *Demonstration of Printwise TM: A "Near Zero" Lithographic Ink and Blanket Wash System*, 39, University of Massachusetts, Lowel, 1997.

[9] United States Environmental Protection Agency, *Cleaner Technologies Substitutes Assessment: Lithographic Blanket Washes*, EPA 744-R-97-006, United States Environmental Protection Agency, 1997.

[10] R. Vaidyanathan and M. El-Halwagi, "Global Optimization of Nonconvex Nonlinear Programs Via Interval Analysis", *Computers and Chemical Engineering*, 1994, Vol. 18, No. 10, pp. 889–897.

Ramon E. Moore and 45 Years of Interval Analysis

G. Alefeld
Karlsruhe

In his fundamental book (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966) Ramon E. Moore collected for the first time his and others' results on interval analysis known up to this date. Furthermore many basic ideas on where and how interval analysis can be applied were indicated. In this talk it is shown by a couple of examples, how many of today's well known facts in interval analysis can be traced back to Moore's work. Furthermore some new results are presented.

Total-Step and Successive Overrelaxation Methods for LCP-Problems with Interval Data

G. Alefeld and U. Schäfer
University of Karlsruhe

Let there be given an (n, n) matrix M and a vector $q \in \mathbb{R}^n$. The linear complementarity problem (LCP-problem) consists in finding a vector $x^* \geq 0$ such that

$$Mx^* + q \geq 0 \quad \text{and} \quad x^{*T}(Mx^* + q) = 0, \qquad \text{(LCP)}$$

or to show that no such vector exists. This problem has many applications; see [1] and [2], for example.

In this talk, we are starting with an (n, n) interval matrix $[M]$ and an interval vector $[q]$ with n components. Using the total-step method and the successive overrelaxation method, respectively, we compute interval vectors $[x^k]$ which (under certain conditions on $[M]$ and $[x^0]$) contain the solutions of (LCP) for all $M \in [M]$ and all $q \in [q]$. Furthermore the convergence of $\{[x^k]\}$ to some limit $[x^*]$ is shown. Applications to this problem can be found in [3]. Some numerical examples are given.

References

[1] R. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, 1992.

[2] K. G. Marty, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, Berlin, 1988.

[3] U. Schäfer, *Das lineare Komplementaritätsproblem mit Intervalleintrgen*, Ph.D. Thesis, Fakultät fr Mathematik, Universität Karlsruhe, 1999.

On an Order Relation Between Distributions with Applications to Interval Estimation

R. Alt[1], I. Dakov[2], and S. Markov[2]

[1]Laboratoire LIP6, Universite Pierre et Marie Curie
4 Place Jussieu, 75005 Paris, France
[2]Section "Biomathematics"
Inst. of Mathematics and Computer Sci.
Bulgarian Academy of Sciences
"Acad. G. Bonchev" st., block 8,
BG-1113 Sofia, Bulgaria

1 Introduction

A problem which appears quite often in the validation of the results of numerical algorithms using intervals is the branching based on the comparison of two intervals when these intervals have a noon-empty intersection. This paper deal with some possibility of ordering intervals with some probability and a technique for the computation of this probability.

Let f be a density function (i.e., a Lebesgue integrable function on $\mathbb{R}$ such that $f(x) \geq 0$ and $\int_{-\infty}^{\infty} f(x)dx = 1$) and let $F(x) = \int_{-\infty}^{x} f(t)dt$ be the distribution (function) of f.

Given two independant real random variables ξ, η with known densities (or distributions) we want to compute the probability $P(\xi > \eta)$. This may be useful whenever the random variables represent numbers containing stochastic errors as is the case with stochastic numbers [1–4].

Denote by f_ξ, g_η the density functions of the random variables ξ, η resp. If both densities f_ξ, g_η have intervals as support sets $A, B \subset \mathbb{R}$, that is:

i) $f_\xi \neq 0$ for $\xi \in A$, and $f_\xi = 0$ for $\xi \notin A$;

ii) $g_\eta \neq 0$ for $\eta \in B$, and $g_\eta = 0$ for $\eta \notin B$,

and if the support sets do not intersect ($A \cap B \neq \emptyset$), then we have that $P(\xi > \eta)$ is either 0 or 1, depending on whether $A > B$ or $A < B$. In these cases one can speak of an order relation between distributions, that is a relation of the form $f_\xi < g_\eta$, resp. $f_\xi > g_\eta$. In the general case, when $0 < P(\xi > \eta) < 1$ the number $P(\xi > \eta)$ can serve as a measure (indicator) for such an ordering. For this reason we shall denote $M(f_\xi, g_\eta) = P(\xi > \eta)$.

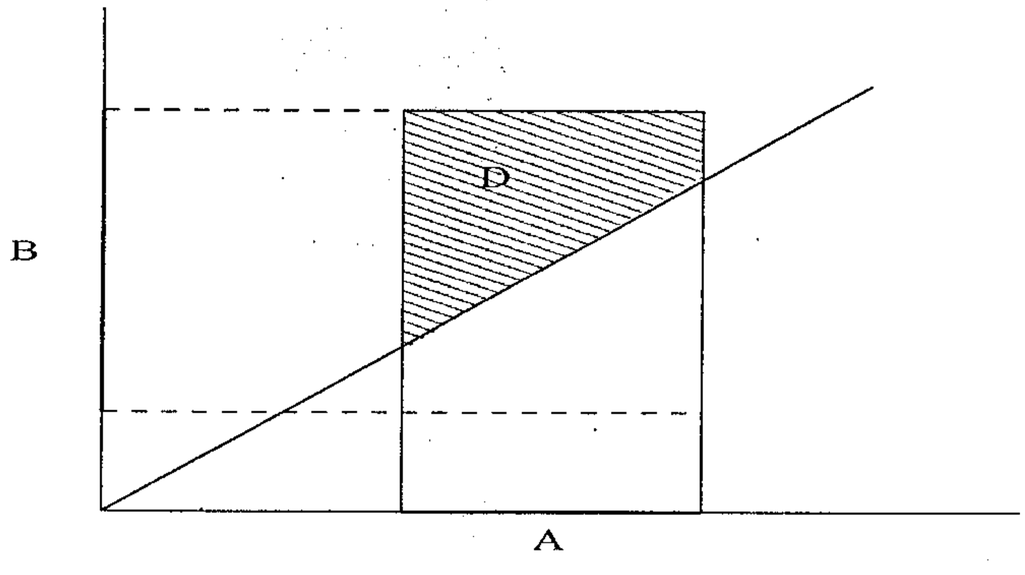


Figure 1: *domain definition for* $Pr(\zeta < 0)$

In what follows we investigate some rules for the computation of $M(f_\xi, g_\eta)$ especially suitable for the case when the support sets of the densities are (compact) intervals. A possible way to compute $M(f_\xi, g_\eta)$ is classically the following. The definition domain of $\Pr(\zeta < 0)$ is represented in figure (1), so we have:

$$\begin{aligned} M(f_\xi, g_\eta) &= P(\xi > \eta) = P(\zeta > 0) = 1 - P(\zeta \leq 0) \\ &= 1 - \int_{-\infty}^{0} h_\zeta(\zeta) d\zeta = 1 - \int_{z=-\infty}^{0} \int_{t=-\infty}^{\infty} f_\xi(z+t) g_\eta(t) dz\ dt, \end{aligned} \quad (1)$$

where $f_\xi(x)$ is the density of ξ, and $g_\eta(x)$ is the density of η.
Thus formula (1) can be used for the computation of $M(f_\xi, g_\eta) = P(\xi > \eta)$. However, we show here that an easier technique can be used in the case when the support sets of the densities are intervals.

So, in the present work a simple method is proposed for the computation of the probability $P(\xi > \eta)$, where ξ, η are two real random variables with known densities. The method is illustrated for some familiar densities Gaussian, uniform).

2 Technique

Let $\mathbb{R}^* = \mathbb{R} \cup \{-\infty\} \cup \{\infty\}$. Let $B = [\underline{b}, \overline{b}]$, $\underline{b} \leq \overline{b}$, $\underline{b}, \overline{b} \in \mathbb{R}^*$. The probability $P(\xi \in B)$ for a random variable with a density f to belong to the interval B is given by $P(\xi \in B) = F(\overline{b}) - F(\underline{b}) = \int_{\underline{b}}^{\overline{b}} f_\xi(t) dt$; note that $F(-\infty) = 0$ and $F(\infty) = 1$.

Proposition. *For any two random variables* ξ, η *with given density functions* f_ξ, g_η *and any integer* $n \geq 0$ *and system of real numbers* $t_1, \ldots, t_n \in \mathbb{R}$ *and* $t_i < t_{i+1}$ *(for* $n = 0$ *the system is considered empty), we have:*

$$M(f_\xi, g_\eta) = \sum_{k \geq l, k, l=1}^{n+1} p_{kl} \int_{t_{k-1}}^{t_k} f_\xi(t) dt \int_{t_{l-1}}^{t_l} g_\eta(t) dt, \quad (2)$$

wherein

$$p_{kl} = \begin{cases} 0, & k < l, \\ 1, & k > l. \end{cases} \tag{3}$$

In the case $k = l$ the value of p_{kk} depends on the details of f_ξ and g_η.

Proof. Denote $A_1 = [-\infty, t_1], A_i = [t_{i-1}, t_i], i = 1, \ldots, n, A_{n+1} = [t_n, \infty]$. Applying the conditional probabilities formula for the division of $\mathbb{R}$ we can write:

$$\begin{aligned} M(f_\xi, g_\eta) &= P(\xi > \eta) = \sum_{k,l=1}^{n+1} P(\xi > \eta, \xi \in A_k, \eta \in A_l) \\ &= \sum_{k,l=1}^{n+1} P(\xi > \eta \mid \xi \in A_k, \eta \in A_l) P(\xi \in A_k) P(\eta \in A_l) \\ &= \sum_{k,l=1}^{n+1} p_{kl} P(\xi \in A_k) P(\eta \in A_l), \end{aligned}$$

where $p_{kl} = P(\xi > \eta \mid \xi \in A_k, \eta \in A_l) = \{0, \text{ if } k < l; 1, \text{ if } k > l\}$.

In the case $k = l$, that let us call $\tilde{f}_\xi$ and $\tilde{g}_\eta$ the densities of ξ and η on A_k. (Note that they are not the restrictions of f_ξ and g_η to A_k). Then we have:

$$p_{kk} = \int_{x=t_{k-1}}^{t_k} \int_{y=t_{k-1}}^{x} \tilde{f}_\xi(x) \tilde{g}_\eta(y) dx dy \tag{4}$$

This implies the proposition.

The above formula can be considered as a discretisation of (1) but it has the advantages that the integrals (4) are defined on the same sub-interval for the two variables and are thus easier to compute. Moreover the considered sub-intervals can be small and it may be easy to obtain bounds for the coefficients p_{kk}.

Three obvious cases when the value of M is easily determined can be retreived with this formula:

For any two random variables ξ, η with given density functions f, g, resp., we have

a) If $f \equiv g$, then $M(f, g) = 1/2$;

b) In the case when $f(x) = 0$ for every $x \leq t$, and $g(x) = 0$ for every $x \geq s$, $t \geq s$, then $M(f, g) = 1$;

c) In the case when $g(x) = 0$ for every $x \leq t$, and $f(x) = 0$ for every $x \geq s$, $t \geq s$, then $M(f, g) = 0$.

These cases are retreived under appropriate division $\{t_i\}$ on $\mathbb{R}$. Namely, in the case a) take $\mathbb{R}$ as a single interval (empty division set, $n = 0$); in the cases b), c) take $n = 1$, $t_1 = (s + t)/2$.

In what follows the preceding technique based on formula (2) is applied to the computation of the measure M in the cases of some well-known distributions.

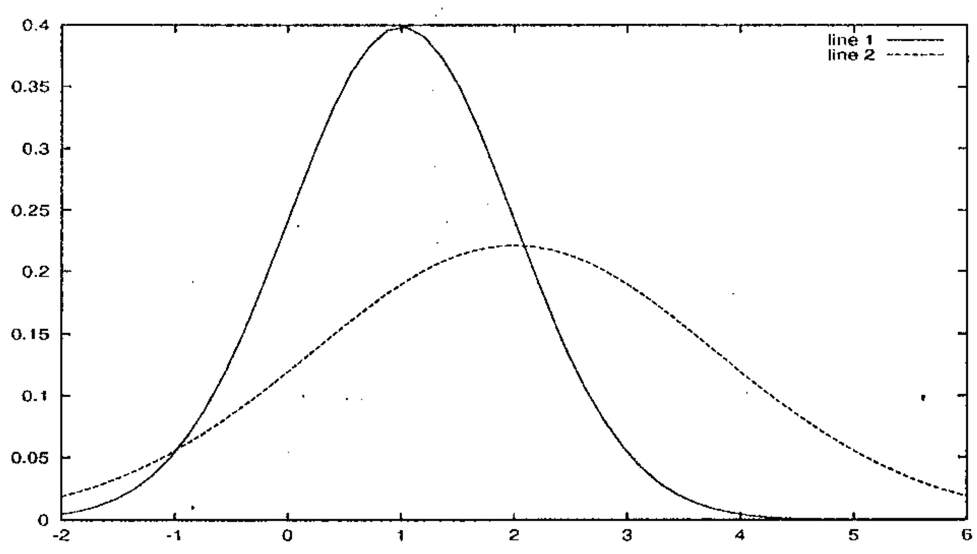


Figure 2: *two Gaussian distributions*

3 Uniform Distributions

Let us consider two random variables ξ and η defined on two intersecting intervals A and B. $A = [\underline{a}, \overline{a}]$ and $B = [\underline{b}, \overline{b}]$ with $\overline{a} > \underline{b}$. Take $n = 3$ and $t_1 = \underline{b}$ and $t_2 = \overline{a}$. With the notations of (3) we have obviously: $p_{12} = p_{13} = p_{23} = 0$ and $p_{21} = p_{31} = p_{32} = 1$. Concerning p_{11}, p_{22}, p_{33} we are in the case of both ξ and η belonging to the same sub-interval with a same constant probability density. So $p_{11} = p_{22} = p_{33} = 1/2$. Note that in fact only p_{22} has an interest as in formula (2) the probabilities $P(\eta \in A_1)$ and $P(\xi \in A_3)$ are null.

4 Gaussian Distributions

Let us consider the same intersecting intervals as above but with two different Gaussian distributions f_ξ, g_η on each interval, having mean values m_1, m_2, resp., and variances σ_1^2, σ_2^2, resp. Thus the density functions of ξ and η are: $f_\xi(x) = (2\pi\sigma_1^2)^{-1/2} e^{-(x-m_1)^2/(2\sigma_1^2)}$, $g_\eta(x) = (2\pi\sigma_2^2)^{-1/2} e^{-(x-m_2)^2/(2\sigma_2^2)}$. Assume $m_1 \leq m_2$. Take $n = 2$, t_1 and t_2 is the abscissas of the points common to $f_\xi(x)$ and $g_\eta(x)$, and denote $A_1 = [-\infty, t_1]$, $A_2 = [t_1, t_2]$, and $A_3 = [t_2, \infty]$. As before formula (3) gives $p_{kl} = 0$ for$k < l$ and $p_{kl} = 1$ for $k > l$. Concerning the coefficients $p_{kk}, k = 1, 2, 3$ they depend on the values of $m_1, m_2, \sigma_1, \sigma_2$ and they have to be computed with the integral of formula (4). As example let us here compute p_{11}. By hypothese ξ and η are in A_{11} then $Pr(\xi) \in A_1 = 1$ and $Pr(\eta) \in A_1 = 1$. So let us call: $S_1 = \int_{-\infty}^{t_1} f_\xi(t)dt$ and $S_2 = \int_{-\infty}^{t_1} g_\eta(t)dt$. Then $\tilde{f}_\xi = f_\xi/S_1$ and $\tilde{g}_\eta = g_\eta/S_2$ and $p_{11} = (\int_{x=-\infty}^{t_1} \int_{y=-\infty}^{x} f_\xi(x) g_\eta(y) dxdy)/(S_1 S_2)$.

Special case 1. $m_1 = m_2$. Then $(t_1 + t_2)/2 = m_1$. The problem is symmetric and $p_{11} = 1 - p_{33}$.

Special case 2. $\sigma_1 = \sigma_2 = \sigma$. In this case $t_1 = t_2$ and $A_{22} = \varnothing$.

Similar formulae have been deduced for the Beta distribution.

5 A Note on Applications

These results can be be used for treating branchings using comparisons of intervals when these intervals have a non empty intersection. They can also be applied to the problems of interpolation and approximation in the case of interval (uncertain but bounded) data with given densities (distributions). Thus this approach may find useful applications in mathematical modelling situations. On the other hand this approach contributes to the arithmetic theory of stochastic numbers [3].

References

[1] R. Alt and S. Markov, "On the Algebraic Properties of Stochastic Arithmetic. Comparison to Interval Arithmetic", In: W. Krämer and J. Wolff von Gudenberg (eds)., *Scientific Computing, Validated Numerics, Interval Methods*, Kluwer, 2001, pp. 331–341.

[2] R. Alt and J. Vignes, "Validation of Results of Collocation Methods for ODEs with the CADNA Library", *Appl. Numer. Math.*, 1996, Vol. 20, pp. 1–21.

[3] J. M. Chesneaux and J. Vignes, "Les fondements de l'arithmétique stochastique", *C.R Acad. Sci.*, Paris, Sér. I, Math., 1992, Vol. 315, pp. 1435–1440.

[4] J. Vignes and R. Alt, "An Efficient Stochastic Method for Round-Off Error Analysis", In: *Accurate Scientific Computations*, L.N.C.S. 235, Springer, 1985, pp. 183–205.

On Some Algebraic Properties of Stochastic Numbers

Rene Alt[1] and Svetoslav Markov[2]

[1]Laboratoire LIP6, Universite Pierre et Marie Curie
4 Place Jussieu, 75005 Paris, France
[2]Section "Biomathematics"
Inst. of Mathematics and Computer Sci.
Bulgarian Academy of Sciences
"Acad. G. Bonchev" st., block 8,
BG-1113 Sofia, Bulgaria

1 Introduction

Interval arithmetic and stochastic arithmetic have been both developed for the same purpose, i. e. to control errors coming from floating point arithmetic of computers and validate the results of numerical algorithms performed on computers. Interval arithmetic delivers guaranteed bounds for numerical results but requires special analysis and algorithms. On the other hand stochastic arithmetic is a model for the Cestac method which provides confidence intervals with known probability and can be easily implemented in existing numerical softwares. This work continues our study from [1] of the algebraic properties of stochastic arithmetic based on the comparison with interval arithmetic in midpoint-radius form, and on the algebraic structures that are induced by the operations on the two sets (stochastic numbers and intervals) cf. [7].

In the present paper following similar developements of interval arithmetic we introduce spaces analogous to quasilinear spaces [5, 6].

2 Stochastic Arithmetic

Stochastic arithmetic has been mainly studied in [3, 4, 9]. A *stochastic number* X is a gaussian random variable with a known mean value m and a known standard deviation σ and is denoted $X = (m, \sigma)$. The set of stochastic numbers is denoted as $S = \{(m, \sigma) \mid m \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$. Stochastic arithmetic is in fact a theoretical model for the discrete stochastic arithmetic which is used in the Cestac method in which m and σ are computed using a Monte-Carlo technique consisting in performing each arithmetic operation several times with

an arithmetic with a random rounding mode, see [2, 8, 9]. Hence the Cestac method takes naturally into account the correlation between errors whereas stochastic arithmetic actually does not. Anyhow in most applications the results predicted with stochastic arithmetic are identical or very close to those provided by the Cestac method. Thus stochastic arithmetic is considered as giving a good algebraic model for the Cestac method which uses the following classical property.

Property: If $X = (m, \sigma) \in S$, $0 \leq \beta \leq 1$ and r is a realization of X, then there exist λ_β only depending on β, such that

$$P\left(r \in [m - \lambda_\beta \sigma, m + \lambda_\beta \sigma]\right) = 1 - \beta. \tag{1}$$

$I_{\beta,X} = [m - \lambda_\beta\ \sigma, m + \lambda_\beta\ \sigma]$ is the *confidence interval* of X with probability $1 - \beta$. Equality (1) is a well-known property of gaussian random variables. For $\beta = 0.05, \lambda_\beta \approx 1.96$. The Cestac method computes m and σ by sampling, stochastic arithmetic computes m and σ algebraically.

3 Arithmetic Operations Between Stochastic Numbers

Let $X_1 = (m_1, \sigma_1)$ and $X_2 = (m_2, \sigma_2)$ be two stochastic numbers. (Usual) equality between two stochastic numbers X_1, X_2 is defined by: $X_1 = X_2$, if $m_1 = m_2$ and $\sigma_1 = \sigma_2$.

In this work we concentrate on the operations addition

$$X_1 + X_2 = (m_1 + m_2, \sqrt{\sigma_1^2 + \sigma_2^2})$$

and multiplication by scalars $\gamma \in \mathbb{R}$

$$\gamma * X = (\gamma m, |\gamma|\sigma).$$

We have shown in [1] that the set S is an abelian monoid with respect to addition with cancellation law.

Multiplication by scalars satisfies:

a) *First distributive law:* $\lambda * (X + Y) = \lambda * X + \lambda * Y$;

b) *Associativity:* $\lambda * (\mu * X) = (\lambda\mu) * X$;

c) *Identity:* $1 * X = X$.

Remark. The second distributive law: $(\lambda + \mu) * X = \lambda * X + \mu * X$ does not hold in general. Moreover, it does not generally hold even for λ, μ nonnegative. We thus have no quasi-distributive law (as in the case of intervals).

The mean values satisfy the distributive law and thus form a linear space. The standard deviations satisfy the following law:

$$(\sqrt{\lambda^2 + \mu^2}) * \sigma = \lambda * \sigma + \mu * \sigma, \quad \lambda \geq 0, \mu \geq 0,$$

or, equivalently,

$$(\sqrt{\lambda + \mu}) * \sigma = \sqrt{\lambda} * \sigma + \sqrt{\mu} * \sigma, \quad \lambda \geq 0, \mu \geq 0.$$

We investigate the space of standard deviations by embedding it in an additive group, obtaining thus a space close to a quasilinear space with group structure [6].

References

[1] R. Alt and S. Markov, "On the Algebraic Properties of Stochastic Arithmetic. Comparison to Interval Arithmetic", In: W. Krämer and J. Wolff von Gudenberg (eds.), *Scientific Computing, Validated Numerics, Interval Methods*, Kluwer, 2001, pp. 331–341.

[2] R. Alt and J. Vignes, "Validation of Results of Collocation Methods for ODEs with the CADNA Library", *Appl. Numer. Math.*, 1996, Vol. 20, pp. 1–21.

[3] J. M. Chesneaux, "The Equality Relation in Scientific Computing", *Numer. Algo.*, 1994, Vol. 7, pp. 129–143.

[4] J. M. Chesneaux and J. Vignes, "Les fondements de l'arithmétique stochastique", *C.R Acad. Sci.*, Paris, Sér. I, Math., 1992, Vol. 315, pp. 1435–1440.

[5] M. Kracht and G. Schröder, "Zur Intervallrechnung in Quasilinearen Räumen", *Computing*, 1973, Vol. 11, pp. 73–79.

[6] S. Markov, "On the Algebraic Properties of Intervals and Some Applications", *Reliable Computing*, 2001, Vol. 7, No. 2, pp. 113–127.

[7] S. M. Rump, "Fast and Parallel Interval Arithmetic", *BIT*, 1999, Vol. 3, pp. 534–554.

[8] J. Vignes, "Review on Stochastic Approach to Round-Off Error Analysis and its Applications", *Math. and Comp. in Sim.*, 1988, Vol. 30, No. 6, pp. 481–491.

[9] J. Vignes, "A Stochastic Arithmetic for Reliable Scientific Computation", *Math. and Comp. in Sim.*, 1993, Vol. 35, pp. 233–261.

Computer Algebra Style (Multiprecision) Interval and Complex Arithmetic

Franky Backeljauw[1], Annie Cuyt[1],
Brigitte Verdonk[1], and Johan Vervloet[1,2]

[1]Dept of Mathematics and Computer Science
University of Antwerp (UIA)
Universiteitsplein 1, B2610 Antwerp, Belgium
{backelj,cuyt,verdonk,jvvloet}@uia.ua.ac.be
[2]Research Assistant FWO–Vlaanderen

1 Motivation

Using the approach in [2, 4, 5], interval bounds for values in $\mathbb{R}$ or $\mathbb{R} \cup [-\infty, +\infty]$, which is what one is interested in in most industrial applications, can be computed. These approaches, however, do not in all cases reflect all that is known about the interval valued expressions, as is required during prototyping or in a computer algebra environment.

A similar remark holds for the complex arithmetic guidelines proposed in Annex G of the latest C programming language standard [1]. The approach is sufficiently correct when some additional background information is available about the evaluated expression. The lack of such information, however, may lead to the ambiguous interpretation of results.

In the authors' implementation, a more theoretical point of view on interval and complex arithmetic is proposed to tackle the above issue. It is natural, when taking a computer algebra style viewpoint, to deal not only with double precision, but also consider true higher precisions.

2 Interval Arithmetic: Theory versus Implementation

In both [2] and [4], the authors review the implementation of the basic operations in interval arithmetic, and in particular discuss the different approaches given in the literature for interval division when the divisor interval contains zero.

Division by an interval containing zero is a special case of an interval function for which the interval arguments contain points outside the domain of the un-

derlying point function. In [5] a general approach is presented to deal with such situations and to remove any restrictions on the domain of interval functions. This approach fully exploits the availability of the underlying IEEE hardware and has been efficiently implemented in [3].

While interval division is defined differently in [2] and [5] when the divisor contains zero, part of the difference can be traced back to the following. Underlying any implementation of interval arithmetic are two sets, a number set $\mathbb{S}$ and a set $I(\mathbb{S})$ of intervals, which is a subset of $2^{\mathbb{S}}$. In [2]

$$\begin{aligned}\mathbb{S} =& \mathbb{R}\\ I(\mathbb{S}) =& \{[a,b] \mid a,b \in \mathbb{S}, a \leq b\} \cup \{]-\infty, b] \mid b \in \mathbb{S}\}\\ & \cup \{[a,+\infty[\mid a \in \mathbb{S}\} \cup \{]-\infty,+\infty[\} \cup \{\emptyset\}\end{aligned}$$

while in [5]

$$\begin{aligned}\mathbb{S} &= \mathbb{R} \cup \{-\infty,+\infty\}\\ I\mathbb{S} &= \{[a,b] \mid a,b \in \mathbb{S}, a \leq b\}\end{aligned}$$

In both cases, an interval can be easily represented by a pair of (properly rounded) IEEE floating-point numbers. This ease of representation comes at a price, however, because the choice of the set $\mathbb{S}$ has crucial implications for the definition of interval functions. Even though an interval function satisfies the containment principle, it can only contain the range of the underlying point function in $\mathbb{S}$. If the range of the point function is a subset of $\mathbb{S}$, there is no problem. But if the underlying point function is undefined or complex-valued for some values of the interval arguments, returning an element of $I(\mathbb{S})$ may lead to unintuitive results, as we shall illustrate. This also explains why in [2]

$$[1,2]/[0,0] = \emptyset$$

while according to [5]

$$[1,2]/[0,0] = \{\infty\} \cap (\mathbb{R} \cup \{-\infty,+\infty\}) \subset [-\infty,+\infty]$$

In this presentation we give an alternative, computer algebra style approach to remove restrictions on the domain of interval functions. To achieve this, we allow for the efficient representation of non-real results. We indicate some important properties and advantages of this approach and show how the presented ideas can be implemented in a multiprecision interval arithmetic library without performance overhead.

3 Complex Arithmetic: Theory versus Implementation

In the same way that the approach in [3] is designed to offer interval arithmetic in a way that seamlessly blends in with IEEE floating-point arithmetic, the Annex

G of the latest C programming language standard [1] lists recommendations for implementations of complex arithmetic which fully respect the underlying IEEE floating-point arithmetic.

We explained how in interval arithmetic this leads to discussions about expressions where the result is either undefined or not exclusively real. In complex arithmetic conflicting results also come from the difficulty of representing and computing with the Riemann infinity. Support for projective infinity has mostly been dropped in floating-point implementations and hence implementations of complex arithmetic struggle in situations which require correct infinity arithmetic.

In order to salvage possibly incorrect complex results, the Annex G suggests to assign a double meaning to the IEEE NaN (Not-a-Number) when it occurs in complex expressions involving infinities: a NaN real or imaginary part indicates either an undefined value or an unknown value. This, however, does not fully take care of all problems, as the following example illustrates.

$$
\begin{aligned}
&\texttt{double } x, y \\
&\texttt{complex } z \\
&x = 2 \\
&y = (x^2 - 4)/(x - 2) \\
&z = 2^{1024} - 10^{340}\texttt{i} \\
&z = z \times (y + x\texttt{i})
\end{aligned}
$$

The correct mathematical result is $z = \texttt{undefined}$. The Annex G proposal however returns $z = \infty + \infty\texttt{i}$ because it fails to recognize y as mathematically undefined. When trying to rectify the result of $z \times (y + x\texttt{i})$ involving an infinite z, it jumps to the other interpretation of the NaN value y.

We shall indicate how this type of problems can be overcome by the introduction of yet additional special values. Our alternative approach gives rise to an efficient implementation which is fully compliant with the theory of complex analysis, as would be required in a proper computer algebra style implementation.

References

[1] ANSI/ISO/IEC 9899-1999. *International C Standard, Annex G: IEC 60599 – compatible complex arithmetic (informative)*, ANSI, 1999.

[2] T. Hickey, J. Qun, and M. Van Emden. "Interval arithmetic: from principles to implementation", *Journal of the ACM*, 2001, Vol. 48, No. 5, pp. 1038–1068.

[3] Sun Microsystems, *Interval arithmetic in the Forte[tm] C++ compiler*; available at http://www.sun.com/forte/cplusplus/interval.

[4] D. Ratz, *On extended interval arithmetic and inclusion isotonicity*, Technical Report, Institut für Angewandte Mathematik, Universität Karlsruhe, 1996.

[5] G. W. Walster, *The extended real interval system*, Technical Report, 1998, available at http://www.mscs.mu.edu/ globsol/readings.html.

An Improved Tool for Distribution Envelope Determination, a Technique for Interval-Based, Verified Arithmetic on Random Variables

Daniel Berleant, Lizhi Xie, Jianzhong Zhang,
and Gerry Sheblé
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011, USA
berleant@iastate.edu

1 Summary

When random variables possessing arbitrary distribution functions must be combined via +, −, *, /, min(), max(), etc., Monte Carlo simulation is commonly employed. However, Monte Carlo simulation assumes either independence or (less commonly) some other specific dependency relationship, among other limitations (Ferson 1996). Discretization of the distribution function followed by a numerical method is an alternative. Numerical methods can relax the requirement of Monte Carlo that the distributions have a known dependency relationship, in which case the results are typically envelope curves within which the cumulative distribution of the result must lie regardless of the dependency relationship between the operands. The operands themselves can also be expressed with envelopes in order to bound the effects of discretization of the input distributions (Berleant 1993; Williamson and Downs 1990). This paper describes Statool, a software tool that implements Distribution Envelope Determination (DEnv), a numerical algorithm for performing arithmetic on distribution function operands (Berleant and Goodman-Strauss 1998). Our previously reported tool was limited to independent random variables (Berleant and Cheng 1998), a significant limitation. Improvements to Statool are currently being driven by the needs of applications in accordance with our research strategy, which is to identify such applications and then to modify Statool as needed to support them. However, identifying good applications is itself a research topic. We are currently exploring applications to the electric power industry (Sheblé and Berleant 2002; Berleant et al. 2002), and have obtained recent results on time to completion of multiple tasks and time to failure of two components [7,8].

2 Introduction

Random variables may be combined using standard operations such as +,-,*, /, min(), and max(). When the random variable operands are assumed independent, results may be calculated using a discretized convolution approach (Ingram et al. 1968; Colombo and Jaarsma 1980; Kaplan 1981). Discretization error may be bounded by an interval based extension (Berleant 1993). We have described a tool implementing this (Berleant and Cheng 1998), however it is desirable though non-trivial to extend that work by eliminating the assumption that the random variables are independent, thereby handling the case where their dependency relationship is unknown and unspecified. In this case of unspecified dependency, obtaining bounded results requires that the entire range of possible dependency relationships be accounted for, including independence as one of the infinite number of possible dependencies. While the traditional approach of Monte Carlo simulation does not bound the range of results that are possible when dependency is unspecified (Ferson 1996), the desired bounds can be obtained with other techniques. A copula-based approach (Frank et al. 1987) which was significantly extended by Williamson and Downs (1990) and termed Probabilistic Arithmetic, has been implemented in a commercially available software system, RiskCalc (Ferson et al. 1998). DEnv (Distribution Envelope Determination) is described by Berleant and Goodman-Strauss (1998). A comparison of DEnv and Probabilistic Arithmetic reveals underlying similarities (Regan et al., submitted), as well as differences (Berleant and Goodman-Strauss 1998) that motivate its software implementation as well as continued development in other ways.

This paper reports a software implementation of DEnv (see Figures 1 and 2). This tool represents an advance over our previously developed tool, as described next.

- Calculation of $z = f(x, y)$ when x and y are not assumed independent (Berleant and Goodman-Strauss 1998) is now supported. The previously described tool assumes random variables are independent. The current tool bounds the range of results that are plausible when independence is not assumed. Figure 1 shows an example.

- Calculation of $max(x, y)$ and $min(x, y)$ for random variables x and y is now supported. This can be useful in problems like determining the time to complete two concurrent tasks, because the completion time of both is the same as the completion time of the task that finishes second, i.e., the maximum of the two individual completion times.

- Calculation of $z = f(x, y)$ in some instances where the interval expression for $f(x, y)$ leads to excess width is now supported. Although in DEnv x and y are probability distributions, DEnv reduces operations on distributions to operations on intervals, and the net effect of excess width in the interval calculated for $f(x, y)$, x and y intervals, is excessively wide envelopes derived for $f(x, y)$, where x and y are distributions. The tool

handles such expressions under the severe restriction that the function is monotonic over the box defined by the range over which distributions x and y are non-zero. While it would be desirable to incorporate more advanced techniques for reducing excess width for non-monotonic functions, even the current capability extends the state of the art for performing operations on distributions of unknown dependency, allowing evaluation of expressions such as that which produced Figure 2 without excess width in the envelopes because excess width is removed from the underlying interval evaluations of the expression.

- Calculation of cascaded operations is now supported. These are cases in which the result of one operation is used as an input to the next operation. The distributions used as inputs to an operation are discretized density functions, while the output of an operation consists of bounding envelopes which are cumulative distributions. Thus to use the output of an operation as the input to another operations requires converting a pair of bounding CDF envelopes into a discretized density function. We have done this by generalizing the histogram representation of an input to allow overlapping bars. This in turn enables conversion of the envelopes to the generalized histogram form, as will be described in the full paper. The generalized histogram form can then be used as an input to an operation the same way an ordinary histogram discretization of a density function can.

3 Algorithmic Issues

Calculation of results in the case of unspecified dependency between operands is based on a joint distribution tableau in which discretizations of each operand into intervals and associated probability masses form the marginals, and the interior cells are subject to constraints imposed by the marginals. Linear programming is called subject to these constraints, as a subroutine to find each desired point on the left and right envelopes. Only a limited number of points need to be found this way, because the discrete nature of the problem allows connecting the points safely to produce staircase-like envelopes in which each point is a bend in the staircase. While many details were covered in Berleant and Goodman-Strauss (1998), the linear programming aspects were not. Therefore we will review the DEnv algorithm in the full paper, emphasizing the linear programming aspects. Details on the algorithm as it applies to particular problems, including its linear programming aspects, may also be found in other works under review and available from the authors.

References

[1] S. Ferson, "What Monte Carlo Methods Cannot Do", *Human and Ecological Risk Assessment*, 1996, Vol. 2, pp. 990–1007.

[2] D. Berleant, "Automatically verified reasoning with both intervals and probability density functions", *Interval Comutations*, 1993, No. 2, pp. 48–70.

[3] D. Berleant and H. Cheng, "A software tool for automaticaly verified reasoning on intervals and probability distributions", *Reliable Computing*, 1998, Vol. 4, No. 1, pp. 71–82.

[4] D. Berleant and C. Goodman-Strauss, "Bounding the results of arithmetic operations on random variables of unknown dependency using intervals", *Reliable Computing*, 1998, Vol. 4, No. 2, pp. 147–165.

[5] D. Berleant, J. Zhang, R. Hu, and G. Sheblé, "Economic dispatch: applying the interval-based Distribution Envelope algorithm to an Electric Power Problem", *Validated Computing 2002*, May 23–25, Toronto.

[6] G. Sheblé and D. Berleant, "Bounding the composite value at risk for energy service company operation with DEnv, an interval-based algorithm", *Validated Computing 2002*, May 23–25, Toronto.

[7] D. Berleant, J. Zhang, and G. Sheblé, *On completion times of networks of concurrent and sequential tasks*, 2001; submitted 6/01.

[8] D. Berleant, J. Zhang, and G. Sheblé, *On bounding failure times in two-component systems*, in preparation; current manuscript available upon request.

[9] A. G. Colombo and R. J. Jaarsma, "A powerful numerical method to combine random variables", *IEEE Transactions on Reliability*, 1980, Vol. R-29, No. 2, pp. 126–129.

[10] R. A. Evans, "Bayes Paradox", *IEEE Transactions on Reliability*, 1982, Vol. R-31, No. 4, p. 321.

[11] G. E. Ingram, E. L. Welker, and C. R. Herrmann, "Designing for reliability based on probabilistic modeling using remote access computer systems", *Proc. 7th Reliability and Maintainability Conference*, American Society of Mechanical Engineers, 1968, pp. 492–500.

[12] S. Kaplan, "On the Method of Discrete Probability Distributions in Risk and Reliability Calculations, Applications to Seismic Risk Assessment", *Risk Analysis*, 1981, Vol. 1, No. 3, pp. 189–196.

[13] H. Regan, S. Ferson, and D. Berleant, *Equivalence of five methods for bounding uncertainty*, submitted.

[14] R. Williamson and T. Downs, "Probabilistic arithmetic I: numerical methods for calculating convolutions and dependency bounds", *International Journal of Approximate Reasoning*, 1990, Vol. 4, pp. 89–158.

Figure 1: Two normal distributions each with $\mu = 1$ and $\sigma = 1$ were tail-trimmed to within $[-3, 5]$ (because the tool is currently limited to numerically valued bounds). These distributions were used as input variables. Given no assumptions about their dependency relationship, staircase-shaped left and right envelopes were computed which enclose the space within which the distribution of (a sufficiently large number of) products of samples of the inputs must travel regardless of their dependency relationship. There are also three smoother curves showing the product distributions for three particular dependency relationships that allow the curves to be computed relatively easily. One of these is for independent inputs, and was computed using the Monte Carlo-generated products of 100,000 samples of the inputs. The other two are analytically derived distributions of the product assuming Pearson correlations of 1 and -1.

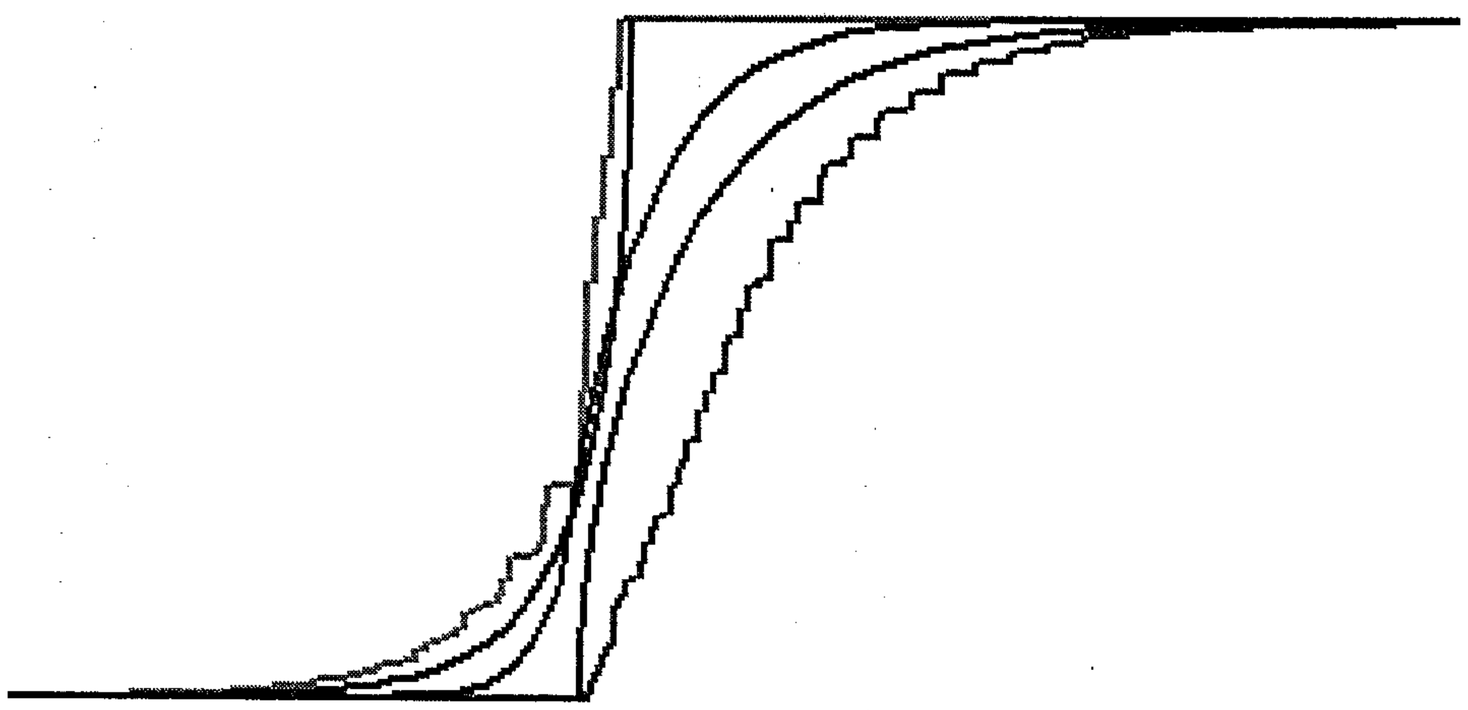


Figure 1:

Figure 2 follows on next page: X and Y are inputs. Z constitutes envelopes around the result when the dependency relationship between X and Y is unspecified, and $Z = (38 * Y - 8 * X)/(0.08 * Y + 0.048 * X)$. The cumulative forms of histogram discretizations of PDFs (X and Y) are pairs of CDF bounds that each look like two staircases in which the top bends of the lower curve touch the bottom bends of the upper curve. The cumulative form of the result does not in general obey that constraint, and hence cannot in general be displayed correctly as a histogram. It can be displayed correctly in cumulative form, as shown in the lower subwindow.

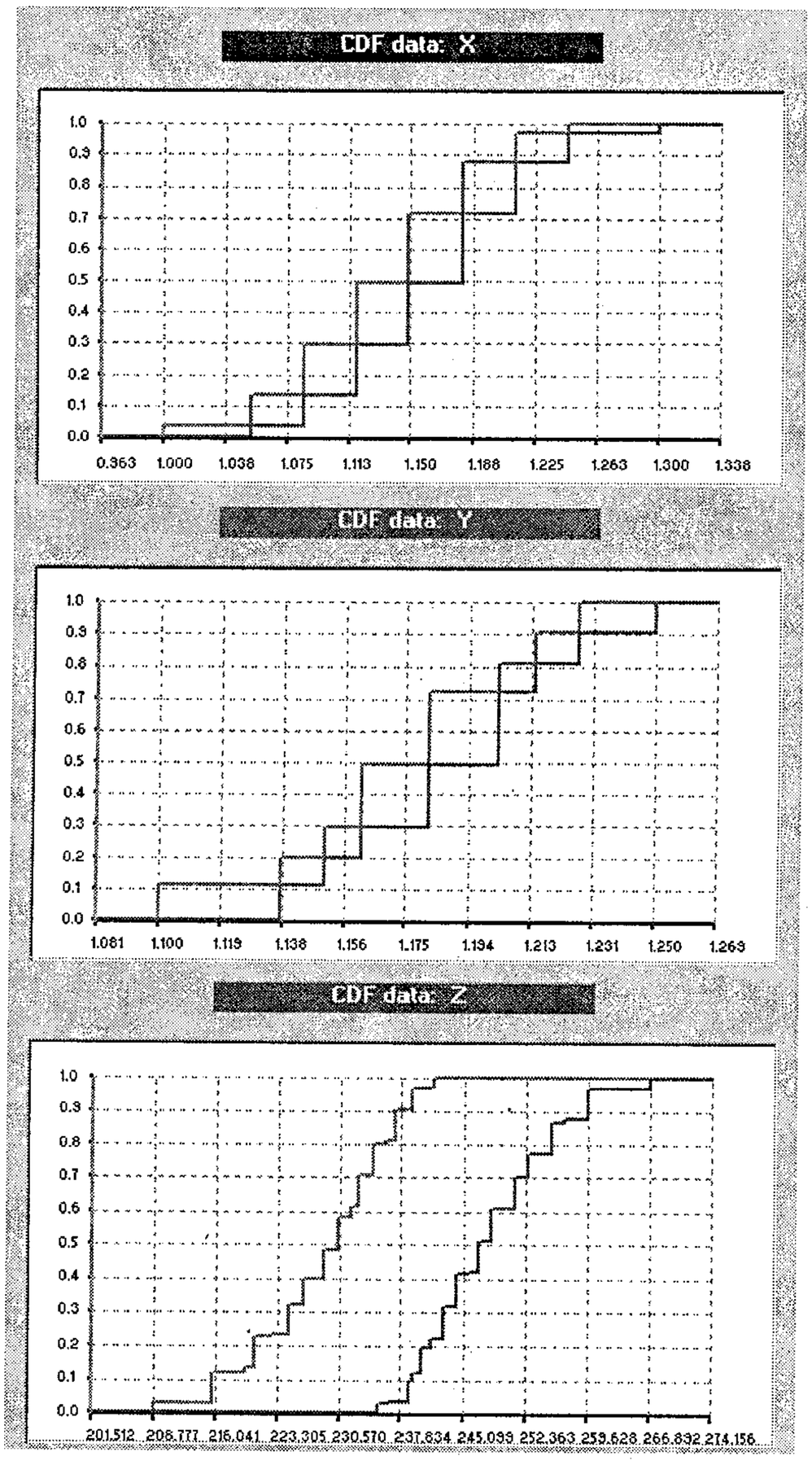

CDF data: X
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.363 1.000 1.038 1.075 1.113 1.150 1.188 1.225 1.263 1.300 1.338
CDF data: Y
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1.081 1.100 1.119 1.138 1.156 1.175 1.194 1.213 1.231 1.250 1.269
CDF data: Z
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
201.512 208.777 216.041 223.305 230.570 237.834 245.099 252.363 259.628 266.892 274.156

Figure 2:

Economic Dispatch: Applying Interval-Based Dependency Analysis to an Electric Power Problem

Daniel Berleant, Jianzhong Zhang, Rujun Hu,
and Gerald Sheblé
Dept. of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50014, USA

1 Summary

A common way to model uncertainty in the value of a quantity is to use a probability density function (PDF) or its integral, a probability distribution function (CDF). When two such values are combined to form a new value equal to their sum, product, max, etc., the new value is termed a *derived distribution*[5]. It is well-known that derived distributions may be obtained by numerical convolution, Monte Carlo simulation, and analytically for specific classes of input distributions, under the assumption that the input distributions are independent. It is also possible to obtain derived distributions for specified dependency relationships other than independence. However, it is not always the case that the dependency relationship is known. Thus there is a need for obtaining solutions without assuming independence or any other specific dependency relationship. There are two numerical algorithms that have been implemented in software for this. Numerical approaches have the advantage of applicability to a very wide class of distributions. Probabilistic Arithmetic [6] is implemented in the commercially available software tool RiskCalc [3]. Interval-Based Dependency Analysis (IBDA) [2], which extends our previous tool [1] by eliminating the independence assumption, is implemented in the software tool Statool and is available upon request from the authors. While the two tools have fundamental similarities [4], a significant difference with respect to the present problem is that IBDA supports, and Statool implements, excess width removal in the underlying interval calculations, from some expressions. In this paper we apply IBDA to generalize a solution to the well-known *economic dispatch* problem in electric power generation to the case where the dependency relationship between the fuel costs of two generators is unspecified.

2 The Problem

The economic dispatch problem in electric power generation may be stated as follows. It is desired to determine how much power should be generated by each of two generators to meet a given level of demand so that total generation cost is minimized. One of a number of approaches to solving this problem is termed LaGrangian Relaxation [7]. We incorporate uncertainty into the LaGrangian Relaxation technique for solving the sample problem by modeling uncertainty in the cost of fuel to run the generators with probability distributions, postulating in addition that the dependency between the two fuel costs of the two generators is unknown (as would occur if one generator burns oil and the other coal). The uncertainties are then propagated through the algebraic expression derived by the LaGrangian Relaxation technique.

First, we specify the cost equations as

$$F_1 = v_1(8P_1 + 0.024P_1^2 + 80), \quad F_2 = v_2(6P_2 + 0.04P_2^2 + 120),$$

where P_1 and P_2 are the power outputs of generators 1 and 2 in megawatts; v_1 and v_2 are the fuel costs for generators 1 and 2 in \$ per M Btu; and F_1 and F_2 are the generation costs for given power output levels and fuel cost rates. Therefore generation costs change nonlinearly with power output according to the following equations.

$$\frac{dF_1}{dP_1} = v_1(8 + 0.048P_1), \quad \frac{dF_2}{dP_2} = v_2(6 + 0.08P_2). \tag{1}$$

Solving the problem requires minimizing an objective function

$$F = F_1 + F_2 = v_1(8P_1 + 0.024P_1^2 + 80) + v_2(6P_2 + 0.04P_2^2 + 120),$$

subject to the constraint $P = P_1 + P_2$ where P is the total customer demand for electric power which for this example we take as 400 megawatts. This gives a constraint function

$$P = P_1 + P_2 = 400. \tag{2}$$

By the method of Lagrangian multipliers from calculus, at an extreme value of this objective function,

$$\frac{dF_1}{dP_1} = \frac{dF_2}{dP_2} = \lambda \tag{3}$$

for some λ. This is derived from the Lagrange function L which relates objective function F and constraint (1) according to $L = F + \lambda \cdot P$, which implies

$$\frac{\partial L}{\partial P_1} = \frac{dF_1(P_1)}{dP_1} - \lambda = 0$$

for generator 1 and similarly for generator 2.

From (1) and (3),

$$v_1(8 + 0.048P_1) = \lambda = v_2(6 + 0.08P_2), \quad P_2 = 400 - P_1,$$

and solving simultaneous equations for P_1 gives

$$P_1 = \frac{38v_2 - 8v_1}{0.08v_2 + 0.048v_1}, \quad P_2 = 400 - P_1, \tag{4}$$

as the most economical amounts of power to generate from generators 1 and 2 to meet the demand (assuming those amounts are within the capacity of both generators). P_1 and P_2 are easily calculated for real values of v_1 and v_2, but given distribution functions for v_1 and v_2, the problem requires evaluating an expression on random variables v_1 and v_2 involving a sum, difference and quotient. Solving it by dividing a difference of random variables by a sum results in excessively wide envelopes on the CDFs for P_1 and P_2 because the same operands occur in both terms, leading to excess width in the underlying interval calculations. Instead the entire expression must be treated as a single binary operation on v_1 and v_2. Figure 1 shows the results given PDFs describing v_1 and v_2.

3 Discussion and Conclusion

Statool currently has certain limitations. Planned extensions include the following.

1. Asymptotic pdf tails. The process of discretizing a pdf into a histogram does not presently allow for the case where a pdf tail trails off to plus or minus infinity. Yet this implies setting definite bounds, though any specific such bounds might be hard to justify. Indeed unusual and extreme values can occur in the electric power domain, as happened for example in the California power crises recently. The solution is to allow the discretization to include open intervals with an end point at ∞ or $-\infty$. This in turn would require the arithmetic operations to be defined on such intervals. Fortunately this is possible, e.g., $[1, \infty) + [1, 2] = [2, \infty), (-\infty, -1] * [-2, -1] = [1, \infty), [1, 2]/[-1, 1] = (-\infty, \infty)$, etc.

2. Partial dependency. While the system currently can calculate either under the assumption of independence, or with no assumption about dependency, partial information about dependency is often present in real problems. Correlation values are a typical example. An example would be prices of different fuels, for which one would expect a generally positive correlation.

In the full paper we will explain the IBDA algorithm, and also include explanations and figures, showing how assuming independence results in stronger results, while excess width in interval evaluation of equation (4) leads to weaker results. We will also remark on the implications of the CDF bounds to decision-makers.

Figure 1: Solution for P_1 of equation (4), given the histogram-discretized PDFs for v_1 and v_2 shown. The CDF for optimum power generation from generator 1 will be within the envelopes shown regardless of the dependency

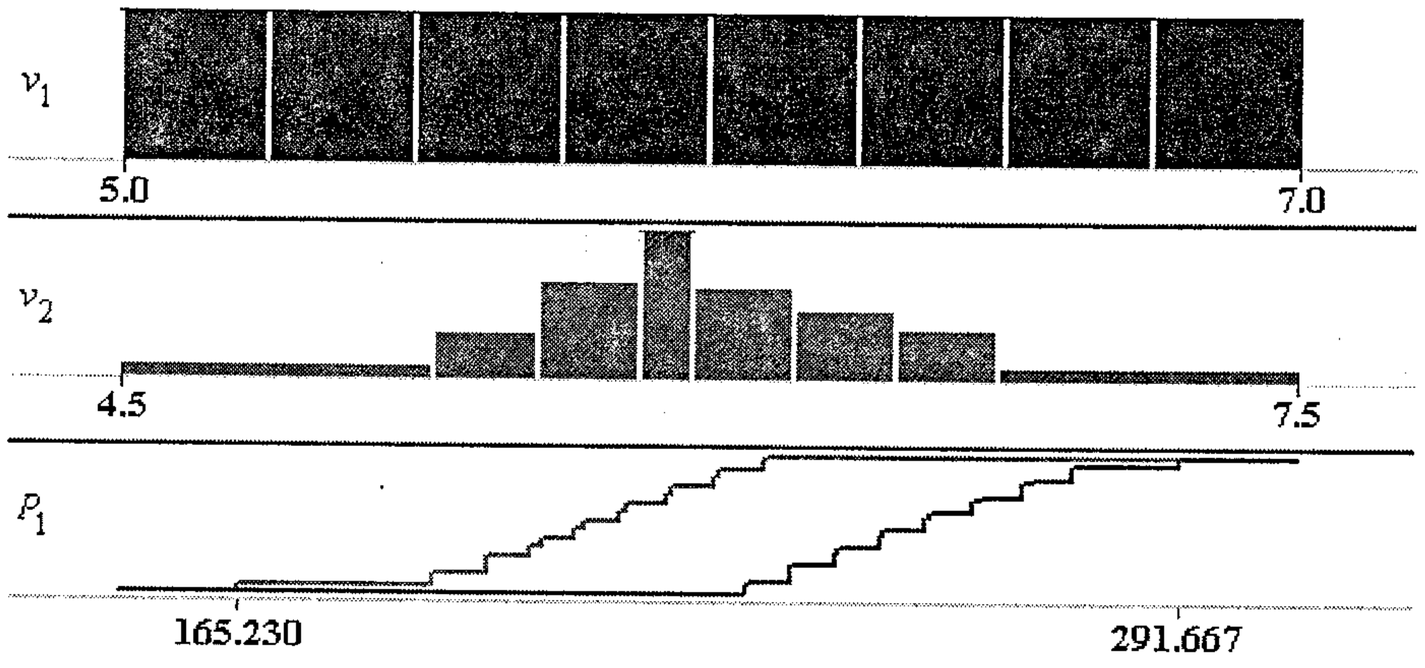


Figure 1:

relationship between inputs v_1 and v_2. The envelopes might be sufficient for a decision, or might point out the need for additional information gathering to sharpen the input distributions and/or identify their dependency relationship sufficiently to support a decision.

References

[1] D. Berleant and H. Cheng, "A Software Tool for Automatically Verified Operations on Intervals and Probability Distributions", *Reliable Computing*, 1998, Vol. 4, No. 1.

[2] D. Berleant and C. Goodman-Strauss, "Bounding the Results of Arithmetic Operations on Random Variables of Unknown Dependency using Intervals", *Reliable Computing*, 1998, Vol. 4, No. 2, pp. 147–165.

[3] S. Ferson, W. T. Root, and R. Kuhn, *RAMAS Risk Calc: Risk Assessment with Uncertain Numbers*, Applied Biomathematics, Setauket, New York, 1998. See also http://www.ramas.com/riskcalc.htm.

[4] H. Regan, S. Ferson, and D. Berleant, *Equivalence of Five Methods for Bounding Uncertainty*, submitted.

[5] M. D. Springer,. *The Algebra of Random Variables*, Wiley, 1979.

[6] R. Williamson and T. Downs, "Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds", *International Journal of Approximate Reasoning*, 1990, Vol. 4, pp. 89–158.

[7] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed., Wiley, 1996.

Validated Integration of Asteroid Orbits

Martin Berz[1] and Kyoko Makino[2]

[1]Department of Physics and Astronomy, and
National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing, MI 48824, USA, berz@msu.edu
[2]Department of Physics,
University of Illinois at Urbana-Champaign,
Urbane, IL 61801-3080, USA, makino@uiuc.edu

Dedicated to Ramon Moore, who started it all

From the earliest days of self-validating methods, the integration of orbits in the solar system has ranked high on the interest of researchers; in fact the early work of Ramon Moore in this area presented one of the first practical applications of self-validated and interval methods. In this talk we focus on recent developments in this direction, specifically the integration of near-earth asteroids and the exclusion of collisions with earth. We review both previous work and show various new improvements to the performance of the technique.

The approach is based on the Taylor model approach, which allows for a description of functional dependencies without some of the limitations of direct interval methods. We review both the Taylor model approach as well as the basic ideas of the verified integrator VI. First, the method has a sharpness that scales with a high order of the domain size. Furthermore, it largely avoids the so-called dependency problem that arises in the validated evaluation of extended calculations. This is done by explictly describing dependencies on initial conditions as a high-order Taylor polynomial and a small interval remainder at any time step. Various practical examples related to this question are given. Finally, the method scales very favorably to higher dimensions and can alleviate the so-called dimensional curse. We present a few examples of the methods for validated optimization and quadrature. We also discuss the various platforms through which the methods can be used, including the C++, F90, and COSY interfaces.

When applied to the verified integration of ODEs, the method successfully avoids the so-called wrapping effect problem, which is in essence a manifestation of the dependency problem that unavoidably occurs in the very extended

functional dependence of final conditions on initial conditions as prescribed by the step-by-step integration scheme. It also allows a simple combination of the frequently separated steps of finding an a priori inclusion and then subsequently performing a more sophisticated actual step. Many of the details of the methods are discussed in a talk by Kyoko Makino at this meeting.

For the purpose of integration of asteroid orbits, a far-reaching control of the wrapping effect is crucial because of the unavoidably rather large initial domains to which the dynamical quantities are known. We discuss in detail the performance of the Taylor-model based integrators for a full model of dynamics in the solar system, including relativistic and other corrections that leads to an accuracy within the kilometer domain for typical integrations. The validated methods allow a control of the overestimation to a few percent over integration periods of up to 100 years, which is sufficient for detailed study of possible collisions of near-earth asteroids with earth.

Computing Tight Bounds for the L_1-Norm of Peano Kernels

Chin-Yun Chen
Department of Applied Mathematics
National Chiayi University, Taiwan
email: cychen@mail.ncyu.edu.tw

The purpose of this paper is to point out essential consideration for computing tight bounds for the error constants required in the error estimation of a numerical quadrature rule.

Due to Peano, cf. [2, 6], the quadrature error $E(f) := I(f) - S(f)$ resulting from approximation to the definite integral $I(f) := \int_a^b w(x)\, f(x)dx$ by a quadrature rule $S(f) := \sum_{i=1}^{M} w_i\, f(x_i)$ of degree d, where $w_i > 0$, can be represented as

$$E(f) = \int_a^b K_r(t)\, f^{(r)}(t)\, dt\ , \tag{1}$$

where $f \in C^r[a,b]$, $1 \le r \le d+1$ and the Peano kernel $K_r(t)$ is defined by

$$K_r(t)\ :=\ E_x\left((x-t)_+^{\langle r-1\rangle}\right)$$

with $(x-t)^{\langle r\rangle} := \dfrac{(x-t)^r}{r!}$ and

$$(x-t)_+^r := \begin{cases} (x-t)^r\ , & \text{for}\ \ x \ge t\ , \\ 0\ , & \text{for}\ \ x < t\ . \end{cases}$$

Generally, the quadrature error (1) is estimated according to

$$|E(f)| \le \|f^{(r)}\|_\infty \cdot \int_a^b |K_r(t)|\, dt\ , \tag{2}$$

or for validated computation

$$E(f)\quad \in\quad f^{(r)}([a,b]) \cdot \int_a^b K_r^+(t)\, dt\ -\ f^{(r)}([a,b]) \cdot \int_a^b K_r^-(t)\, dt\ , \tag{3}$$

where $K^+(t) := \max(K(t),\, 0)$, $K^-(t) := \max(-K(t),\, 0)$.

If $K_r(t)$ is definite on $[a,b]$, i. e. $K_r(t)$ does not change sign on $[a,b]$, then in (2), (3) there remains the integral $\int_a^b K_r(t)\,dt$ to be computed, which might not be a difficult task, at least not for $w(x) \equiv 1$. However, this is not the usual case. According to (1), we have

$$E(x^r) = r! \cdot \int_a^b K_r(t)\,dt\,, \qquad 1 \le r \le d+1\,. \tag{4}$$

It follows immediately that $\int_a^b K_r(t)\,dt = 0$ for $1 \le r \le d$. This implies that only the Peano kernels of highest orders may be definite. If $K_{d+1}(t)$ is definite on $[a,b]$, then $\int_a^b K_r(t)\,dt$ is equal to $E(x^{d+1})/(d+1)!$, which can be easily computed. On the other hand, if $K_{d+1}(t)$ changes its sign on $[a,b]$ and/or f is not sufficiently smooth on $[a,b]$, i. e. $f \in C^r[a,b]$, $1 \le r \le d+1$, then for the error estimation (2) and (3) the error constants $\int_a^b K_r^+(t)\,dt$ and $\int_a^b K_r^-(t)\,dt$ have to be known, where $1 \le r \le d+1$.

A straightforward method for computing the error constant $\int_a^b |K_r(t)|\,dt$ is to compute all the zeros of $K_r(t)$ at first, then to integrate $K_r(t)\Big|_{[x_i,x_{i+1}]}$ between every two adjacent zeros within each subinterval $[x_i, x_{i+1}]$, cf. [11, 9, 10]. Since it is not easy to identify all the zeros of Peano kernels numerically, hence, in [11] the method was only applied to K_r for $r = 1, 2$, in [9] the method was only applied to K_d by analytically confirming that for Gauss-Legendre rules K_d possesses only the zero 0. In [10] the first trial for the whole range $1 \le r \le d+1$ was undertaken and interval computations was used. However, the numerical results presented in [11, 9, 10] reveal themselves to be validated or improved. In the literature there was also much effort given for estimating an upper bound for $\int_a^b |K_r(t)|\,dt$, cf. [2, 5, 3, 4]. Among them, good results in general can only be obtained for $r = d+1$. For $1 \le r \le d$, the most suggested upper bounds are relatively coarse. This paper adopts the same method used in [11, 9, 10] and has successfully gained significant improvement in the computational quality for the whole range $1 \le r \le d+1$.

In the presentation, essential consideration for doing the computation, the algorithms as well as numerical results with comparison to published bounds are proposed. All the ideas presented in this paper can also be applied to one-dimensional Sard kernels appearing in the error representation of a numerical cubature rule.

References

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computation*, Academic Press, New York, 1983.

[2] H. Brass, *Quadraturverfahren*, Vandenhoeck & Ruprecht, Goettingen, 1st ed., 1977.

[3] H. Brass, "Error Bounds based on Approximation Theory", In: T. O. Espelid and A. Genz (eds.), *Numerical Integration*, Kluwer Academic Publishers, 1991, pp. 147–163.

[4] H. Brass, "Error Estimation for the Clenshaw-Curtis Quadrature Method", In: U. Wannagat (ed.), *Abh. Braunschw. Wiss. Ges.*, Verlag Erich Goltze, 1992, p. 43–53.

[5] H. Brass and K.-J. Foerster, "On the estimation of linear functionals", *Analysis*, 1987, Vol. 7, pp. 237–258.

[6] P.-J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 2nd ed., 1984.

[7] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz, *Numerical Toolbox for Verified Computing I*, Springer, Berlin, 1993.

[8] R. Klatte, U. Kulisch, M. Neaga, D. Ratz, and Ch. Ullrich, *PASCAL-XSC*, Springer-Verlag, 1991.

[9] F.-G. Lether, "Cubature error bounds for Gauss-Legendre product rules", *SIAM. J. Numer. Anal.*, 1971, Vol. 8, No. 1, pp. 36–42.

[10] U. Storck, *Verifizierte Berechnung mehrfach geschachtelter singulaerer Integrale der Gaskinetik*, Ph.D. thesis, Universitaet Karlsruhe, 1995.

[11] A.-H. Stroud, and D. Secret, *Gaussian Quadrature Formulas*, Pretince Hall, Englewood Cliffs, New Jersey, 1966.

Towards Validated Global Optimal Control

Bruce Christianson
Numerical Optimisation Centre
University of Hertfordshire
Hatfield, England, Europe

Consider a discrete-time optimal control problem in the following *direct* form: choose control values $u_i \in R^p$ for each timestep $0 \leq i < N$ so as to minimize $z = F(x_N)$ where x_0 is some fixed constant and the state equation is $x_{i+1} = f_i(x_i, u_i)$ for $0 \leq i < N$. Here each f_i is a smooth map from $R^q \times R^p \to R^q$ and F is a smooth map from R^q to R. The dimension of u_i may depend upon the timestep i, but for notational convenience we omit this refinement.

We lose nothing by restricting attention to target functions of this form: the more usual formulation where z has the form $z = \sum_{i=0}^{N-1} F_i(x_i, u_i) + F_N(x_N)$, can be reduced to the form $z = F(x_N)$ by augmenting each state x_i with a new component $v_i \in R$ defined by $v_0 = 0; v_{i+1} = v_i + F_i(x_i, u_i)$ and defining $F(x_N, v_N) = v_N + F_N(x_N)$.

In the direct formulation the Np independent variables are the controls u_i : $0 \leq i < N$. Typically the number of timesteps, and hence the number of independent variables is very large (millions). This makes the validated solution of such problems difficult.

In the alternative *indirect* formulation, the only independent variables are the q components of an initial *costate* $\tilde{x}_0$. At each timestep i, the current controls u_i and the successor costate $\tilde{x}_{i+1}$ are implicitly defined, in terms of the current state x_i and current costate $\tilde{x}_i$, by the costate equations and the Pontryagin equations

$$\tilde{x}_i - [f'_{x,i}(x_i, u_i)]^T \tilde{x}_{i+1} = 0; \quad [f'_{u,i}(x_i, u_i)]^T \tilde{x}_{i+1} = 0.$$

The state equation $x_{i+1} = f_i(x_i, u_i)$ then gives x_{i+1} in terms of x_i and u_i. In the indirect formulation, the requirement for the path to be optimal is that $\tilde{x}_N - F'(x_N) = 0$ which we observe can be regarded as a form of the transversality condition. For an optimal path (although not in general) the numerical values of the costates $\tilde{x}_i$ are equal to those of the *adjoint* states $\bar{x}_i = \partial z / \partial x_i$.

For a non-optimal path, the residual value $r = \tilde{x}_N - F'(x_N)$ of the transversality equation gives a measure of how far the initial costate value $\tilde{x}_0$ differs from that for the optimum path. An important advantage of the indirect approach

for validated methods is the drastic reduction in the number of independent variables, from Np to q.

In 1983 Pantoja described a computationally efficient stagewise construction of the Newton direction for the direct formulation. Recently [6] we formulated an indirect analogue of Pantoja's algorithm, which gives exactly the Newton step a_0 for the initial costate with respect to a terminal transversality condition. We believe this indirect reformulation of Pantoja's algorithm potentially forms a useful tool for attacking the problem of verified global optimal control using interval methods.

We conclude this abstract by giving a scalar (non-interval) form of the indirect Pantoja algorithm, and then indicate some of the possibilities.

Step 1. Given the fixed initial value for x_0, set a trial initial value for $\tilde{x}_0$. For i from 0 up to $N-1$ calculate $u_i \in R^p; \tilde{x}_{i+1}, x_{i+1} \in R^q$ by solving the implicit costate and Pontryagin equations, respectively

$$\tilde{x}_i - \left[f'_{x,i}\right]^T \tilde{x}_{i+1} = 0; \qquad \tilde{u}_i = \left[f'_{u,i}\right]^T \tilde{x}_{i+1} = 0,$$

for u_i and $\tilde{x}_{i+1}$ and setting $x_{i+1} = f_i(x_i, u_i)$.

Step 2. Set $z = F(x_N)$, and define $a_N \in R^q, D_N \in R^{q\times q}$ by

$$D_N = F''(x_N); \qquad a_N = -r \quad \text{where } r = \tilde{x}_N - F'(x_N).$$

Step 3. For i from $N-1$ down to 0 calculate $a_i \in R^q; A_i, D_i \in R^{q\times q}; B_i \in R^{p\times q}; C_i \in R^{p\times p}$ by

$$A_i = \left[f'_{x,i}\right]^T D_{i+1} \left[f'_{x,i}\right] + (\tilde{x}_{i+1})^T \left[f''_{xx,i}\right]$$

$$B_i = \left[f'_{u,i}\right]^T D_{i+1} \left[f'_{x,i}\right] + (\tilde{x}_{i+1})^T \left[f''_{ux,i}\right]$$

$$C_i = \left[f'_{u,i}\right]^T D_{i+1} \left[f'_{u,i}\right] + (\tilde{x}_{i+1})^T \left[f''_{uu,i}\right]$$

where [.] denotes evaluation at (x_i, u_i), and we write (for example)

$$\left(\left[f'_{u,i}\right]^T D_{i+1} \left[f'_{x,i}\right]\right)_{j,k} \text{ for } \sum_{l=1}^{q}\sum_{m=1}^{q} \left[\frac{\partial (x_{i+1})_l}{\partial (u_i)_j}\right] (D_{i+1})_{l,m} \left[\frac{\partial (x_{i+1})_m}{\partial (x_i)_k}\right] \text{ etc.}$$

If C_i is singular then the algorithm fails, otherwise set

$$D_i = A_i - B_i^T C_i^{-1} B_i$$

$$a_i = \left[f'_{x,i}\right]^T a_{i+1} - B_i^T C_i^{-1} \left[f'_{u,i}\right]^T a_{i+1}$$

and **STOP**.

Either the algorithm fails to terminate, or else at the end a_0 satisfies

$$\tilde{x}_N - F'(x_N) + a_0 \cdot \frac{\partial}{\partial \tilde{x}_0}(\tilde{x}_N - F'(x_N)) = 0.$$

Since in the region of an optimum path we have that all the C_i are positive definite [6], the indirect algorithm can be combined with a variational analysis to provide a largeish box around the (believed) global optimum for $\tilde{x}_0$ in which interval Newton establishes that only one solution to the transversality equation exists.

For example if we set r to be a cartesian basis vector, then the algorithm gives the corresponding row of $A = J^{-1}$, where $J = \partial r/\partial \tilde{x}_0$ for the midpoint of the box $[\tilde{x}_0] = \tilde{x}_0 + \Delta$. Using Automatic Differentiation techniques [1] we can differentiate through the costate and Pontryagin equations to evaluate ranges for the derivatives $B = [dr/d\tilde{x}_0]$. Then any optimal point in $[\tilde{x}_0]$ is also in $\phi([\tilde{x}_0] = \tilde{x}_0 + a_0 + \Delta[I - AB]$.

This should significantly ease the task of proving other boxes to either contain no solution to the transversality equations, or to be suboptimal.

We stress that the outline given here is very simplistic (it is assumed that all state and control constraints have been incorporated into the target function by penalty terms, for instance) and that much work remains to be done before global optima for control problems can be validated rigorously in a reasonable time. Nevertheless we believe that the approach set out here is a viable manifesto for a programme to achieve this.

References

[1] M. Bartholomew-Biggs, "Automatic Differentiation of Implicit Functions using Forward Accumulation", *Computational Optimization and Applications*, 1998, Vol. 9, pp. 65–84.

[2] M. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon, "Automatic Differentiation of Algorithms", *Journal of Computational and Applied Mathematics*, 2000, Vol. 124, pp. 171–190.

[3] B. Christianson, "Reverse Accumulation and Implicit Functions", *Optimization Methods and Software*, 1998, Vol. 9, pp. 307–322.

[4] B. Christianson, "Cheap Newton Steps for Optimal Control Problems: Automatic Differentiation and Pantoja's Algorithm", *Optimization Methods and Software*, 1999, Vol. 10, pp. 729–743.

[5] B. Christianson and M. Bartholomew-Biggs, "Globalization of Pantoja's Optimal Control Algorithm", In: *From Simulation to Optimization: 3rd International Conference on Automatic Differentiation*, Springer LNCS, 2001 (to appear).

[6] B. Christianson, "A Self-Stabilizing Pantoja-Like Indirect Algorithm for Optimal Control", *Optimization Methods and Software*, 2001, Vol. 16, pp. 131–149.

[7] T. Coleman and A. Liao, "An Efficient Trust Region Method for Unconstrained Discrete-Time Optimal Control Problems", *Computational Optimization and Applications*, 1995, Vol. 4, pp. 47–66.

[8] D. Conforti and M. Mancini, "A Curvilinear Search Algorithm for Unconstrained Optimization", *Optimization Methods and Software*, 2001 (to appear).

[9] L. Dixon and M. Bartholomew-Biggs, "Adjoint-Control Transformations for Solving Practical Optimal Control Problems", *Optimal Control Applications and Methods*, 1981, Vol. 2, pp. 365–381.

[10] H. Kagiwada *et al*, *Numerical Derivatives and Nonlinear Analysis*, Plenum, New York, 1986.

[11] J. F. A. De O. Pantoja, *Algorithms for Constrained Optimization Problems*, Ph.D. thesis, Imperial College of Science and Technology, University of London, 1983.

[12] J. F. A. De O. Pantoja, "Differential Dynamic Programming and Newton's Method", *Int J Control*, 1998, Vol. 47, No. 5, pp. 1539–1553.

Computing with Sets of Probability Measures

Fabio Gagliardi Cozman[1], José Carlos Ferreira da Rocha[2],
and Cassio Polpo de Campos[3]

[1]Escola Politécnica, Univ. de São Paulo
São Paulo, SP – Brazil
[2]Univ. Estadual de Ponta Grossa
Ponta Grossa, PR – Brazil
[3]Pontifícia Universidade Católica
São Paulo, SP – Brazil

The computational manipulation of probability measures often requires the treatment of interval values, not only due to numerical errors, but also due to more fundamental difficulties: we may want to model imprecise beliefs; we may have incomplete knowledge about probability values; we may be interested in merging beliefs from groups of experts; and we may wish to verify the effect of perturbations in probabilistic models [2, 17]. Such difficulties have often led to the study of interval probability and related theories. The goal of this paper is to present a brief overview of methods and results that can be relevant to the validated manipulation of probabilistic models.

The most general representation for imprecision in probabilistic models seems to be provided by the theory of sets of probabilities (called *credal sets* [13]). In this work we focus on closed convex credal sets; there are axiomatic derivations of such credal sets and other variants [12, 16, 17]. Consider two examples. First, consider a binary variable X and the set of measures defined by the interval $P(X = x_0) \in [0.3, 0.4]$, where $P(X = x_0)$ is the probability of the event $\{X = x_0\}$ — here a single interval can define the entire credal set. Second, consider a variable Y that can take three values, $\{y_0, y_1, y_2\}$. A probability distribution for Y is entirely defined by a three-valued vector $\{p_0, p_1, p_2\}$ such that $p_i \geq 0$ and $\sum_i p_i = 1$. We can build a credal set by taking a distribution $p(Y)$ and considering the set of all distributions $r(Y)$ such that the difference $|R(A) - P(A)|$ is always smaller than some positive ϵ for any event A (where $R(\cdot)$ is the measure induced by $r(Y)$ and $P(\cdot)$ is the measure induced by $p(Y)$). This type of credal set is called a *total variation* neighborhood in robust statistics [10].

Given a credal set $Q(X)$, we can obtain *upper expectations* for any bounded function: $\overline{E}[f(X)] = \max_{P \in Q} E_P[f(X)]$. Likewise, we can define *lower expectations*: $\underline{E}[f(X)] = \min_{P \in Q} E_P[f(X)]$. Lower and upper expectations define

expectations intervals, and the theory of credal sets can be viewed as a theory that manipulates expectation intervals in a principled manner. We assume discrete models in this paper, noting that an assessment of the form $\overline{E}[f(X)] = \gamma$ is equivalent to a linear inequality $\sum_X f(x)p(x) \leq \gamma$.

Conditioning is generally taken to mean elementwise application of Bayes rule; the *conditional* credal set $Q(X|Y)$ is obtained by applying Bayes rule to each element of the *joint* credal set $Q(X,Y)$ [8, 13].

Consider first the computation of upper expectations $\overline{E}[f(X)]$ with respect to credal sets specified by linear constraints. We obtain a linear program with analysis going back to the work of Boole and with extensions based on column-generated methods, as reviewed by Hansen et al [9].

A more interesting challenge is the computation of upper posterior expectations $\overline{E}[f(X)|Y]$. Still assuming linear constraints, we now have a linear fractional optimization problem [15]. The most efficient method to deal with these problems seems to be the Charnes–Cooper transformation, which reduces the fractional problem to a linear program [11, 14]. Other methods, such as Walley's iterative scheme and the Dinkelbach-Jagannatham algorithm (known in statistics as Lavine's method) can be of value in specific cases [6].

An important situation in practice is the computation of $\overline{E}[f(X)|Y]$ with respect to a credal set $Q(X)$ and a collection of "likelihood" credal sets $Q(Y|X = x)$, for all values of X. Surprisingly, we can still reduce this problem to a linear program with some mild assumptions on the sets $Q(Y|X = x)$, using an algorithm presented in [6].

We now consider the impact of independence relations. The first difficulty is that there are several definitions of independence for credal sets [5].

One possible definition (*epistemic independence*) states that variables X and Y are independent when $\overline{E}[f(X)|Y] = \overline{E}[f(X)]$ and $\overline{E}[g(Y)|X] = \overline{E}[g(Y)]$ for any bounded functions $f(X)$ and $g(Y)$. Algorithms for inference in multivariate models based on epistemic independence are presented in [7], but their computational complexity seems to be quite high. A simple Markov chain as $W \rightarrow X \rightarrow Y \rightarrow Z$, where all variables are binary, all probabilities are defined by intervals, and each variable is epistemically independent of all ascendants given the direct ascendant, defines a credal set $Q(W,X,Y,Z)$ with more than 6 million extreme points!

A second possible definition for independence (*strong independence*) requires that any extreme point of $Q(X,Y)$ satisfies $p(X|Y) = p(X)$ and $p(Y|X) = p(Y)$. Computation of upper posterior expectations is now a multilinear program with many possible local maxima. There has been great effort to solve such programs when multivariate models are represented by directed graphs (following the successful theory of Bayesian networks). Exhaustive algorithms have been implemented; the JavaBayes system, freely distributed by the first author at http://www.cs.cmu.edu/~javabayes, offers some support for strong independence. Simulated annealing and genetic search have also been tested [3, 4]. Although the optimization problem is a reverse geometric program [1], geometric duality cannot be easily used here, because the number of dual variables is potentially huge. The most promising approach seems to be branch-and-bound al-

gorithms, coupled with redundancy-elimination computations. Because graphical structures can be used to generate bounds on probabilities, it is possible to gradually "cut" the sizes of credal sets when looking for a global maximum. At the same time, convex hull algorithms can be used to eliminate redundant vertices from credal sets.

References

[1] M. Avriel, *Advances in Geometric Programming*, Plenum Press, New York, 1980.

[2] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

[3] A. Cano, J. E. Cano, and S. Moral. "Convex sets of probabilities propagation by simulated annealing", *Proc. of the Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1994, pp. 4–8.

[4] A. Cano and S. Moral, "A genetic algorithm to approximate convex sets of probabilities", *Proc. of the Int. Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1996, Vol. 2, pp. 859–864.

[5] I. Couso, S. Moral, and P. Walley, "Examples of independence for imprecise probabilities", *Proc. of the First Int. Symp. on Imprecise Probabilities and Their Applications*, Ghent, Belgium, 1999, pp. 121–130.

[6] F. G. Cozman, "Calculation of posterior bounds given convex sets of prior probability measures and likelihood functions", *Journal of Computational and Graphical Statistics*, 1999, Vol. 8, No. 4, pp. 824–838.

[7] F. G. Cozman, "Credal networks", *Artificial Intelligence*, 2000, Vol. 120, pp. 199–233.

[8] F. J. Giron and S. Rios, "Quasi-Bayesian behaviour: A more realistic approach to decision making?", In: *Bayesian Statistics*, University Press, 1980, pp. 17–38.

[9] P. Hansen, B. Jaumard, M. P. de Aragão, F. Chauny, and S. Perron, "Probabilistic satisfiability with imprecise probabilities", *International Journal of Approximate Reasoning*, 2000, Vol. 24, No. 2-3, pp. 171–189.

[10] P. J. Huber, *Robust Statistics*, Wiley, New York, 1980.

[11] B. Jaumard, P. Hansen, and M. P. de Aragão, "Column generation methods for probabilistic logic", *ORSA Journal on Computing*, Spring 1991, Vol. 3, No. 2, pp. 135–148.

[12] V. P. Kuznetsov, *Interval Statistical Methods.* Radio i Svyaz Publ., 1991 (in Russian).

[13] I. Levi, *The Enterprise of Knowledge*, MIT Press, Cambridge, Mass., 1980.

[14] M. P. Pacifico, G. Salinetti, and L. Tardella, *Fractional optimization in Bayesian robustness*, Technical Report Serie A n. 23, Dipartamento di Statistica, Probabilita e Statistiche Applicate, Universita di Roma La Sapienza, Italy, 1994.

[15] S. I. Schaible and W. T. Ziemba, *Generalized Concavity in Optimization and Economics*, Academic Press, 1981.

[16] T. Seidenfeld, M. J. Schervish, and J. B. Kadane, "A representation of partially ordered preferences", *Annals of Statistics*, 1995, Vol. 23, No. 6, pp. 2168–2217.

[17] P. Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, London, 1991.

The Application Fields of the RejectIndex Parameter in Interval Methods for Global Optimization

Tibor Csendes[1], Leocadio Casado[2], and Inmaculada García[2]

[1]Institute of Informatics, University of Szeged
H-6701 Szeged, Pf. 652, Hungary
[2]Dept. Arquitectura de Computadores y Electrónica
Universidad de Almería, 04120 Almería. Spain

Suppose that for a given function $f : X \subseteq I\!R^n \to I\!R$, the global minimum f^*, or an approximation of it ($\hat{f}$) is a priori known. It may be obtained for example using a local search algorithm or during the execution of the B&B algorithm for global minimization. Let us define a parameter $pf^*(Y)$ called RejectIndex as:

$$pf^*(Y) = \frac{f^* - \underline{F}(Y)}{\overline{F}(Y) - \underline{F}(Y)} \in [0,1],\ Y \subseteq X.$$

Capitals denote intervals, underline and overline the lower and upper bounds, respectively. The inequality $f^* \geq \underline{F}(Y)$ holds, since pf^* is evaluated for intervals Y for which $F(Y)$ contains the global minimum value.

This parameter was designed mainly based on the following: traditionally an interval Y with the minimal value of $\underline{F}(Y)$ was considered as the best candidate to contain a global minimum. However, usually the larger the interval Y, the larger the over-estimation of the range $f(Y)$ obtained in $F(Y)$. Therefore a box could be considered as a good candidate to contain a global minimum just because it is larger than others. In order to compare subintervals with different size fairly we normalize the distance between f^* and $\underline{F}(Y)$.

The idea behind pf^* is simply that we expect the over-estimation to be nearly symmetric, i.e. the over-estimation $\overline{F}(Y) - \overline{f}(Y)$ above $f(Y)$ is closely equal to the over-estimation $\underline{f}(Y) - \underline{F}(Y)$ below $f(Y)$, for small sub-intervals containing a global minimizer point (that is at the same time a stationary point). Hence, for such intervals Y the relative place of the global optimum value inside the $F(Y)$ interval should be high, while for intervals far from global minimizer points pf^* must be small. Obviously, there are exceptions, and there exist no theoretical proof that pf^* would be a reliable indicator of nearby global minimizer points.

Based on these ideas, several application fields of pf^* and its variants were investigated. The talk plans to summarize them in a systematic way. The

RejectIndex has been previously used in parallel interval B&B algorithms as a predictor of the computational work associated to boxes in the work tree. In [1] it has been shown that using RejectIndex an almost perfect work load balance for parallel implementations of the Interval B&B algorithms can be obtained. It can also be applied to improve the multisection decision rule to achieve better overall efficiency (see [2]), since in the neighbourhood of minimizer points the intervals must be subdivided into more subintervals to decrease the number of function evaluations.

For hard to solve global minimization problems it may be an option to drop the guaranteed reliability (at least short term), and to get rid of those generated subintervals that can hardly contain global minimizer points. This can be done again on the basis of the pf^* values. According to our experiences, this technique could be an effective measure, and problems unsolved by traditional interval methods could be solved by interval minimization methods using the related heuristic rejection rules [3, 4]. To keep the reliability of interval methods, the rejected subintervals can be written into an output file for a possible later processing.

The fourth way for utilizing the RejectIndex is to use it in the decision which one from the list of candidate subintervals is to be chosen for the next subdivision. The interval selection rule is a very sensitive part of the B&B method, since over two decades no new paradigm was suggested in this field. Casado and coworkers suggested a new (not necessarily reliable) interval selection method based on pf^*, and reported improvements in efficiency in [5]. It was shown in [6] that with known global minimum value or with a good approximation of it a different new interval selection rule ensures the convergence of the minimization procedure to global minimizer points, and that it improves again the efficiency of the algorithm substantially.

Two further papers address additional issues on RejectIndex: the article of Kreinovich and Csendes shows ([7]) that the interval selection rule based on pf^* is optimal in certain sense; and Markót and coworkers suggest new algorithm parameters similar to RejectIndex for constrained minimization problems [8].

Acknowledgments. The research was supported by the Ministry of Education of Spain (CICYT TIC99-0361), and by the Grants FKFP 0449/99, OMFB ES 24/2001, OTKA T 017241, T 32118 and T 034350.

References

[1] L. G. Casado and I. García, "New load balancing criterion for parallel interval global optimization algorithms", *Proceedings of the 16th IASTED International Conference on Applied Informatics*, Garmisch-Partenkirchen, Germany, 1998, pp. 321–323.

[2] L. G. Casado, I. García, and T. Csendes, "A new multisection technique in interval methods for global optimization", *Computing*, 2000, Vol. 65, pp. 263–269.

[3] L. G. Casado, I. García, and T. Csendes, "A heuristic rejection criterion in interval global optimization algorithms", *BIT*, 2001, Vol. 41, pp. 683–692.

[4] L. G. Casado, I. García, and T. Csendes, "Heuristic Rejection in Interval Global Optimization", Submitted for publication. Available at `http://www.inf.u-szeged.hu/~csendes/publ.html`.

[5] L. G. Casado, J. A. Martínez, and I. García, "Experiments with a new selection criterion in a fast interval optimization algorithm", *J. Global Optimization*, 2001, Vol. 19, pp. 247-264.

[6] T. Csendes, "New subinterval selection criteria for interval global optimization", *J. Global Optimization*, 2001, Vol. 19, pp. 307-327.

[7] V. Kreinovich and T. Csendes, "Theoretical Justification of a Heuristic Subbox Selection Criterion for Interval Global Optimization", *Central European Journal of Operations Research CEJOR*, 2001, Vol. 9, No. 3, pp. 255–265.

[8] M. C. Markót, J. Fernández, L. G. Casado, and T. Csendes, "New interval methods for constrained global optimization", Manuscript, available at `http://www.inf.u-szeged.hu/~markot`.

Progress in Rational Approximation Theory Using Interval Arithmetic

Annie Cuyt

In the past 25 years a lot of effort has been put in the investigation of multivariate rational approximation techniques. See for instance the extensive bibliography in [2]. In this talk we will discuss the added value of interval arithmetic [4] when it comes to:

- the formulation of conjectures about the convergence behaviour of certain multivariate Padé approximants, such as)a Nuttall-Pommerenke analogue in higher dimensions (see [3]),
- the development of a reliable and fast method for the computation of multivariate rational approximants that give rise to a structured linear system of defining equations (see [1]).

References

[1] S. Becuwe and A. Cuyt, "Multivariate Padé approximants: homogeneous or not, that's the question", *Proceedings of IMACS 2000*, 2000.

[2] A. Cuyt, "How well can the concept of Padé approximant be generalized to the multivariate case", *J. Comput. Appl. Math.*, 1999, Vol. 105, pp. 25–50.

[3] A. Cuyt, K. Driver, and D. Lubinsky, "A direct approach to convergence of multivariate nonhomogeneous Padé approximants", *J. Comput. Appl. Math.*, 1996, Vol. 69, pp. 353–366.

[4] A. Cuyt and B. Verdonk, "The need for knowledge and reliability in numerical computation: case study of multivariate Padé approximation", *Acta Appl. Math.*, 1993, Vol. 33, pp. 273–302.

Interval Arithmetic Applied to Structural Design of Uncertain Mechanical Systems

Olivier Dessombz, Fabrice Thouverez,
Jean-Pierre Laîné, and Louis Jézéquel

Laboratoire de Tribologie et Dynamique des Systèmes
École Centrale de Lyon
BP 163, 69131 Ecully Cedex, France
E-Mail: dessombz@mecasola.ec-lyon.fr

If we are interested in the static and dynamic behavior of an industrial mechanical structure, one has to consider Finite Element Modeling, which leads to matrices (such as stiffness, mass, or damping matrix). Thus, linear systems of equations are to be solved. If some of the mechanical parameters are uncertain at design stage, or are variable such as the weight of a tank, they can be modeled using the interval theory. The uncertain parameters can be geometrical ones (length, thickness, ...), or physical ones (Young's Modulus, ...). Then the matrices given by the Finite Element theory are interval matrices, and the problem is written as:

$$[A]\{x\} = \{b\} \tag{1}$$

with $[A] \in [\boldsymbol{A}]$ and $\{b\} \in \{\boldsymbol{b}\}$. Although several problems can be distinguished, as done by Chen and Ward in [1] and by Shary in [8], we will focus exclusively in this paper on the outer problem which is defined as $\Sigma_{\exists\exists}([\boldsymbol{A}], \{\boldsymbol{b}\})$,where $[\boldsymbol{A}]$ is an interval matrix and $\{\boldsymbol{b}\}$ an interval vector:

$$\Sigma_{\exists\exists}([\boldsymbol{A}], \{\boldsymbol{b}\}) = \{x \in \mathbb{R}^n | (\exists [A] \in [\boldsymbol{A}]), (\exists \{b\} \in \{\boldsymbol{b}\}) / [A]\{x\} = \{b\}\} \tag{2}$$

In general this set is not an interval vector. It is a non convex polyhedra. The Oettli and Prager theorem [5] give the exact solution set.

Nevertheless, this method is quite difficult to use with matrices corresponding to real physical cases in a n-dimensional problem. Most of the time, we will consider the smallest interval vector containing $\Sigma_{\exists\exists}([\boldsymbol{A}], \{\boldsymbol{b}\})$, which is defined as $\square\Sigma_{\exists\exists}([\boldsymbol{A}], \{\boldsymbol{b}\})$. In this case, this ensures that the true solution is included in the numerical solution found $\square\Sigma_{\exists\exists}([\boldsymbol{A}], \{\boldsymbol{b}\})$.

The existing algorithms used to solve $\Sigma_{\exists\exists}([\boldsymbol{A}], \{\boldsymbol{b}\})$ have been formulated for reliable computing on a numerical point of view. In an interval matrix for

instance, each term can vary independently of each other in its interval, which is generally sharp.

If the interval formulation has to be adapted to mechanics, the dependence between the parameters must be taken into account. Many of the terms of the matrices are depending on the same parameters. For example if the Young's modulus varies in $\boldsymbol{E}$, the stiffness matrix can formally be written $\boldsymbol{\alpha}\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$, which is not the same as $\begin{bmatrix} \boldsymbol{\alpha} k_{11} & \boldsymbol{\alpha} k_{12} \\ \boldsymbol{\alpha} k_{21} & \boldsymbol{\alpha} k_{22} \end{bmatrix}$, that is treated as $\begin{bmatrix} \alpha_1 k_{11} & \alpha_2 k_{12} \\ \alpha_3 k_{21} & \alpha_4 k_{22} \end{bmatrix}$, with $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ varying in $\boldsymbol{\alpha}$ independently.

Moreover, the stiffness matrices are symmetric positive and definite. If all the matrices $[K] \in [\boldsymbol{K}]$ are considered, we must notice that many of them do not physically correspond to stiffness matrices. We shall then consider several problems. For a system with few independent degrees of freedom, general algorithms can be used. Nevertheless, any information on the special form of the matrices (symmetric positive and definite for a stiffness matrix, for instance) is lost, and the solution set can be widely overestimated. The problem is then formulated as:

$$[A] = [A_0] + \sum_{n=1}^{N} \boldsymbol{\epsilon_n} [A_n] \qquad \{b\} = \{b_0\} + \sum_{p=1}^{P} \boldsymbol{\beta_p} \{b_p\} \tag{3}$$

N and P are the number of parameters to be taken into account when building the matrix $[A]$ and the vector $\{b\}$. $\boldsymbol{\epsilon_n}$ and $\boldsymbol{\beta_p}$ are independent centered intervals, generally $[-1, 1]$. $[A_0]$ and $\{b_0\}$ correspond to the matrices and vector built from the mean values of the parameters.

Nevertheless, a special algorithm is required, because the solution set is not given by a combination of the bounds of the parameters. The following section is devoted to the presentation of a novel algorithm which enables to obtain a robust and including solution.

Our aim is obtaining an including solution by means of a new algorithm.

The algorithm that has been choosen is the Rump's inclusion ([4]), which relies on the fixed point theorem. In the general case, this algorithm converges rapidly, with a good accuracy (see [6]), and the convergence conditions have been studied by Rohn and Rex in [7].

In order to apply this algorithm to the mechanical formulation some adaptations are required, due to the specific differences of the mechanical problems highlighted previously.

Let us consider a system in which only one parameter is an interval. The general equation of this system is:

$$([A_0] + \alpha[A_1])\{x\} = \{b\} \qquad \alpha \in \boldsymbol{\alpha} \tag{4}$$

And the particular form of the matrix has to be taken into account (see [2]).

$$[\boldsymbol{A}] = [A_0] + \boldsymbol{\alpha}[A_1] \tag{5}$$

A method to ensure the convergence of this algorithm is also proposed (the algorithm is also based on the fixed point theorem, and the iteration matrix must be contracting). The strategy proposed is to split the interval into a partition of it, and then work on narrower intervals, on which the convergence condition will be verified.

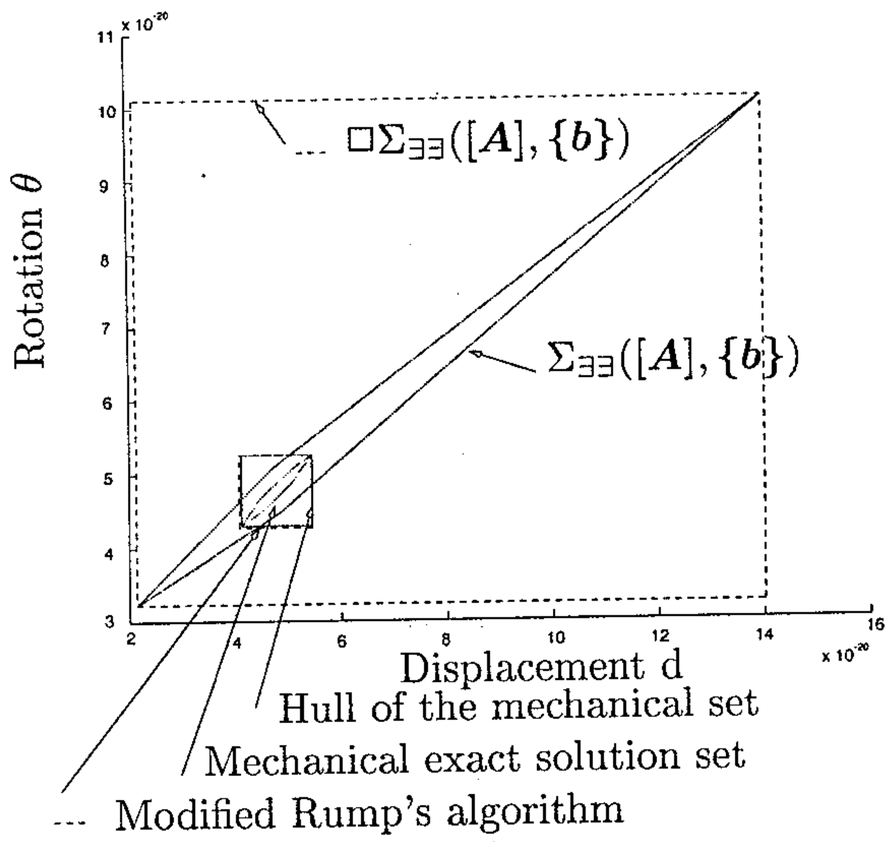


Figure 1: Solution sets for the clamped free beam. EI is uncertain ($\pm 2\%$). Numerical global problem, and reduced mechanical problem, and their respective hulls.

We can first compare the results of the general algorithm and the adapted one, to show the importance of the factorization.

The results found with the modified Rump's algorithm are often much sharper than the ones found with the classical formulation, which will be shown on a simple structure: a clamped free beam.

To test our algorithm, we have computed the result of the modified Rump's algorithm. It is illustrated on Figure 1. As we can see, it is overestimating the exact solution, but it gives a good idea of the size of the solution.

As it had been noticed in [1], a large overestimation is obtained when including the parameters in the elements of the matrices. For finite element matrices, this overestimation can become critical, and often leads to an insolvable problem. As we have shown above, even on 2×2 matrices, the overestimation can reach 10 times or more. Such an adaptation of this algorithm enables its use for industrial problems involving huge size matrices.

If static problems can be solved, which means finding solutions for linear interval systems, the algorithm can also be applied to the calculation of transfer functions.

We consider some kind of realistic structure. It is a two dimensional frame

structure (Figure 2). It can model for instance the structure of a building, as in [3].

It is a 18 elements structure, with 12 degrees of freedom. Only a concentrated load on the beams is considered, applied on the DOFF 3, the torque $F = 10^3\ Nm$. The parameters of the model are the lengths of the beams $L1 = L2 = 1\ m$, the inertia $I = \pi\ 10^{-8}/4\ m^4$ and their area $S = \pi\ 10^{-4}\ m^2$. We assume that the bending rigidity E is uncertain ($E = 210 \pm 10\%\ GPa$).

We will study the two-dimensional frame structure from a dynamical point of view. In the Finite Element Model, we use a Euler Bernoulli Beam model, leading to a stiffness and a mass matrix. We suppose that there is hysteretic damping ($\eta = 2\%$) in the model. Because all of the beams are identical, some of the modes are relatively close to each other. This means that when the bending rigidity is varying, the eigenmodes can overlap each others. This behavior is illustrated in the Figure 3, where a harmonic torque is applied on node 3 of the truss. The collocated transfer function $H(3,3)$ is computed thanks to the proposed algorithm, and compared to deterministic transfer functions calculated for various values of the Young's modulus.

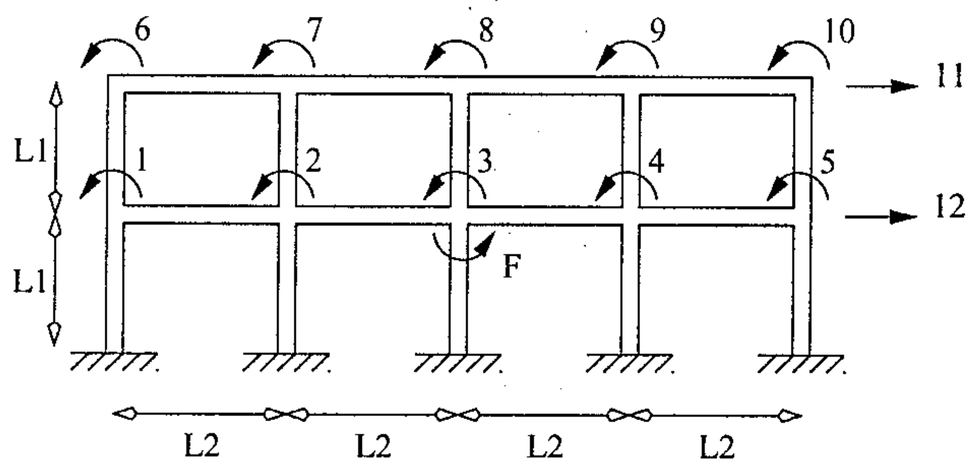


Figure 2: Two dimensional frame structure with 12 DOFF. A torque F is applied on node 3.

As we can see in Figure 3, the proposed algorithm can take overlapping eigenfrequencies into account. It leads to an envelope of the modulus of the transfer function. Some of the deterministic transfer function have been plotted, to show that the envelope found does not overestimate the real solution too much.

We will also consider a three blades wheel that is modeled with a 7 DOFF system (see Figure 4). As the 3 blades are identical in the deterministic model, the eigenfrequencies are found as multiple eigenvalues of a matrix system. If one of the blades is mistuned, then the eigenvalues are no more multiple ones, and new resonances can appear.

On Figure 5 the transfer function $H(1,3)$ is shown. Blue line represent the deterministic case for which all of the three blades are identical, and magenta lines the envelope of that transfer function, when the blade one has an uncertain Young's modulus ($E = E_0 \pm 10\%$). The envelope shows four resonance zones This is due to the mistuning phenomenon.

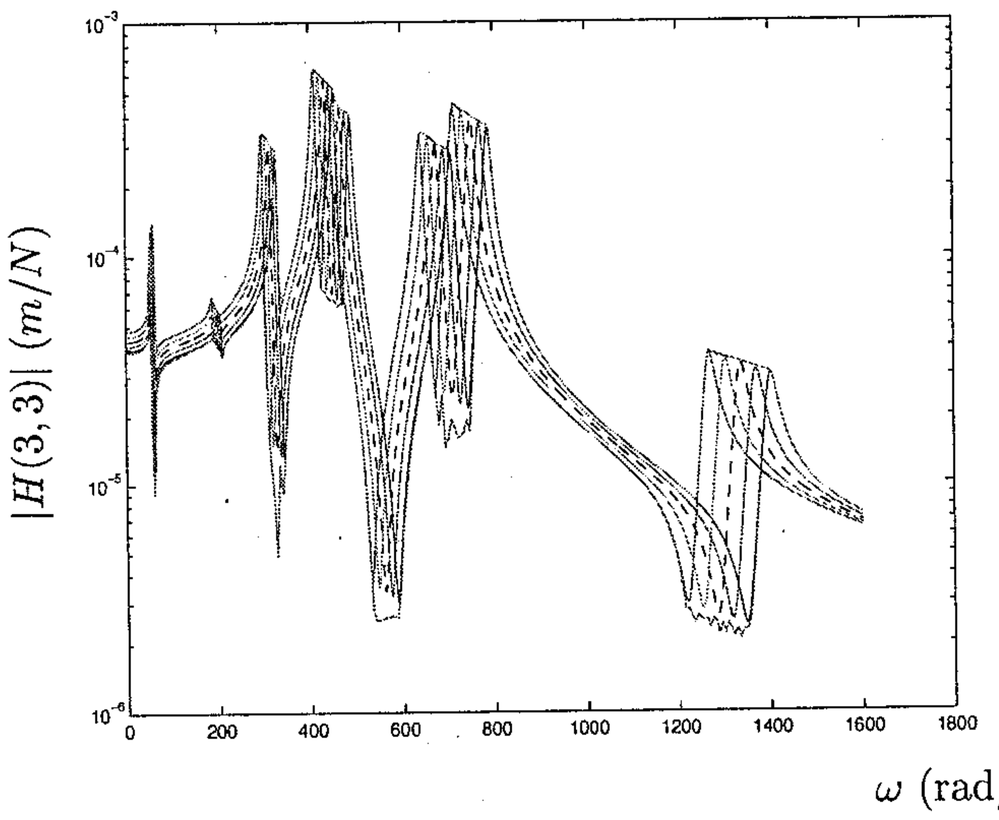


Figure 3: Modulus of the collocated transfer function $H(3,3)$ for the frame. The bending rigidity is uncertain ($E = 210 \pm 10\%$ GPa). The min and max values calculated with the modified algorithm are represented, wrapping the transfer function for several values of E.

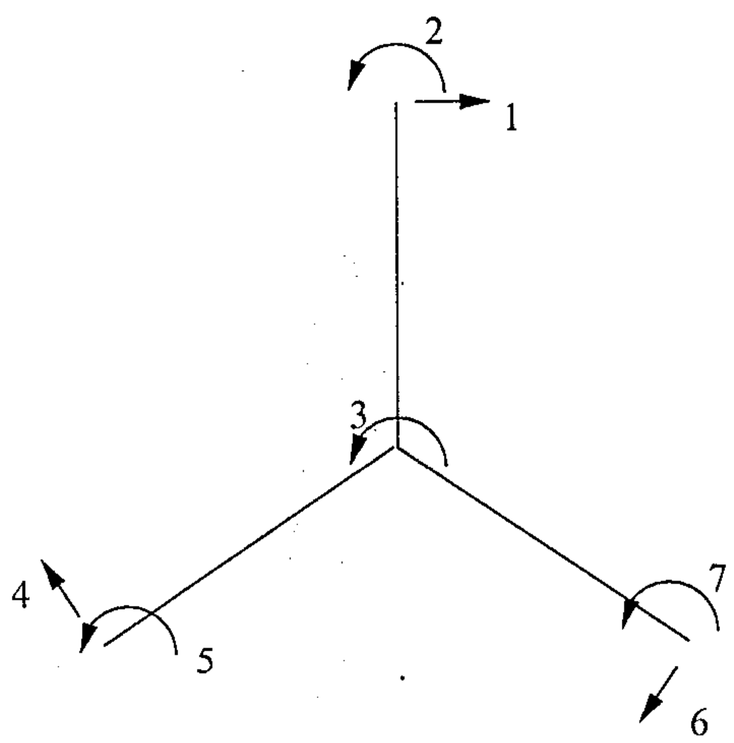


Figure 4: 3 blades wheel, and the 7 DOFF.

The modified algorithm gives accurate results, once again with the advantage of getting a robust envelope. This method can improve considerably the accuracy of prediction of the dynamic behavior of mechanical systems.

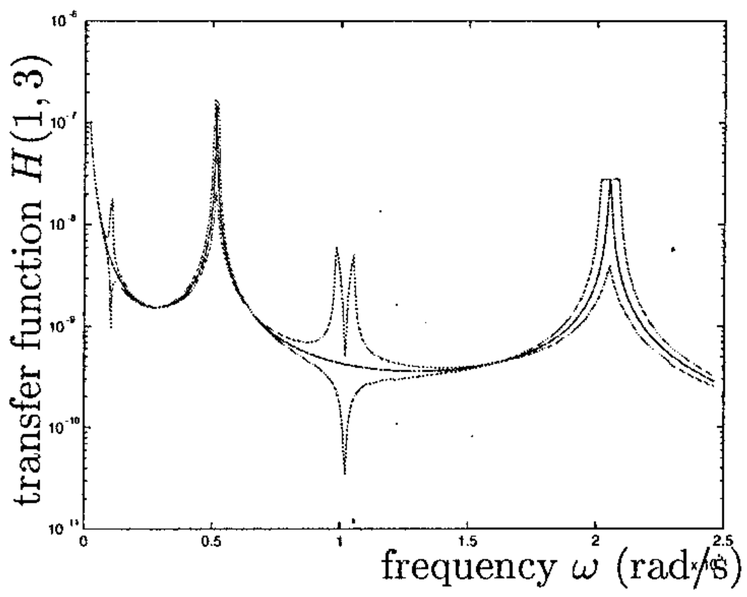


Figure 5: Modulus of the transfer function $H(1,3)$. The Young's modulus of the first blade is uncertain ($E = E_0 \pm 10\%$).

References

[1] R. Chen and A. C. Ward, "Generalizing interval matrix operations for design", *Journal of Mechanical Design*, 1997, Vol. 119, pp. 65–72.

[2] O. Dessombz, F. Thouverez, J.P. Laîné, and L. Jézéquel, "Analysis of mechanical systems using interval computations applied to finite elements methods", *Journal of Sound and Vibration*, 2001, Vol. 239, No. 5, pp. 949–968.

[3] H. U. Koyluoglu, A. S. Cakmak, and S. R. K. Nielsen, "Interval algebra to deal with pattern loading and structural uncertainties", *Journal of Engineering Mechanics*, Nov. 1995, pp. 1149–1157.

[4] S. M. Rump, "Solving algebraic problems with high accuracy", In: U. Kulisch and W. Miranker, *A New Approach to Scientific computation*, Academic Press, New York, 1983, pp. 51–120.

[5] W. Oettli and W. Prager, "Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides", *Math.*, 1964, Vol. 6, pp. 405–409.

[6] J. Rohn, *Validated solutions of linear equations*, Technical Report 620, Institute of Computer Science, Academy of Sciences of the Czech Republic, Jan. 1995.

[7] J. Rohn and G. Rex, *Enclosing solutions of linear equations*, Technical Report 666, Institute of Computer Science, Academy of Sciences of the Czech Republic, Feb. 1996.

[8] S. P. Shary, "On optimal solution of interval linear equations", *SIAM J. Numer.Anal.*, 1995, Vol. 32, No. 2, pp. 610–630.

Modelling Geometrical Tolerances with Intervals Using ISO-Standard STEP

Eva Dyllong, Wolfram Luther, and Holger Traczinski
Institute of Computer Science and Interactive Systems
Faculty of Engineering
Gerhard-Mercator-University of Duisburg
D-47048 Duisburg, Lotharstr. 65, Germany
{luther,dyllong,traczinski}@informatik.uni-duisburg.de

A development of complex mechatronic systems requires a multitude of decentralised designing and processing systems. The various software and file formats which are involved in the development process need a quick and smooth data exchange via standardised data models and interfaces. The ISO 10303 norm STEP (STandard for the Exchange of Product model data) has been developed by the International Organisation for Standardisation (ISO) within the Technical Committee 184 "Industrial Automation System & Integration". STEP provides a basic approach for an unambiguous representation of computer interpretable product information throughout the whole life-cycle of a product [1]. There are three key components of STEP: the EXPRESS data specification language is used to define data structures, entities, attributes, relationships and constraints together with EXPRESS-G, which is intended as a graphic representation of data models [10]. The Implementation Forms provide a physical file format for data that conforms to any EXPRESS schema and a Standard Data Access Interface (SDAI) to databases and applications. The Standard Data Models include integrated resources for almost universal applicability and specific models fulfilling the requirements of a particular industrial application. Around the kernel, the Application Protocols of models are defined, based on the integrated resources and implementation methods.

Thus, the ISO-Standard STEP can be interpreted as a construction kit with application protocols, which use predefined elementary construction kits under specific rules and norm methods. An application protocol consists of three parts. The first part is the application activity model (AAM), which describes the functionality and life-cycle of a product using the SADT (Structure Analysis and Design Technique) method. The second part contains the application reference model (ARM). In this part the formal descriptions of the objects take place which are identified in AAM using the description language EXPRESS or EXPRESS-G. The Application Interpreted Model (AIM) stands at the end of the application protocol and concerns the mapping of the ARM-model into an

integrated resource model.

In 1997 a new project, MechaSTEP, was initiated within the scope of ISO 10303 focusing on a neutral data format for the data exchange between mechatronic systems.

This abstract presents results in the field of the requirements analysis on ISO-Standard 10303 STEP for interval arithmetic purposes. We are mainly concerned with Part 42 of the integrated resources, which is entitled "Geometrical and Topological Representation" and with Part 47, which deals with shape variation tolerances.

Part 42 includes construction specifications for the geometrical and topological representations of an object. Furthermore, it contains requisites for an explicit representation of an object model. Geometric description alternatives are the volume based, the surface based and the wireframe based designs. Special attention is paid to Faceted Boundary REPresentation (FBREP) models belonging to the first alternative together with Manifold solid Boundary REPresentation (MBREP) and Constructive Solid Geometry (CSG) models.

Recently, there is a growing interest in researching methods for validation and verification in applications. For example, path-planning, localization and tracking of a mobile robot using techniques of scientific computation are described in [6]. In a recent project funded by the German Research Council (DFG) we implemented efficient and accurate algorithms for distance calculation between a flexible robot and a target or obstacles in the complex environment and for the resulting contact problem [7]. Robust solutions to these problems are also used in the collision-free path planning if a given end-effector is moving amid a collection of known obstacles from an initial to a desired final position. For simulation purposes, the obstacles are taken to be a collection of polyhedral surfaces. Accurate floating-point algorithms have been implemented based on suitable projections and using controlled rounding and the precise dot product whereby verified error-bounds are ensured [2, 3]. If the end-effector or the sensor is taken to be a single moving point, an efficient distance algorithm applicable to non-convex polyhedral surfaces is to be used [4]. In [9] is described, how interval methods can help a moving robot with uncertain sonsors to avoid collisions with obstacles.

To allow a data transfer to mechatronic modelling and simulation software, the representation of polyhedra using the FBREP models of the STEP-standard has been highlighted. Geometric elements like points, curves, and surfaces have been described and topological relationships between them identified: the vertex points of a polyhedron in a Cartesian coordinate system as geometrical and the position of these points on its faces as topological input data. This information is given in a simple ASCII-file.

We implemented an interface which transforms this file used in our algorithms into a STEP-based one and vice-versa using Part 21 of the implementation methods in STEP for clear text encoding of the exchange structure. The output file consists of two parts: the header and the data section. The header section includes entity declarations and the data section the descriptions of the given input data using the EXPRESS language. The latter is divided

into three subparts: the first for the description of the boundary of each face of the polyhedron with specification of the Cartesian coordinates, the second for the description of the faceted surface and the third for the description of the polyhedron as a closed shell. In general, STEP-files are very large: a more complex geometrical object will require several hundred lines with many pieces of repeated information. Thus, automatic generation is necessary.

In the next version, the interface should also handle cylindrical, conical, spherical, B-Spline or offset surfaces as a part of the integrated resources in Part 42.

STEP Part 47 is about shape variation tolerances. Information about tolerances is important for product definition. Many interface definitions consider tolerances only in design information like the Initial Graphics Exchange Specification (IGES), an ANSI-standard for exchanging geometric design information between CAD systems or the corresponding French national Standard d'échange et de transfert SET. The disadvantage here is that these dimensioning and tolerancing specifications are not supported in current Computer-Aided Engineering and Manufacturing systems. STEP wants to remedy this situation and integrates various single-feature and related-feature tolerances. A tolerance in STEP can either be a tolerance for sizes or for geometry. To describe tolerances for dimensions, uncertainties in lengths or angles are stored in intervals with lower and upper bounds. A range of acceptable values, also known as 'limits and fits', may be selected from a standard catalogue of acceptable ranges. Additionally, a tolerance can be defined by only one bound and by significant digits. Finally, a nominal value may have no tolerance bounds associated with it.

Geometric tolerances address the acceptable deviation of the form of a manufactured object and are usually expressed as an area or volume in which the realised form must lie [11]. Tolerances in geometry mainly concern specifications about orientation (perpendicularity, parallelism, angularity), location (concentricity, symmetry) or shape (flatness, straightness, cylindricity) [12]. In this paper, a general STEP-based dimensioning and tolerancing data model is developed and implemented.

Tolerances of measures and geometry are suitable for interval calculus in two directions. In modern modelling systems like MOBILE extensions for interval arithmetic have been added [8]. In [5] a new kinetostatic transmission element was modelled with interval arithmetic. Thus, not only uncertainties in the length of the arms of multibody systems or uncertain weights at the end-effector of a manipulator can be modelled in simulation systems but also new types of (geometrical) elements. For a standardised description of such a modelled system both types of tolerances in measure and geometry are needed. Thus far the STEP concept is basically right. Although STEP offers the possibility of storing uncertain information in intervals this information is linked with the specific objects of a model. Here, a more general usage of intervals would be desirable. For example, a complex modelled object which consists of vertices, edges, and faces is stored as an instance with exact values in STEP. Additionally, tolerances can be introduced at one point of the description, such as at a line segment. Although these tolerances affect further elements of the object, for

instance a vertex point on the line, these dependencies are not stored in a STEP-file. However, they can be worked out from other information. Thus, the ability to store information about uncertainty in all parts of a STEP-file would be helpful. For this purpose a basic data type for intervals is needed.

To include self- and cross-reference tolerances, Tsai et al. [12] introduced the notion of a tolerance network which represents shape features as nodes and geometric tolerance specifications as arcs connecting nodes. They construct these networks for single pieces and, by means of recursive algorithms, for assembled products as well. The corresponding STEP product data file implements the tolerance network with appropriate back and cross references.

The other way to use interval calculus in building tolerance dependencies is by checking the consistency of given tolerances with interval arithmetic. Interval arithmetic provides a tool with which to calculate the guaranteed enclosures of computational results and enables us to determine the enclosure of a value at one part of a STEP description resulting from tolerance information in other parts of the file. If the STEP tolerance value for this part does not fit in the enclosure, the whole STEP-file is not consistent. Otherwise, the result of such a computation should be stored within the STEP-file. If there are different ways to calculate an enclosure for the same value, the intersection of the intervals should be chosen to get tighter enclosures. At this point, another reason arises for a basic interval data type.

To illustrate the benefit of interval arithmetic in tolerancing data models, we checked the complete "geometric dimensions and dimensional tolerance example" described in [11] and proved its consistency by transferring the STEP-data into the extended MOBILE modelling system. This example covers the representation of all the dimensional tolerances supported, including plus-or-minus deviations, maxima, minima and nominal dimensions, limits and fits, and significant digits.

References

[1] R. Anderl, and D. Trippner, *STEP: Standard for the Exchange of Product Model Data*, Teubner, Stuttgart, 2000.

[2] E. Dyllong, W. Luther, and W. Otten, "An accurate distance-calculation algorithm for convex polyhedra", *Reliable Computing*, 1999, Vol. 3, No. 5, pp. 241–254.

[3] E. Dyllong and W. Luther, "An accurate computation of the distance between a point and a polyhedron", Vol. 80 of *GAMM 99 Annual Meeting*, Metz, France, April 12–16, Wiley-VCH, Berlin, 2000, pp. S771–S772.

[4] E. Dyllong and W. Luther, "Distance-Calculation Between a Point and a NURBS Surface", In: P.-J. Laurent, P. Sablonnire, L. L. Schumaker (eds.), *Curve and Surface Design – Saint Malo, 1999*, Vanderbilt University Press, Nashville TN, 2000, pp. 55–62.

[5] C. Hörsken and H. Traczinski, "Modeling of multibody systems with interval arithmetic", In: W. Krämer and J. Wolff von Gudenberg (eds.), *Scientific Computing, Validated Numerics, Interval Methods, Proc. of SCAN 2000*, 2001, pp. 317–328.

[6] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer, Berlin, 2001.

[7] W. Luther, E. Dyllong, D. Fausten, W. Otten, and H. Traczinski, "Numerical verification and validation of kinematics and dynamical models for flexible robots in complex environments", In: U. Kulisch, R. Lohner, A. Facius (eds.), *Perspectives on Enclosure Methods*, Springer, Wien, 2001, pp. 181–199.

[8] W. Luther and H. Traczinski, "Error Propagation Control in MOBILE: Extended Basic Mathematical Objects and Kinetostatic Transmission Elements", In: A. Kecskeméthy, M. Schneider, and C. Woernle (eds.), *Advances in Multibody Systems and Mechatronics*, 1999, pp. 267–276.

[9] D. Morales, and T. C. Son, "Interval Methods in Robot Navigation", *Reliable Computing*, 1998, Vol. 4, No. 1, pp. 55–61.

[10] D. Schenck and P. Wilson, *Information Modeling: The EXPRESS Way*, Oxford University Press, New York, 1994.

[11] M. Strub, M. Hauser, and T. Hendrix, *Recommended Practices for Dimensions and Dimensional Tolerances*, April 18, 2000. `http://www.cax-if.org/public/Rec_Prac_DDT_v2.pdf`

[12] J.-C. Tsai, T.-C. Chuang, T.-C., and D.-N. Guo, "Development of a Step-based Dimensioning and Tolerancing Data Model", *Proc. Natl. Sci. Counc. ROC(A)*, 1998, Vol. 22, No. 6, pp. 831–840. `http://nr.stic.gov.tw/ejournal/ProceedingA/v22n6/831-840.pdf`

Interval Methods in Digital Signal Processing

William W. Edmonson, Ph.D.
Department of Electrical Engineering
Hampton University
Hampton, Virginia

Interval methods represent a relatively new research direction in digital signal processing. Though, in the closely related field of controls there has been much work that can also be applied to signal processing. In general interval methods provide a way for providing verification of computations or as an optimization procedure. Interval arithmetic provides a method for determining how numerical errors scale as a result of implementing algorithms on computational machines of various word lengths and number representation. The tracking of numerical errors can be exploited in signal processing through the use interval arithmetic since the computations are done on various types of computers. Computing systems used range from dedicated processors using fixed point arithmetic with short word lengths to supercomputers using floating point arithmetic with very large word lengths.

The estimation of system parameters from noisy data represents another important topic in signal processing. If the system has feedback or is nonlinear then the associated objective function to be optimized can be nonconvex. This would require global optimization methods to insure that convergence to the absolute optimal is achieved. Interval methods, a deterministic optimization method, advantage over other global optimization methods is its ability to find the global optimum of nonconvex differentiable or non-differentiable objective functions. It represents the method to attempt first if the one has no knowledge of where the global optimum might exist on the parameter space.

This talk will focus on the research that was done by the author in applying interval methods to digital signal processing. Work which covers both optimization and analysis. In optimization, we will discuss the use of interval methods for solving the sinusoidal parameter estimation problem. Research on adaptive systems using interval analysis to validate the results and to monitor stability and errors will also be discussed. The talk will end with a discussion on problems and prospects of using interval methods in signal processing.

The sinusoid parameter estimation problem consist of determining the maximum-likelihood estimates of sinusoid parameters from a signal that consists of sinusoids and additive noise. We will present three algorithms that integrate

interval methods for global optimization with procedures that decompose the problem into smaller ones. The interval method used is a global optimization technique that is based upon the branch and bound principle. More specifically, decomposition of the problem is accomplished via the expectation-maximization algorithm and the grouped coordinate descent algorithm. Although a formal proof of convergence is not addressed, the performance of the algorithms from simulations was shown to be superior to the popular iterative quadratic maximum likelihood (IQML) method.

An adaptive system is one that can adapt to a changing environment through optimization of the systems parameters. Thereby, its objective function can vary as a function of time. Adaptive filtering algorithms can have problems converging to the optimal parameters due to numerical errors and tracking a time-varying objective function. These errors are manifested through instability of the algorithm, arithmetic precision caused by finite word length of the processor, slow convergence and the ability to track a time varying minima. The use of interval arithmetic yields a better performing algorithm by:

- Monitoring certain parameters of the optimization algorithm to eliminate instabilities caused by mathematical operations of numbers of very different order of magnitude.
- Bounding the results.
- Bounding the parameter space to insure the algorithm converges to a stable filter.
- In conjunction with heuristics or evolutionary strategies for fast convergence to the global minimum.

Many of the adaptive filtering applications will be implemented on a finite word length machine that are optimized to implement multiply-sum operations fast and efficiently, i.e. digital signal processors. These machines implement arithmetic operations, basic to signal processing algorithms, of addition, subtraction, and multiplication. This implies that a reduced version of interval arithmetic operations can be implemented and is feasible for development on field programmable gate arrays (FPGA's). This could further lead to developing hardware for an adaptive system that adapts to both the parameters and the structure (word length, filter order, and the number of inputs and outputs). To change or adapt the systems structure, it must have the ability to monitor performance and for the hardware to evolve based on changing conditions of its operating environment.

The problems associated with applying interval methods to signal processing is the additional time required for arithmetic operations and the lack of awareness amongst the signal processing community. To overcome the problem of speed is to develop dedicated hardware that is optimal for interval arithmetic operations, e.g. FPGA's.

References

[1] W. Edmonson, W. Lee, and J. M. M. Anderson, "Interval method for sinusoidal parameters estimation: A comparative analysis," *Reliable Computing*, 2000, Vol. 6, No. 3.

[2] W. Edmonson, K. Srinivasan, C. Wang, and J. Principe, "A global least mean square algorithm for adaptive IIR filtering," *IEEE Transaction on Circuits and Systems, Part II*, 1998, Vol. 45, No. 3.

[3] A. P. Voshchinin, "Some questions of application of interval mathematics in parameter estimation and dicision making," *Interval Computations*, 1992, No. 4(6).

[4] C. P. Callender and C. F. N. Cowan, "Implementation of numerically robust fast recursive least squares adaptive filters using interval arithmetic," *Electronics Letters*, 1992, Vol. 28, No. 7.

[5] M. J. Schulte and E. E. Swartzlander, Jr., "Variable-precision, interval arithmetic coprocessors," *Reliable Computing*, 1996, Vol. 1, No. 2.

Exact Bounds on Sample Variance of Interval Data

Scott Ferson[1], Lev Ginzburg[1],
Vladik Kreinovich[2], and Monica Aviles[2]

[1]Applied Biomathematics, 100 North Country Road,
Setauket, NY 11733, USA, {scott,lev}@ramas.com
[2]Computer Science Department, University of Texas at El Paso
El Paso, TX 79968, USA, {maviles,vladik}@cs.utep.edu

Abstract

We provide a feasible (quadratic time) algorithm for computing the lower bound $\underline{V}$ on the sample variance of interval data. The problem of computing the upper bound $\overline{V}$ is, in general, NP-hard. We provide a feasible algorithm that computes $\overline{V}$ for many reasonable situations.

Formulation of the problem. When we have n results $x_1, \ldots, x_n$ of repeated measurement of the same quantity, traditional statistical approach usually starts with computing their sample average

$$E = \frac{x_1 + \ldots + x_n}{n}$$

and their sample variance

$$V = \frac{(x_1 - E)^2 + \ldots + (x_n - E)^2}{n - 1}$$

(or, equivalently, the sample standard deviation $\sigma = \sqrt{V}$); see, e.g., [1].

Sample variance is an unbiased estimator of the variance of the distribution from which observations are assumed to be randomly sampled. For Gaussian distribution, this estimator is a maximum likelihood estimator of the distribution variance.

In some practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i. This happens, for example, if instead of observing the actual value x_i of the random variable, we observe the value $\widetilde{x}_i$ measured by an instrument with a known upper bound Δ_i on the measurement error; then, the actual (unknown) value is within the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

As a result, the sets of possible values of E and V are also intervals. The interval $\mathbf{E}$ for the sample average can be obtained by using straightforward interval computations, i.e., by replacing each elementary operation with numbers by the corresponding operation of interval arithmetic:

$$\mathbf{E} = \frac{\mathbf{x}_1 + \ldots + \mathbf{x}_n}{n}.$$

What is the interval $[\underline{V}, \overline{V}]$ of possible values for sample variance V?

When the intervals $\mathbf{x}_i$ intersect, then it is possible that all the actual (unknown) values $x_i \in \mathbf{x}_i$ are the same and hence, that the sample variance is 0. In other words, if the intervals have a non-empty intersection, then $\underline{V} = 0$. Conversely, if the intersection of $\mathbf{x}_i$ is empty, then V cannot be 0, hence $\underline{V} > 0$. The question is (see, e.g., [2]): What is the total set of possible values of V when the above intersection is empty?

For this problem, straightforward interval computations sometimes overestimate: E.g., for $\mathbf{x}_1 = \mathbf{x}_2 = [0, 1]$, the actual $V = (x_1 - x_2)^2/2$ and hence, the actual range $\mathbf{V} = [0, 0.5]$. On the other hand, $\mathbf{E} = [0, 1]$, hence

$$(\mathbf{x}_1 - \mathbf{E})^2 + (\mathbf{x}_2 - \mathbf{E})^2 = [0, 2] \supset [0, 0.5].$$

Three intervals $\mathbf{x}_i$ equal to $[0, 1]$ show that a centered form also does not always lead to the exact range.

The problem reformulated in statistical terms. The traditional sample variance is an unbiased estimator for the following problem: observation points x_i satisfy the equation $x_i = u - \varepsilon_i$, where u is an unknown fixed constant and the ε_i are independently and identically distributed random variables with zero expectation and unknown variance σ^2.

In our paper, we want to handle a situation in which each observation point $\widetilde{x}_i$ satisfies the condition $\widetilde{x}_i - u - \varepsilon_i \in \Delta_i \cdot [-1, 1]$, where the values Δ_i are assumed to be known. From this model, we can conclude that each $u + \varepsilon_i$ is contained in the corresponding interval $\widetilde{x}_i + \Delta_i \cdot [-1, 1] = \mathbf{x}_i$. As a solution to this problem, we take the interval consisting of all the results of applying the estimator V to different values $x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n$.

Our first result: computing $\underline{V}$. First, we design a *feasible* algorithm for computing the exact lower bound $\underline{V}$ of the sample variance. Specifically, our algorithm is *quadratic-time*, i.e., it requires $O(n^2)$ computational steps for n interval data points $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$. We have implemented this algorithm in C++, it works really fast. The algorithm is as follows (the proof that this algorithm is correct will be provided in the full paper):

- First, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(n)}$. This sorting requires $O(n \cdot \log(n))$ steps.
- Second, we compute $\underline{E}$ and $\overline{E}$ and select all "small intervals" $[x_{(k)}, x_{(k+1)}]$ that intersect with $[\underline{E}, \overline{E}]$.

- For each of selected small intervals $[x_{(k)}, x_{(k+1)}]$, we compute the ratio $r_k = S_k/N_k$, where

$$S_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \geq x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \leq x_{(k)}} \overline{x}_j,$$

and N_k is the total number of such i's and j's. If $r_k \notin [x_{(k)}, x_{(k+1)}]$, we go to the next small interval, else we compute

$$V_k' \stackrel{\text{def}}{=} \frac{1}{n-1} \cdot \left(\sum_{i:\underline{x}_i > x_{(k+1)}} (\underline{x}_i - r)^2 + \sum_{j:\overline{x}_j < x_{(k)}} (\overline{x}_j - r)^2 \right).$$

(if $N_k = 0$, we take $V_k' \stackrel{\text{def}}{=} 0$).

- Finally, we return the smallest of the values V_k' as $\underline{V}$.

Second result: computing $\overline{V}$ is NP-hard. Our second result is that the general problem of computing $\overline{V}$ from given intervals $\mathbf{x}_i$ is NP-hard.

Third result: a feasible algorithm that computes $\overline{V}$ in many practical situations. NP-hard means, crudely speaking, that there are no general ways for solving all particular cases of this problem (i.e., computing $\overline{V}$) in reasonable time.

However, we show that there are algorithms for computing $\overline{V}$ for many reasonable situations. For example, we propose an efficient algorithm $\mathcal{A}$ that computes $\overline{V}$ for the case when the "narrowed" intervals $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ – where $\widetilde{x}_i = (\underline{x}_i + \overline{x}_i)/2$ is the interval's midpoint and $\Delta_i = (\underline{x}_i - \overline{x}_i)/2$ is its half-width – do not intersect with each other. We also propose, for each positive integer k, an efficient algorithm $\mathcal{A}_k$ that works whenever no more than k "narrowed" intervals can have a common point.

Acknowledgments. This work was supported in part by NASA under cooperative agreement NCC5-209 and grant NCC 2-1232, by NSF grants CDA-9522207, ERA-0112968 and 9710940 Mexico/Conacyt, by the Air Force Office of Scientific Research grant F49620-00-1-0365, and by Grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund.

The authors are greatly thankful to the anonymous referees for very useful suggestions.

References

[1] S. Rabinovich, *Measurement Errors: Theory and Practice*, American Institute of Physics, New York, 1993.

[2] G. W. Walster, "Philosophy and practicalities of interval arithmetic", In: R. E. Moore (ed.), *Reliability in Computing*, 1988, pp. 307–323.

Absolute Bounds on the Mean of Sum, Product, etc.: A Probabilistic Extension of Interval Arithmetic

Scott Ferson[1], Lev Ginzburg[1],
Vladik Kreinovich[2], and Jorge Lopez[2]
[1]Applied Biomathematics, 100 North Country Road,
Setauket, NY 11733, USA, {scott,lev}@ramas.com
[2]Comp. Sci., U. Texas, El Paso, TX 79968, vladik@cs.utep.edu

Abstract

We extend the main formulas of interval arithmetic for different arithmetic operations $x_1 \oplus x_2$ to the case when, for each input x_i, in addition to the interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values, we also know its mean E_i (or an interval $\mathbf{E}_i$ of possible values of the mean), and we want to find the corresponding bounds for $x_1 \oplus x_2$ and its mean.

Error estimation for indirect measurements: an important practical problem. A practically important class of statistical problems is related to data processing (indirect measurements). Some physical quantities y – such as the distance to a star or the amount of oil in a given well – are impossible or difficult to measure directly. To estimate these quantities, we use *indirect* measurements, i.e., we measure some easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y by a known relation $y = f(x_1, \ldots, x_n)$, and then use the measurement results $\widetilde{x}_i$ $(1 \le i \le n)$ to compute an estimate $\widetilde{y}$ for y as $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$. For example, to find the resistance R, we measure current I and voltage V, and then use the known relation $R = V/I$ to estimate resistance as $\widetilde{R} = \widetilde{V}/\widetilde{I}$.

Measurement are never 100% accurate, so in reality, the actual value x_i of i-th measured quantity can differ from the measurement result $\widetilde{x}_i$. In probabilistic terms, x_i is a random variable; its probability distribution describes the probabilities of different possible value of measurement error $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$. It is desirable to describe the error $\widetilde{y} - y$ of the result of data processing.

Often, we know (or assume) that the measurement error Δx_i of each direct measurement is normally distributed with a known standard deviation σ_i, and that measurement errors corresponding to different measurements are independent. These assumptions – justified by the central limit theorem, according to which sums of independent identically distributed random variables with finite

moments tend quickly toward the Gaussian distribution – underly the traditional engineering approach to estimating measurement errors.

In some situations, the error distributions are not Gaussian, but we know their exact shape (e.g., lognormal). In many practical measurement situations, however, we only have *partial* information about the probability distributions.

The need for robust statistics. Traditional statistical techniques deal with the situations when we know the exact shape of the probability distributions. To deal with practical situations in which we only have a partial information about the distributions, special techniques have to be invented. Such techniques are called methods of *robust statistics*. They are called robust because they are usually designed to provide guaranteed estimates, i.e., estimates which are valid for all possible distributions from a given class.

Interval computations as a particular case of robust statistics. An important case of partial information about a random variable x is when we know (with probability 1) that x is within a given interval $\mathbf{x} = [\underline{x}, \overline{x}]$, but we have no information about the probability distribution within this interval. In other words, x may be uniformly distributed on this interval, it may be deterministic (i.e., distributed in a single value with probability 1), distributed according to a truncated Gaussian, bimodal distribution – we do not know.

So, we arrive at the following problem: for each of n random variables $x_1, \ldots, x_n$, we know that it is located (with probability 1) within a given interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$. We do not know the distributions within the intervals, and we do not know whether the random variables x_i are independent or not. What can we then conclude about the probability distribution of $y = f(x_1, \ldots, x_n)$?

Since the only information we have about each variable x_i consists of its lower bound $\underline{x}_i$ and upper bound $\overline{x}_i$, it is natural to ask for similar bounds $\mathbf{y} = [\underline{y}, \overline{y}]$ for y. As a result, we arrive at the following problem:

GIVEN: an algorithm computing a function $f(x_1, \ldots, x_n)$ from R^n to R and n intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$,

TAKE: all possible joint probability distributions on R^n for which, for each i, $x_i \in \mathbf{x}_i$ with probability 1;

FIND: the set $\mathbf{Y}$ of all possible values of a random variable $y = f(x_1, \ldots, x_n)$ for all such distributions.

One can easily prove that $\mathbf{Y}$ is equal to the range $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of the given function f on given intervals, i.e., to $\{f(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, \mathbf{x}_n\}$.

This is exactly the problem solved by interval computations. The main interval computations approach to solving this problem is to take into consideration that inside the computer, every algorithm consists of elementary operations (arithmetic operations, min, max, etc.). For each elementary operation $f(x, y)$, if we know the intervals $\mathbf{x}$ and $\mathbf{y}$ for x and y, we can compute the exact range $f(\mathbf{x}, \mathbf{y})$; the corresponding formulas form the so-called *interval arithmetic*. We

can therefore repeat the computations forming the program f step-by-step, replacing each operation with real numbers by the corresponding operation of interval arithmetic. It is known that, as a result, we get an enclosure for the desired range.

Comment. In the above text, we considered the case when we have no information about the correlation between the random variables. We have proven that in the above problem, if we assume independence, we still get the same range.

For functions of two variables, we can consider two additional cases: when x_1 and x_2 are highly positively correlated (i.e., crudely speaking, that x_1 is (non-strictly) increasing in x_2, and when x_i is highly negatively correlated (i.e., when x_1 is decreasing in x_2). In both cases, we get the same range $\mathbf{Y}$ as in the above case of no information about the correlation.

New problem. In some practical situations, in addition to the lower and upper bounds on each random variable x_i, we know the bounds $\mathbf{E}_i = [\underline{E}_i, \overline{E}_i]$ on its mean E_i. In such situations, we arrive at the following problem:

GIVEN: an algorithm computing a function $f(x_1, \ldots, x_n)$ from R^n to R; n intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and n intervals $\mathbf{E}_1, \ldots, \mathbf{E}_n$,

TAKE: all possible joint probability distributions on R^n for which, for each i, $x_i \in \mathbf{x}_i$ with probability 1 and the mean E_i belongs to $\mathbf{E}_i$;

FIND: the set $\mathbf{Y}$ of all possible values of a random variable $y = f(x_1, \ldots, x_n)$ and the set $\mathbf{E}$ of all possible values of $E[y]$ for all such distributions.

A similar problem can be formulated for the case when x_i are known to be independent, and for the cases when $n = 2$ and the values x_i are highly positively or highly negatively correlated.

If we can find the range for degenerate intervals $\mathbf{E}_i = [E_i, E_i]$, then we can use interval computation to extend these formulas to arbitrary intervals $\mathbf{E}_i$.

Similarly to interval computations, our main idea is to find the corresponding formulas for the cases when $n = 2$ and $f = \oplus$ is one of the basic arithmetic operations ($+$, $-$, $\cdot$, min, max). For example, if we know two "triples" $(\underline{x}_i, E_i, \overline{x}_i)$, $(i = 1, 2)$, what are the possible triples $(\underline{y}, E, \overline{y})$ for $y = x_1 \cdot x_2$?

Main results. For all basic operations, the interval part $(\underline{y}, \overline{y})$ of the result is the same as for interval arithmetic.

We provide explicit formulas for the interval $\mathbf{E}$ of possible values of $E = E[y]$. For example, for multiplication, when we know nothing about the correlation,

$$\overline{E} = \min(p_1, p_2) \cdot \overline{x}_1 \cdot \overline{x}_2 + \max(p_1 - p_2, 0) \cdot \overline{x}_1 \cdot \underline{x}_2 + \max(p_2 - p_1, 0) \cdot \underline{x}_1 \cdot \overline{x}_2 +$$

$$\min(1 - p_1, 1 - p_2) \cdot \underline{x}_1 \cdot \underline{x}_2,$$

where $p_i \stackrel{\text{def}}{=} (E_i - \underline{x}_i)/(\overline{x}_i - \underline{x}_i)$.

Convex–Concave Extensions for Polynomials

Jürgen Garloff[1], Andrew P. Smith[1], and Christian Jansson[2]

[1]University of Applied Sciences / FH Konstanz
Konstanz, Germany
[2]Technical University Hamburg–Harburg
Hamburg, Germany

A frequently used approach for solving nonlinear systems, combinatorial optimization, or constrained global optimization problems is the generation of relaxations and their use in a branch and bound framework. Generally speaking, a relaxation of a given problem has the properties that

(i) each feasible point of the given problem is feasible for the relaxation,

(ii) the relaxation is easier to solve than the given problem, and

(iii) the solutions of the relaxation converge to the solutions of the original problem, provided the maximal width of the set of feasible points converges to zero.

For many problems a relaxation can be constructed, if the functions which define the problem can be bounded from below by affine or, more generally, by convex functions.

In our talk we address the construction of convex lower bounding and equally concave upper bounding functions for multivariate polynomials. Both functions together constitute a so-called *convex–concave extension*. For polynomials this is obtained in a natural way if we represent the given polynomial (for simplicity we consider here only the univariate case and concentrate on the unit interval $I = [0, 1]$)

$$p(x) = \sum_{i=0}^{n} a_i x^i$$

in its Bernstein form

$$p(x) = \sum_{i=0}^{n} b_i B_i(x)$$

where the

$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i},\ i = 0, 1, \ldots, n$$

are the Bernstein polynomials. The coefficients of this expansion, the so-called Bernstein coefficients, can easily be computed from the coefficients of p:

$$b_i = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{n}{j}} a_j, \; i = 0, 1, \ldots, n \quad \text{(note that } b_0 = p(0), b_n = p(1)\text{)}.$$

A fundamental property of the Bernstein expansion is its *convex hull property*

$$\left\{ \begin{pmatrix} x \\ p(x) \end{pmatrix} \; : \; x \in I \right\} \subseteq conv \left\{ \begin{pmatrix} \frac{i}{n} \\ b_i \end{pmatrix} \; : \; i = 0, 1, \ldots, n \right\}$$

which states that the graph of p over I is contained in the convex hull (denoted by *conv*) of its control points. Based on this property, convex–concave extensions of increasing complexity can be constructed (we are giving here only the construction of the lower bounding function). E.g., we obtain an affine lower bounding function if we consider the straight line which passes through a facet of the lower part of the convex hull of the control points, the slope of which is given by the absolute value of the slope between the control points associated with the smallest and next to smallest Bernstein coefficients. A convex lower bounding function is provided by the lower part of the convex hull of the control points.

In Figures 1 and 2 convex–concave extensions for a polynomial of fourth degree over the intervals $[0, 0.5]$, $[0, 0.6]$, $[0, 0.7]$ and $[0, 1]$ are displayed. In Fig. 1 the extension is based on one affine upper and lower bounding function. The figure shows that this convex–concave extension is not inclusion isotone. In Fig. 2 the extension is provided by the convex hull of the control points. In this special example the convex hull is inclusion isotone. We show that this property holds generally. However, it should be noted that inclusion isotonicity is not a necessary prerequisite for constructing and using convex–concave extensions.

In the multivariate case the affine lower bounding function c can be characterized as the optimal solution of a linear programming problem. We present an upper bound for the difference $p - c$ which exhibits in the univariate case quadratic convergence with respect to the width of the interval.

Due to rounding errors, inaccuracies may be introduced into the calculation of the Bernstein coefficients and therefore of the bounding functions. This may lead to erroneous results in applications. We are giving some suggestions for the way in which the calculations have to be performed so that verified results are obtained.

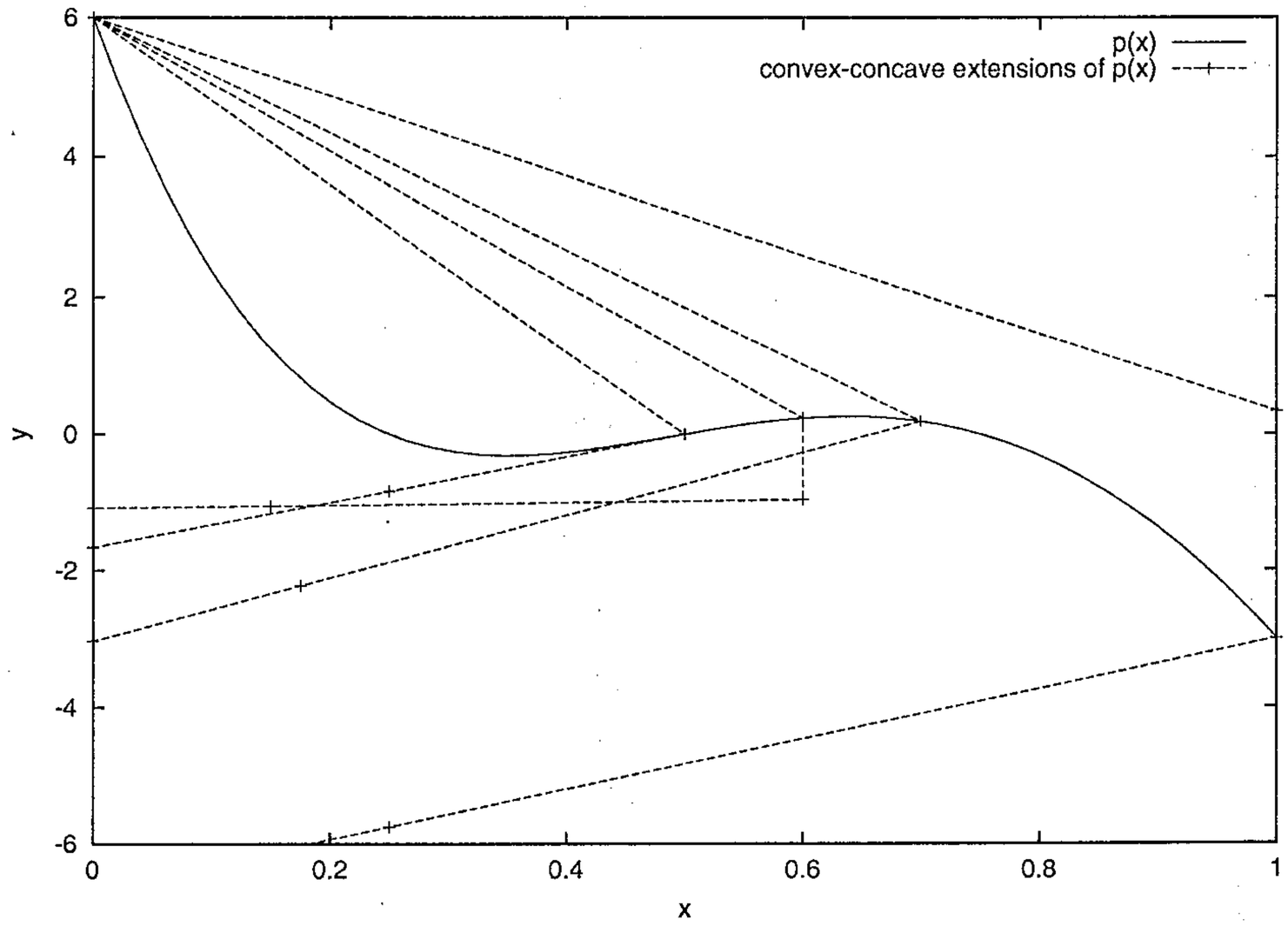


Fig. 1. Failure of inclusion isotonicity with one affine function.

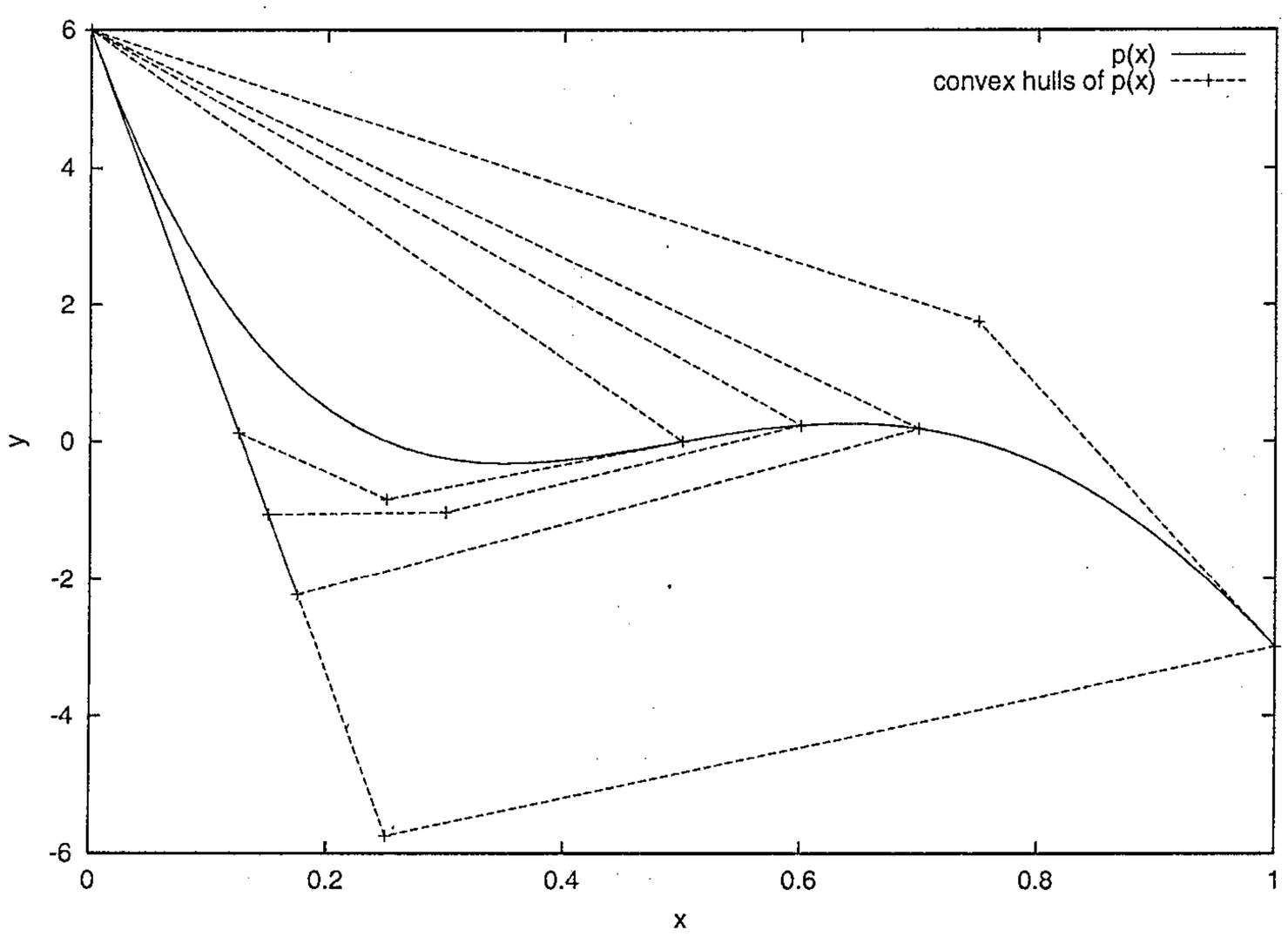


Fig. 2. The convex hull is inclusion isotone.

Ontologies for Continuous Global Optimization

Laurent Granvilliers and Mina Ouabiba
IRIN – University of Nantes
B.P. 92208 – 44322 Nantes cedex 3 – France
e-mail: {granvilliers, ouabiba}@irin.univ-nantes.fr

Global Optimization tackles the problem of finding all feasible points of a set of constraints that optimize an objective function. In the following, we restrict our attention to Continuous Global Optimization, and more specifically to Global Optimization over interval domains [6, 9].

We believe that Software Design for Continuous Global Optimization has things in common with Astronomy of Sixteenth Century. Many powerful systems implementing various solving techniques have already been developed. However, there exists no general cooperation architecture [4] describing knowledge sharing among solvers. More precisely, a cooperation model would define representations and specifications of shared knowledge, and protocols of communication.

In Knowledge Engineering, the problem of representing shared and reusable knowledge among software agents has heavily been studied in the recent past. This problem can be addressed by ontologies. An *ontology* [5] is a specification of concepts of a given domain and relationships among them. One of the objectives is to lay foundations for libraries of reusable components and knowledge sharing functionalities.

Knowledge based systems (KBS) are often modeled by means of three concepts: task, PSM (Problem Solving Method) and domain [8]:

- Domain describes the knowledge of a particular domain, e.g., global optimization.
- Tasks define problems that should be solved, e.g., constraint solving.
- PSMs define resolution processes of problems, e.g., LP techniques.

The relation between them can be specified by means of semantic links:

- Intra-concept link for the relationships between two identical concepts task/task, PSM/PSM and domain/domain.
- Inter-concept link for two different concepts. Such a link can be used to transfer information between them.

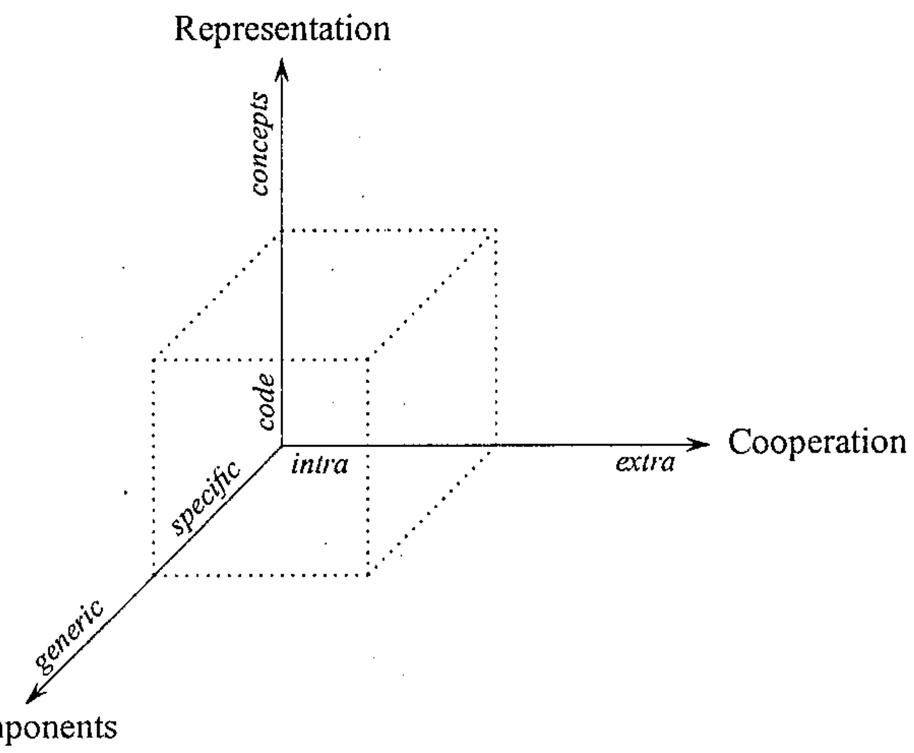


Figure 1: Objectives of Ontologies Design for Solver Cooperation.

A real-world application can be seen as a composition of specific components described by means of tasks, PSMs, domain, intra-concept and inter-concept links. In the optimization domain, task ontologies describe classes of optimization problems and relationships between them.

For example, Glopt [3] is a PSM of some specific problems (tasks) defined by a block-separable objective function subject to bound constraints or block-separable constraints. In that case, task/PSM inter-concept links transfer the specific format of the problem structure called NOP[1]. Glopt implements branch-and-bound technique (PSM) which is often used by many other problems. As a consequence, there is a need for describing a generic branch-and-bound algorithm which can be specialized by means of PSM/PSM intra-concept links.

In this work, we are designing ontologies for the domain of Continuous Global Optimization. This research is part of the COCONUT project from the European Community. We have noticed that most of existing systems and platforms for Continuous Global Optimization implement specific and intra-solver cooperations (dotted box of Figure 1): no heterogeneous encoding of data structures, one cooperation concept, *e.g.*, sequential or concurrent, brick solvers and routing of data fixed *a priori*. We have then identified two directions for future research [2]:

- Modeling of extra-solver cooperations, at two-levels: knowledge sharing and strategies of application of solvers.
- Extraction and definition of generic and reusable components.

Part of this research was concerned by the definition of an ontology for a specific intra-solver cooperation: constraint solving using Branch-and-Prune

[1] a compact format for specifying general constrained nonlinear optimization problems.

Algorithms based on Interval Constraint Satisfaction Techniques [10]. This work has led to the implementation of a C++ library of reusable components and generic strategies. In this extended abstract, we briefly describe design decisions supported by the ontology.

Branch-and-Prune Algorithms are generic in essence; they alternate domain pruning by enforcing local consistency techniques and interval computations [7], and branching to traverse the search space and exhibit all the solutions. Four levels of genericity have been identified and represented in our library:

- **Genericity w.r.t. Interval Arithmetic** (IA). IA is used to compute reliable approximations of ranges of real functions. The evaluation of functions needs to be independent of the IA library in order to plug in extern IA libraries such as `Bias`, `Jail` and `Sun's Forte`. Another motivation is to easily interchange libraries for solving a problem, *e.g.*, for the use of a multi-precision library such as `Mpfi` for handling instable numerical computations. The interval data type is implemented by the *traits* C++ technique defining all services required by Branch-and-Prune Algorithms.

- **Genericity w.r.t. interval extensions**. The truth value of interval constraints is computed in two consecutive steps: evaluation of function expressions using interval extensions, and interpretation of relation symbols. Interval constraints are parameterized by data types for interval extensions and interpretation of relation symbols. Note that for standard constraints, the interpretation of relation symbols is given by the IA library. This mechanism allows one to define constraints for non standard IA, *e.g.*, modal IA, with no additionnal cost.

- **Genericity w.r.t. domain pruning methods**. This is a main task in itself since there exist various kinds of algorithms. The ontology is based on chaotic iteration framework for constraint propagation [1] which has been adapted for event-based programming. Doma in pruning is modeled as an iteration of reduction functions over interval do mains. Areduction function models either a box consistency operator, or an interval Newton operator, *etc.* Scheduling of reduction functions depends on priorities and properties of functions, *etc.* Strong consistencies are described by strategies combining local splitting and domain pruning.

- **Genericity w.r.t. branching strategies**. Branching strategies are represented by three components: a strategy for selecting the next variable domain to be bisected, a method for splitting domains, and a mechanism for managing memorization of domains (copying or trailing).

Such a generic library has many advantages in terms of flexibility, maintenance, reuse of code, prototyping of strategies, cooperation, *etc.* We believe that this research on Branch-and-Prune Algorithms can be extended for the wider domain of Continuous Global Optimization.

Acknowledgments. This work has been partly supported by the COCONUT project IST-2000-26063 from the European Community.

References

[1] K. R. Apt, "The Essence of Constraint Propagation", *Theoretical Computer Science*, 1999, Vol. 221, No. 1/2, pp. 179–210.

[2] The COCONUT Consortium, *Algorithms for Solving Nonlinear Constrained and Optimization Problems: the State of The Art*, 2001. Deliverable 2 of the COCONUT project IST-2000-26063 from the European Community.

[3] S. Dallwig, A. Neumaier, and H. Schichl, "GLOPT – a program for constrained global optimization", In: I. Bomze et al. (eds.), *Developments in Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1997, pp. 19–37.

[4] L. Granvilliers, E. Monfroy, and F. Benhamou, "Symbolic-Interval Cooperation in Constraint Programming", In: *Proceedings of ISSAC'2001, 26th International Symposium on Symbolic and Algebraic Computation*, University of Western Ontario, Canada, 2001, pp. 150–166.

[5] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", In: N. Guarino and R. Poli (eds.), *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, 1993.

[6] E. R. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker Inc., 1992.

[7] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1966.

[8] E. Motta, *Reusable Components for Knowledge Modelling*, IOS Press, Amsterdam, 1999.

[9] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.

[10] P. Van Hentenryck, L. Michel, and Y. Deville, *Numerica: a Modeling Language for Global Optimization*, MIT Press, 1997.

Fast or Tight Propagation of Univariate Taylor Coefficients

Andreas Griewank

It is by now well understood that, for functions defined by computer programs Taylor, polynomials in several variables can be recovered from families of univariate polynomials. We observe here that by adapting an idea of Kronecker it is possible to get by with a single univariate polynomial of rather high order.

Hence the evaluation of higher derivatives can be based on the propagation of univariate Taylor polynomials though the sequence of arithmetic functions and elementary intrinsics defining the function at hand. This can be done with very regular memory access patterns, and using Newton's method or other fixed point iterations everything can be reduced to a small number of convolutions.

Thus the convolution becomes the work horse of higher order differentiation. It may be performed either optimal in the interval sense with a quadratic complexity or fast in essentially linear time using FFT. Either way one achieves significant improvements on the basic recurrences due to Moore, but so far we have not found a "best of both worlds" i.e., fast and tight propagation method.

A Brief History of Interval Analysis

Eldon R. Hansen

In this talk, we discuss the origin and early history of interval analysis. We describe Moore's early work and the work of others that Moore influenced directly. We discuss highlights and milestones in the development of interval analysis. Some topics are:

- The origins of interval arithmetic
- Alternatives to intervals (circular arithmetic, ellipsoids, parallelopipeds)
- Extended interval arithmetic
- The development of means for using interval arithmetic
  - subroutines, precompilers, compilers, languages
- Moore's early work
  - concepts, terminology, and properties of interval arithmetic
  - theorems
  - application of interval analysis to various problem areas
- Early reports, papers, and books
- People who influenced the development of interval analysis
- Interval analysis in Germany
- The development of tools for use in interval computations
  - evaluation of irrational functions
  - representation of interval functions (centered forms, etc.)
  - Taylor expansions, automatic differentiation
  - slopes, generalized interval arithmetic
- The development of interval methods in various specific problem areas
  - linear algebra (Gaussian elimination, preconditioning, eigensolutions, etc.)
  - nonlinear equations (methods, existence and uniqueness, etc.)

- optimization
- ordinary differential equations (initial value and boundary value problems)
- partial differential equations

Training Feedforward Multilayer Interval Artificial Neural Networks

Chenyi Hu
University of Houston-Downtown
Houston, Texas 77002, USA

Ever since the mid-1980's, artificial neural networks have been applied to solve various application problems such as pattern recognition, classification, control, and decision making with imprecise input data. An artificial neural network (ANN) is an information-processing paradigm inspired by the way of human brain processes information. It is composed of a large number of highly interconnected processing elements that are analogous to neurons and are tied together with weighted connections that are analogous to synapses. Training ANNs is similar to learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. Through exposure to a trusted set of input/output data, an ANN adjusts its connection weights (synapses). These connection weights store the knowledge necessary to solve application problems. Therefore, training ANNs is the most critical part of their applications.

In this presentation, we specifically report our work on applying interval methods in supervised training of multilayer feedforward ANNs. We will review the related basic concepts of artificial neural networks at the beginning. Through analyzing the mathematical model of training multilayer feedforward ANNs, we categorize the training problem into two types. Since data sets that used to train a neural network are usually imprecise, they should be better represented with intervals rather than real numbers. Therefore, we have interval valued neural networks. Finally, we will report our result that training a interval weighted multilayer feedforward artificial neural network can be reliably solved by currently available interval algorithms and software packages.

Validation of Feasible Operating Region in Chemical Processes

Haitao Huang

Studies on process safety and flexibility are two major issues for chemical process design. Analysis of process safety that can be ensured within some operating space is mathematically the same as analysis of process feasibility being maintained in certain operating boundaries. Therefore, although they are usually treated separately in research and practice, they share a basic feature: searching feasible operating space from possible design space.

At present, with classic methods, it is impossible to identify the accurate feasible region when the region is in a shape other than hyper-rectangle. To solve the problem, this work begins with the identification of a new modelling relationship, region to region, and then a new concept of region model framework against traditional point model framework for the new relationship is proposed. The basic component in the new framework is a simple region that is a hyper-cube formed by range values of the considered process variables and parameters. The dimensions of a region is mathematically represented by interval vector. The new model also has other advantages such as uncertainty information description and non-linear process behavior modelling.

Secondly, methodologies for region simulation are proposed. One main feature is the proposed bounding strategies. Unlike point simulation, which is done by solving differential and algebraic equations, region simulation is performed through the proposed bounding strategies. For example, for linear problems, natural inclusion of interval analysis is directly applied for region bounding.

The other main feature is the region transition model. Hybrid systems appear commonly in chemical processes especially when closed loop controllers are employed. In a point model, this is dealt with hybrid state transition network. In the region model, as sets of conditions rather than point conditions are considered, new features are introduced. The region transition model is devised in the work to deal with these new features.

Thirdly, based on the concept of region and methodologies of region simulation, an algorithm for searching the feasible space in possible design space is devised. It is illustrated by several case studies that with this new model we are able to get the accurate shape and size of feasible region within acceptable tolerance. Since closed-loop controller application is one of the main features in chemical processes, fourthly, strategies for dealing with dynamic process disturbances and modelling of closed-loop controllers in the region model are proposed.

Finally, an optimisation formulation is proposed for maximising the size of feasible region by tuning control parameters. As shown in case studies, quantitative process safety analysis and flexibility studies are unified in the new model. Since the model is able to identify accurate feasible region, it facilitates inherent safe process design. The framework has been implemented in an object-oriented architecture that makes it easier to represent a system than the equation-oriented approaches.

On Existence and Uniqueness Verification for Non-Smooth Functions

R. Baker Kearfott
Department of Mathematics
University of Louiisiana at Lafayette
U.L. Box 4-1010
Lafayette, Louisiana 70504-1010
USA
email: rbk@louisiana.edu

1 Summary

It is known that interval Newton methods can verify existence and uniqueness of solutions of a nonlinear system of equations near points where the Jacobi matrix of the system is not ill-conditioned. Recently, we have shown how to verify existence and uniqueness, up to multiplicity, for solutions at which the Jacobi matrix is singular. We do this by efficient computation of the topological index over a small box containing the approximate solution. Algorithmically, our techniques mimic the non-singular case (both in algorithmic steps and computational complexity), and can be considered as *incomplete Gauss–Seidel sweeps.*

Since the topological index is defined and computable when the Jacobi matrix is not even defined at the solution, one may speculate that efficient algorithms can be devised for verification in this case, too. In this talk, we discuss, through examples, key techniques underlying our simplification of the calculations that cannot necessarily be used when the function is non-smooth. We also suggest when degree computations involving non-smooth functions may be practical.

Our examples also shed light on our published work on verification involving the topological degree.

2 Introduction and Some Details

Given a system of nonlinear equations $F(x) = 0$, numerical methods produce an approximation $\check{x}$ to a solution x^*. It is then sometimes desirable to compute bounds

$$\begin{aligned} \boldsymbol{x} &= (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \\ &= ([\underline{x}_1, \overline{x}_1], [\underline{x}_2, \overline{x}_2], \ldots, [\underline{x}_n, \overline{x}_n], \end{aligned}$$

such that $\check{x}$ is the center of $\boldsymbol{x}$, and such that $\boldsymbol{x}$ is guaranteed to contain a solution x^* to $F(x) = 0$. This leads to the problem

> Given $F : \boldsymbol{x} \to \mathbb{R}^n$, where $\boldsymbol{x} \in \mathbb{IR}^n$, *rigorously* verify:
>
> - there exists a $x^* \in \boldsymbol{x}$ such that $F(x^*) = 0$.

(1)

Here, $\mathbb{IR}^n$ represents the set of n-dimensional vectors, as $\boldsymbol{x}$, whose components are intervals.

If the Jacobi matrix $F'(x^*)$ is non-singular and continuous in $\boldsymbol{x}$, then we can use interval Newton methods to verify existence and uniqueness of $x^* \in \boldsymbol{x}$, $F(x^*) = 0$; see [3, Chapter 8], [4, pp. 219–223], and the references therein. Such interval Newton methods are of the form

$$\tilde{\boldsymbol{x}} = \boldsymbol{N}(F; \boldsymbol{x}, \check{x}) = \check{x} + \boldsymbol{v}, \tag{2}$$

where

$$\Sigma(\boldsymbol{A}, -F(\check{x})) \subset \boldsymbol{v}, \tag{3}$$

where $\boldsymbol{A}$ is a Lipschitz matrix for F over $\boldsymbol{x}$, and where

$$\Sigma(\boldsymbol{A}, -F(\check{x})) = \{x \in \mathbb{R}^n \mid \exists A \in \boldsymbol{A} \text{ with } AX = -F(\check{x})\}. \tag{4}$$

Here $\check{x}$ is some point in $\boldsymbol{x}$ (often taken to be its midpoint) that, in the context of this paper, we consider to be an approximate solution.

Theorem 1 *([4, Theorem 1.19, p. 62], originally from [8]) Suppose $\tilde{\boldsymbol{x}} = \boldsymbol{N}(F; \boldsymbol{x}, \check{x})$ is the image of $\boldsymbol{x}$ and $\check{x}$ under an interval Newton method. If $\tilde{\boldsymbol{x}} \subseteq \boldsymbol{x}$, it follows that there exists a unique solution of $F(x) = 0$ within $\boldsymbol{x}$.*

Recently, we have developed techniques that can verify existence of solutions to $F(x) = 0$ within $\boldsymbol{x}$, even when $F'(x) = 0$ for some $x \in \boldsymbol{x}$. These techniques are based on computing the *topological degree* $\mathrm{d}(F, \boldsymbol{x}, 0)$ of F over $\boldsymbol{x}$. If every $x \in \boldsymbol{x}$ where $F(x) = 0$ has the Jacobi matrix $F'(x)$ nonsingular, then $\mathrm{d}(F, \boldsymbol{x}, 0)$ is equal to the number of solutions of $F(x) = 0$ in $\boldsymbol{x}$ at which the determinant of $F'(x)$ is positive, minus the number of solutions of $F(x) = 0$ in $\boldsymbol{x}$ at which the determinant is negative. However, the integer $\mathrm{d}(F, \boldsymbol{x}, 0)$ is both continuous in F and depends only on values of F on the boundary $\partial\boldsymbol{x}$. Thus, in theory, F' may be singular, and indeed, even non-smooth, in the interior $\mathrm{int}(\boldsymbol{x})$.

Our recent work, as other work dealing with the topological degree, depends on a basic formula that relates the topological degree to solutions of a derived system over the boundary of $\boldsymbol{x}$. In contrast to previous literature on computing

the topological degree, in our recent work in [1], [7], [5], [6], and [2], we are not given a large box $\boldsymbol{x}$, but we construct $\boldsymbol{x}$ sufficiently small to allow us to use a local model of F to both reduce the dimension of the search on the boundary and to greatly speed the resulting low-dimensional search. The process includes

1. preconditioning the system,
2. applying a local model to the preconditioned system to reduce the dimension, and
3. using a local model to predict where the solutions to the derived system are on $\partial \boldsymbol{x}$.

Our analysis indicates the verification proceeds in $\mathcal{O}\left(n^3\right)$ time for rank-defect 1 Jacobi matrices; this order has been verified experimentally with solutions to finite discretizations of a model problem with n up to 320.

The question we ask here is: "Can we do similar simplifications and devise a successful algorithm if F is defined in a piecewise fashion, or is otherwise non-smooth?" We consider several examples.

See `http://interval.louisiana.edu/preprints/nonsmooth_degree.ps` or `http://interval.louisiana.edu/preprints/nonsmooth_degree.pdf` for a preprint that contains additional details.

References

[1] J. Dian, *Existence Verification of Higher Degree Singular Zeros of Nonlinear Systems*, PhD thesis, University of Louisiana at Lafayette, 2000.

[2] J. Dian and R. B. Kearfott, *Existence verification for singular and non-smooth zeros of real nonlinear systems*, 2001.

[3] E. R. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., New York, 1992.

[4] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, Netherlands, 1996.

[5] R. B. Kearfott and J. Dian, *Existence verification for higher-degree singular zeros of complex nonlinear systems*, 2000, preprint, `http://interval.louisiana.edu/preprints/degree_cplx.0302.pdf`.

[6] R. B. Kearfott and J. Dian, "Verifying topological indices for higher-order rank deficiencies", *Journal of Complexity Theory* (to appear).

[7] R. B. Kearfott, J. Dian, and A. Neumaier, "Existence verification for singular zeros of complex nonlinear systems", *SIAM J. Numer. Anal.*, 2000, Vol. 38, No. 2, pp. 360–379.

[8] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, England, 1990.

C–XSC 2.0: A C++ Class Library for Extended Scientific Computing

Walter Krämer and Werner Hofschuster
University of Wuppertal, Germany

The original version of the C++ class library C–XSC [4] is about ten years old. But in the last decade the underlying programming language C++ has been developed significantly. Since November 1998 the C++ standard [3] is available and more and more compilers support (most of) the features of this standard. The new version C–XSC 2.0 [2] conforms to the C++ standard.

For those who are not familiar with C–XSC let us first motivate the library by quoting essential parts (with slight modifications) from the preface of the book [4]: The programming environment C–XSC (C for eXtended Scientific Computing) is a powerful and easy to use programming tool, especially for scientific and engineering applications. C–XSC makes the computer more powerful arithmetically and significantly simplifies programming in the field of scientific computing (especially in the field of interval mathematics occasionally called mathematical numerics[1]). C–XSC is implemented as a numerical class library in the programming language C++ .

The speed of digital computers is ever increasing. But at processor speeds of gigaFLOPS it is a significant question whether floating-point arithmetic, which may fail already in simple calculations, is still adequate to be used in computers for huge problems.

Mathematicians have contrived algorithms which deliver highly accurate and automatically verified results by dappling mathematical fixed-point theorems. This means that these computations carry their own accuracy control. However, their implementations require suitable arithmetic support and powerful programming tools which were not previously available. The development of C–XSC has aimed at providing these tools within C++. C–XSC is particularly suited for the development of numerical algorithms that deliver highly accurate and automatically verified results, which are essential, for example, in simulation runs where the user has to distinguish between computational artifacts and genuine reactions of the model. Problem-solving functions with automatic result verification have been developed in C–XSC for several standard problems of numerical analysis.

[1] In contrast to Numerical Mathematics, where results are sometimes merely speculative, Mathematical Numerics delivers solutions of numerical problems that are mathematically proven to be correct.

C–XSC consists of a run time system written in ANSI C and C++ including an optimal dot product and many predefined data types for elements of the most commonly used vector spaces such as real and complex numbers, vectors, and matrices. Operators for elements of these types are predefined and can be called by their usual operator symbols. Thus, arithmetic expressions and numerical algorithms are expressed in a notation that is very close to the usual mathematical notation. C–XSC allows to write verification algorithms in a way which is very near to pseudo-code used in scientific publications. All predefined numerical operators are of highest accuracy. That is, the computed result differs from the correct result by at most one rounding.

While the emphasis in computing is traditionally on speed, in C–XSC, the emphasis is more on accuracy and reliability of results. The total time for solving a problem is the sum of the programming effort, the processing time, and the time for the interpretation of results. We contend that C–XSC reduces this sum considerably.

C++ programmers should be able to use and write programs in C–XSC immediately. C–XSC simplifies programming by providing many predefined data types and arithmetic operators. Programs are much easier to read, to write, and to debug.

What is new in C–XSC 2.0?

- All routines are now in the namespace `cxsc`
- Explicit typecast constructors
- Constant values passed by reference are now passed by const reference
- The error handling is done according to the C++ error handling using exception classes
- Modification in the field for subvectors and submatrices
- The library uses templates extensively
- The source code of C-XSC 2.0 is freely available from
  `http://www.math.uni-wuppertal.de/~xsc/xsc/download.html`
- Older C–XSC programs have to be modified slightly to run with C–XSC 2.0. We will discuss this point in our talk
- The source code of a new version of the C++ Toolbox for Verified Computing [1] which works with C-XSC 2.0 is freely available from
  `http://www.math.uni-wuppertal.de/~xsc/xsc/download.html`

The following areas are covered by the **C++ Toolbox for Verified Computing** [1] (Its current version is freely available and runs with C–XSC 2.0):

- One-dimensional problems: Extended interval division, Evaluation of polynomials, Automatic differentiation, Nonlinear equations in one variable, Global optimization, Accurate evaluation of arithmetic expressions, Zeros of complex polynomials.

- Multi-dimensional problems: Linear systems of equations, Linear optimization, Automatic differentiation for gradients, Jacobians, and Hessian, Nonlinear systems of Equations, Global optimization, Initial value problems in ordinary differential equations[2].

Additional self-verifying problem solving routines based on C–XSC 2.0 are available (e.g., a slope arithmetic in forward and in reverse mode, numerical quadrature and cubature [9], validated bounds for Taylor coefficients [5]; for downloads see
`http://www.math.uni-wuppertal.de/~xsc/xsc/cxsc_software.html`) or will be made available. Further developments will be discussed in our talk. We will also comment on other interval tools, especially on INTLAB [6] and Sun compilers [7] supporting intervals. We will see that so called containment computations [8] can be used, e.g., to compute (with rather small programming effort) enclosures of all roots of functions over finite and infinite domains.

Acknowledgments: Many colleagues and scientists (see [2] Paragraph 1) have directly and indirectly contributed to the realization of C–XSC and C–XSC 2.0. The authors would like to thank each of them for his or her cooperation.

Keywords: C–XSC, C++ Class Library, Interval Mathematics, Validated Numerics, Enclosure Methods, INTLAB, Containment Computations

References

[1] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz, *C++ Toolbox for Verified Computing: Basic Numerical Problems*, Springer-Verlag, Berlin, 1995.

[2] W. Hofschuster, W. Krämer, S. Wedner, and A. Wiethoff, *C–XSC 2.0: A C++ Class Library for Extended Scientific Computing*, University of Wuppertal, 2001, pp. 1–24.

[3] ISO/IEC 14882: *Standard for the C++ Programming Language*, 1998.

[4] R. Klatte, U. Kulisch, C. Lawo, M. Rauch, and A. Wiethoff, *C–XSC – A C++ Class Library for Scientific Computing*. Springer-Verlag, Berlin, 1993.

[5] M. Neher, "Validated Bounds for Taylor Coefficients of Analytic Functions", *Reliable Computing*, 2001, Vol. 7, No. 4.

[6] S. M. Rump, "INTLAB –INTerval LABoratory", in T. Csendes (ed.), *Developments in Reliable Computing*, Kluwer, 1999.

[2] Available from R. Lohner (see http://www.uni-karlsruhe.de/~Rudolf.Lohner/)

[7] Sun Forte[TM] Developer 6 update 2 compilers (see `http://www.sun.com/forte/index.html`)

[8] G. W. Walster, et al., *The "Simple" Closed Interval System*, Sun Microsystems, Feb 2000.

[9] S. Wedner, *Verifizierte Bestimmung singulärer Integrale – Quadratur und Kubatur*. Dissertation, Universität Karlsruhe, 2000.

Extended Interval Power Function

W. Krämer and J. Wolff v. Gudenberg

The general power function,

$$\hat{} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$
$$(x, y) \mapsto \mathrm{x\hat{}y} := x^y$$

is not defined for $x < 0$.

On the other hand, well-known formulas exist for $y \in \mathbb{Z}$ or for some $y \in \mathbb{Q}$.

In the framework of an extended interval arithmetic computing containment sets for every function these values should be included in the range, if the domain is accordingly extended.

$$[-2,2]^{[0.9,1.1]} \supseteq [-2,2]^{[1.0,1.0]} \supseteq [-2,2]$$

One goal of containment arithmetic is to provide an exception free evaluation of functions over an arbitrary range. In current libraries like Sun's [1] or ours [2] the power function is defined for positive radicands only. As a consequence the result with the given sample values is

$$[-2,2]^{[0.9,1.1]} = [0,2]^{[0.9,1.1]} = [0,2^{1.1}] = [0,2.14]$$

In this paper we discuss alternative implementations of the power function, compare them with computer algebra systems, develop containment sets and discuss the issue of accuracy.

References

[1] Sun Microsystems, C++ *Interval Arithmetic Programming Reference*, October 2000
`http://www.sun.com/forte/cplusplus/interval/index.html`

[2] W. Hofschuster et al., *The Interval Library* `fi_lib++ 2.0`, *Design, Features and Sample Programs*, Preprint 2001/4, Universität Wuppertal, Dec. 2001
`http://www.math.uni-wuppertal.de/wrswt/literatur.html`

[3] G. W. Walster, E. R. Hanson, and J. D. Pryce, *Extended Real Intervals and the Topological Closure of Extended Real Numbers*, June 1999.

Are There Efficient Necessary and Sufficient Conditions for Straightforward Interval Computations To Be Exact?

Vladik Kreinovich[1], Luc Longpré[1], and James J. Buckley[2]
[1]Department of Computer Science, U. Texas at El Paso
El Paso, TX 79968, USA, {vladik,longpre}@cs.utep.edu
[2]Math., U. of Alabama at Birmingham, buckley@math.uab.edu

One of the main problems of interval computations is to find a range of a given function on given intervals. To be more precise: given n input intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and an algorithm $f(x_1, \ldots, x_n)$ that transforms n real numbers $x_1, \ldots, x_n$ into a real number $y = f(x_1, \ldots, x_n)$, find the range

$$\mathbf{y} = f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \{f(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

Usually, the endpoints of the intervals $\mathbf{x}_i$ come from measurements, and measurement usually produces rational numbers, so we can assume that the intervals $\mathbf{x}_i$ have rational endpoints. If we cannot compute the exact range, we can at least try to find an enclosure $\mathbf{Y} \supseteq \mathbf{y}$ for the range.

Straightforward interval computations: its advantages and drawbacks. Historically the first method for computing the enclosure for the range is the method which is sometimes called "straightforward" interval computations. This method is based on the fact that inside the computer, every algorithm consists of elementary operations (arithmetic operations, min, max, etc.). For each elementary operation $f(x, y)$, if we know the intervals $\mathbf{x}$ and $\mathbf{y}$ for x and y, we can compute the exact range $f(\mathbf{x}, \mathbf{y})$. The corresponding formulas form the so-called *interval arithmetic.* In straightforward interval computations, we repeat the computations forming the program f step-by-step, replacing each operation with real numbers by the corresponding operation of interval arithmetic. It is known that, as a result, we get an enclosure for the desired range.

In some important cases, the enclosure obtained by using straightforward interval computations is actually the exact range. There are several sufficient conditions for straightforward interval computations to be exact: e.g., it is exact when $f(x_1, \ldots, x_n)$ is an explicit expression in which each variable occurs only once; another condition is given by Hansen in his 1997 RC paper.

However, there are known cases when the resulting enclosure is much larger than the actual range. For example, for the expression $f(x_1, x_2) = x_1 + x_1 \cdot x_2$, straightforward interval computations are exact when $\underline{x}_2 \geq 0$ and not exact when, e.g., $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$ is a non-degenerate interval and $\mathbf{x}_2 = [-1, -1]$. Indeed, in the second case, $f(x_1, x_2) = 0$, so we have a 1-point range $[0, 0]$, but straightforward interval computations result in $[\underline{x}_1 - \overline{x}_1, \overline{x}_1 - \underline{x}_1]$.

More sophisticated methods and the first methodological question. Several methods have been proposed to reduce the overestimation: centered form, bisection, monotonicity check, etc. E.g., Hansen's generalized interval arithmetic takes into account dependence between interval variables and thus, computes the range of $x_1 + x_1 \cdot (-1)$ as $[0, 0]$.

Each new method improves the enclosures, often reducing the enclosure to the exact range, but for each known method, there are cases when this method still overestimates.

In such situations, when many methods have been proposed and none of them is perfect, a natural question is: *Is a perfect method* – that would always return the exact range in reasonable time – *possible at all?* This methodological question is important for algorithm designers:

- If a perfect method is possible, then it is reasonable to spend some time looking for it.
- On the other hand, if such a method is not possible at all, then looking for a perfect method would be a waste of time – like looking for a solution-in-radicals of general fifth other algebraic equation or for a ruler-and-compass angle trisection.

If no general perfect method is possible, then, instead of wasting time looking for such a method, we should look either for *classes* of functions and/or domains for which it is possible to compute the exact range, or for algorithms that still overestimate, but produce *better* estimates than the existing ones.

A (known) answer to the first methodological question. For interval computations, this important methodological question was answered in 1981, when Gaganov proved that the problem of computing the range is NP-hard (see, e.g., [1] and references therein).

Crudely speaking, NP-hard means that there are no general ways for solving this problem (i.e., computing the exact range) in reasonable time. (As an aside, it is possible to compute the range exactly in time that increases exponentially with n [1].) Of course, every NP-hard problem has easier-to-solve subclasses, and the problem of range estimation is no exception: as we have mentioned there are several important classes of functions for which we can compute the exact range in reasonable time. However, the NP-hardness result means that when we design a general range estimation algorithm, we can, in general, only compute *enclosures* for the desired range.

Maybe the difficulty from the requirement that the range be computed exactly? In practice, it is often sufficient to compute, in a reasonable amount of time, usefully accurate bounds for $\mathbf{y}$, i.e., bounds which are accurate within a given accuracy $\varepsilon > 0$. Alas, for any ε, such computations are also NP-hard.

Second methodological question. When we use an algorithm – e.g., straightforward interval computations – to estimate the range, we know that the result *may* be an overestimation. But is it?

As we have mentioned, there are many important sufficient conditions under which straightforward interval computations produce an exact range. New better sufficient conditions are being discovered. However, none of the known conditions is necessary: for each of these conditions, there are cases not covered by this condition in which the results are nevertheless exact.

Again, we have a natural question: are perfect (i.e., efficient, necessary and sufficient) conditions possible at all? If they are possible, then it is reasonable to spend some time looking for them. If such conditions are not possible, then looking for such perfect conditions would be a useless waste of time.

Our answer to this question. Let us consider algorithms $f(x_1, \ldots, x_n)$ that consist only of the operations $+$, $-$, $\cdot$, min, and max.

Theorem. *The problem of checking whether for a given algorithm $f(x_1, \ldots, x_n)$ and given intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$, straightforward interval computations are exact, is NP-hard.*

A similar result holds if we allow division as well.

In other words, no feasible necessary and sufficient conditions are possible for checking whether the estimate obtained by using straightforward computations is exact. As a result, instead of trying to find such conditions, we should fully concentrate on identifying classes of functions (or functions and box values) for which straightforward computations lead to the exact range. It is known that Gauss elimination and completing the square of a quadratic lead to exact range. Finding more cases like that is worth the effort.

Related open problems. In practice, it is usually sufficient to compute the range within a given accuracy ε. How difficult is it to check whether for a given algorithm $f(x_1, \ldots, x_n)$ and given intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$, straightforward interval computations are accurate within the given accuracy?

What if we consider other methods – such as centered form?

Acknowledgments. This work was supported in part by NASA grants NCC5-209 and NCC 2-1232, by NSF grants CDA-9522207, ERA-0112968 and 9710940 Mexico/Conacyt, and by AFOSR grant F49620-00-1-0365. The authors are thankful to Eldon Hansen, Weldon A. Lodwick, Bill Walster, and to the anonymous referees for fruitful discussions.

References.

(1) V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of ... interval computations*, Kluwer, Dordrecht, 1997.

Hardware and Software Support for Interval Arithmetic in the Past and in the Future

Ulrich W. Kulisch
Universität Karlsruhe
76128 Karlsruhe, Germany

Interval Mathematics has been developed to a high standard during the last few decades. It provides methods which deliver results with guarantees. However, the arithmetic available on existing processors makes these methods extremely slow. As a result they are not widely used in the scientific computing community as a whole. Three things are necessary to enhance the acceptance of interval mathematics: More sophisticated programming environments, better hardware support for interval arithmetic, and more and better education.

The talk will show that on super scalar processors interval operations can be made as fast as simple floating-point operations with only modest hardware costs.

The talk will briefly sketch the programming environments that have been developed at the author's institute since 1966. They are aiming to obtain highly accurate and guaranteed results. Then the talk will show that on super scalar processors interval operations can be made as fast as simple floating-point operations with only modest hardware costs.

For further details see the paper: "Advanced Arithmetic for the Digital Computer – Design of Arithmetic Units",
`ftp://ftp.iam.uni-karlsruhe.de/pub/documents/kulisch/tsukuba.ps.gz`

Interval Arithmetics over a Field $GF(p)$

L. V. Kupriyanova, D. V. Speransky, and V. G. Samoilov
Saratov State University, Russia
contact email KupriyanovaLV@info.sgu.ru

1 Introduction

Known methods of interval mathematics use, basically, arithmetic of intervals over a field of the real or complex numbers. In this work interval arithmetic over the finite fields $GF(p)$, where p is a prime number, is introduced, some properties of this arithmetic are proved and equations with interval coefficients are considered. Such arithmetic is necessary to develop methods of discrete systems theory. For example, it is possible to deduce the theory of linear sequential machines (LSM), which are defined over the finite fields $GF(p)$. LSM is a mathematical model of widely known actual discrete systems that carry out coding and decoding of information, signature analysis of output reactions of the device for its technical diagnosis, etc. From the physical point of view it is possible to measure levels of values of voltage for input and output values of signals, and also levels of state values of memory elements of electronic devices described by mathematical models LSM, in "quantum" units. As in specifications, the values of voltage have upper bounds, limiting value of this voltage (expressed in "quantum" units) just gives the characteristic p of a field $GF(p)$. From here there is a necessity to operate with quantum values of voltage in arithmetic modulo p. The levels of voltage are measured by devices with some error, and consequently instead of exact values of levels of signals there can be intervals, within the limits of which there are their valid values. To date, there has been no work on interval arithmetic over finite fields, and the present work can partly fill in this blank.

2 Notations and Basic Definitions

We denote the elements of $GF(p) = \{0, 1, \ldots, p-1\}$, where p is a prime number, by small Greek letters $\alpha, \beta, \ldots$, and also by Latin letters with feature from below or from above $\underline{a}, \overline{a}, \underline{b}, \overline{b}, \ldots$

Subset a of $GF(p)$ such, that $a = [\underline{a}, \overline{a}] = \{\alpha \mid \underline{a} \leq \alpha \leq \overline{a},\ \underline{a}, \overline{a} \in GF(p)\}$, we shall call as an *interior* closed interval, where $\underline{a}$ and $\overline{a}$ are its bottom and top bounds, respectively.

We shall interpret record of a form $b = [\underline{b}, \overline{b}]$, where $\underline{b} > \overline{b}$, as a set $GF(p) \setminus [\underline{b}+1, \overline{b}-1]$ and also to call this set as an *exterior* interval.

Interval of a form $[\underline{a}, \overline{a}]$, where $\underline{a} = \overline{a}$, we shall call a singular interval and to interpret it as an element of a field $GF(p)$.

Set of all intervals over $GF(p)$ we shall denote by $IGF(p)$, and the Latin letters we shall reserve behind notations of intervals.

Any exterior interval can be submitted as $[\underline{b}, \overline{b}] = [0, \overline{b}] \cup [\underline{b}, p-1]$.

Let's introduce operations over elements of $IGF(p)$.

Let $* \in \{+, -, \cdot, \div\}$ be a binary arithmetic operation. If $a, b \in IGF(p)$, then

$$a * b := \{\xi = \alpha * \beta \mid \alpha \in a, \beta \in b\}$$

defines binary arithmetic operation over elements of $IGF(p)$. In case of division it is prospective, that $0 \notin b$.

As against real interval arithmetics, result of operation over an interval of $IGF(p)$ can appear set of points not being one interval, and representing association of several intervals scattered on a numerical axis. For example, for $p = 7$, $[1,2] \cdot [2,3] = [2,4] \cup [6,6]$. We shall name subset $A \in GF(p)$ such, that

$$A = \bigcup_{i \in I} a_i,$$

where $a_i \in IGF(p)$, I is a finite set of indexes and for $i \neq j$ $a_i \cap a_j = \emptyset$, as the *generalized* interval of a field $GF(p)$. The generalized interval we shall designate by a capital letter. The usual interval $IGF(p)$ is a special case of the generalized interval. Behind set of all generalized intervals we shall keep a designation $IGF(p)$. Now let's introduce arithmetic operations over the generalized intervals.

Let $A = \bigcup\limits_{i \in I} a_i$, $B = \bigcup\limits_{j \in J} b_j$, where a_i, b_j is usual interval of a field $GF(p)$, then

$$A * B = (\bigcup_{i \in I} a_i) * (\bigcup_{j \in J} b_j) = \bigcup_{i \in I, j \in J} a_i * b_j.$$

Let's introduce unary operation over a usual interval $-x = [-\overline{x}, -\underline{x}]$, where $'-\xi'$ is an element of $GF(p)$, opposite to ξ on addition, then the appropriate operation over the generalized interval is

$$-X = (\bigcup_{i \in I} (-x_i).$$

Let's denote also $1/X = (\bigcup\limits_{\xi \in X} (1/\xi)$, where $'1/\xi'$ is an element of $GF(p)$, inverse to ξ on multiplication. Let's introduce operation of multiplication of an interval X on α, an element of a field $GF(p)$:

$$\alpha * X = \bigcup_{\xi \in X} [\alpha \cdot \xi, \alpha \cdot \xi],$$

As width of a usual interval $x = [\underline{x}, \overline{x}]$ we shall call a value

$$w(x) = \begin{cases} \overline{x} - \underline{x} + 1, & \text{if } x \text{ is interior,} \\ \overline{x} - \underline{x} + p + 1, & \text{if } x \text{ is exterior.} \end{cases}$$

Width of the generalized interval is $w(x) = \sum\limits_{i \in I} w(x_i)$.

3 Properties of Arithmetic Operations in $IGF(p)$

Let a, b be a usual intervals of a field $GF(p)$, A, B is generalized intervals, $\lambda, \mu \in GF(p)$, then the following formul as allowing to calculate or to estimate results of arithmetic operations over intervals with the help of operations with bounds of intervals are valid:

$$a + b = ft \begin{cases} [\underline{a} + \underline{b}, \overline{a} + \overline{b}], & \text{if } w(a) + w(b) \le p, \\ [0, p-1], & \text{otherwise.} \end{cases}$$

$$a - b = \begin{cases} [\underline{a} - \overline{b}, \overline{a} - \underline{b}], & \text{if } w(a) + w(b) \le p, \\ [0, p-1], & \text{otherwise.} \end{cases}$$

$$\lambda \cdot a \subseteq [\lambda \underline{a}, \lambda \overline{a}], \text{ if } \lambda(w(a) - 1) < p - 1,$$

$$a \cdot b \subseteq [\underline{a}\underline{b}, \overline{a}\overline{b}], \text{ if } \sigma(a)(w(b) - 1) + \sigma(b)(w(a) - 1) < 2(p - 1),$$

where $\sigma(a) = \underline{a} + \overline{a}$, $\sigma(b) = \underline{b} + \overline{b}$.

Besides known properties, having a place for real interval arithmetics (commutativity and associativity for adition and multiplication, absence opposite on addition and multiplication for the majority of elements, subdistributivity, etc.), the following properties are valid:

1. $\lambda(A + B) = \lambda A + \lambda B$ (distributivity of multiplication to number);
2. $(\lambda + \mu)A \subseteq \lambda A + \mu A$;
3. $w(A * B) \le w(A) \cdot w(B)$, where $* \in +, -, \cdot, /$;
4. $w(a + b) = w(a - b) = w(a) + w(b) - 1$, if a, b is usual intervals and $w(a) + w(b) < p$;
5. $w(\lambda A) = w(A)$, if $\lambda \neq 0$.

4 Solution Sets of the Equations in $IGF(p)$

As the algebraic solution of the equation relative to X

$$f(A, X) = B, \tag{1}$$

dependent from some parameters $(A_1, \dots, A_s)^T = A$, where $A_i, B \in IGF(p)$, $i = 1, \dots, s$, we shall call s uch generalized interval $X \in IGF(p)$, that at its substitution in the equation the correct equality turns out.

For the equation (1) we shall define also following solution sets.
$X_{\exists\exists} := \{\xi | (\exists\alpha \in a)(\exists\beta \in b) f(\alpha, \xi) = \beta\}$ is a united solution set;
$X_{\forall\exists} := \{\xi | (\forall\alpha \in a)(\exists\beta \in b) f(\alpha, \xi) = \beta\}$ is a tolerable solution set;
$X_{\exists\forall} := \{\xi | (\forall\beta \in b)(\exists\alpha \in a) f(\alpha, \xi) = \beta\}$ is a controlled solution set.

All solution sets and an algebraic solution, if they exist, are included in the united solution set. Also it is obvious the algebraic solution, that all its elements are the tolerable solutions, and therefore, the algebraic solution is included in tolerable solution set.

The linear equation $a + X = b$, where a, b are usual intervals over a field $GF(p)$, has the algebraic solution X if and only if $w(a) \leq w(b)$.

Validation Methods and Fuzzy Set Theory: Theory, Algorithms and Application

Weldon A. Lodwick[1] and K. David Jamison[2]

[1]Department of Mathematics, Campus Box 170
University of Colorado, P.O. Box 173364
Denver, CO 80217-3364
weldon.lodwick@cudenver.edu
[2]Watson Wyatt & Company, 950 17th Street, Suite 1400
Denver, CO 80202

Abstract

The idea of validation arises naturally within the context of interval analysis where the theory and methods are well developed. Approaching fuzzy set theory through fuzzy interval analysis, we develop a validation theory for fuzzy sets. An algorithm is given and applied to an optimization problem.

Keywords: Validation, Interval Analysis, Fuzzy Set Theory, Possibility Theory

1 Introduction

This talk is an extension and amplification of [4]. Our discussion is restricted to domains of real fuzzy numbers (see [1]). Validation in the context of fuzzy set theory is defined below as bounding a fuzzy set by upper and lower functions that enclose membership functions.

Definition 1. *Given a fuzzy number A with associated membership function $\mu_A(x)$, we say that A is* enclosed *if there exist bounding functions $p_A(x)$ and $n_A(x)$ such that $n_A(x) \leq \mu_A(x) \leq p_A(x), \forall x$ and for all valid membership functions, $\mu_A(x)$, of A. Given a fuzzy number A with any associated valid membership function $\mu_A(x)$, the fuzzy number A is (sequentially)* validated *if there exists a sequence of functions $p_k(x)$ and $n_k(x)$ that enclose A ($n_k(x) \leq \mu_A(x) \leq p_k(x)$ for all k) such that $p_k(x) \to \mu_A(x)$ and $n_k(x) \to \mu_A(x)$.*

The membership functions $n(x) = 0$ and $p(x) = 1$ are enclosures for any fuzzy number and hence validate every real fuzzy number. We seek the tightest bounding membership functions. Moreover, given an arbitrary measure μ, possibility and necessity measures can be defined that bound μ associated with a

given measurable set A (see [3]). That is,

$$N(A) \leq \mu(A) \leq \Pi(A).$$

Of interest are the distributions that arise from these measures. Their construction is straightforward (see [2, 3]). It is the construction of such bounding distributions that lead to validation.

2 Constructing Enclosures of Expressions

Fuzzy arithmetic and operations are used on the bounding functions when enclosing an expression whereas those associated with probability are, for example, convolutions for multiplication, so in general, more complex. Fuzzy operations on bounding functions result in measures and distributions that enclose and validate the expression. As a result the expected value of the possibility and necessity distributions bound the expected value of the underlying distribution.

Thus, enclosures can be created for expressions of random variables with known probability measure. On the other hand, when no underlying probabilities exist or are too costly to obtain and one may obtain a fuzzy membership function, this fuzzy membership function can be used to create bounding possibility and necessity distributions. The possibility (respectively necessity) function that one obtains in this case is the upper (respectively lower) bound of all probability distributions (see [2] equations (10.4)–(10.7)). Thus, in [2], there is the view of possibility and necessity of a fuzzy set (membership function) as encoding, between them, all possible probability distributions associated with the imprecise outcome represented by the given fuzzy set. This satisfies the definition of enclosure.

Given probability distributions or membership functions, one can bound the results. The process for constructing the possibility distribution from probabilities or memberships can be found in [3] where the necessity distribution is obtained from the constructed possibility measure which in turn is used to construct the necessity distribution. An example is given.

3 Convergence

Two types of convergence are considered. The first is when uncertainty is reduced to certainty; i.e., when the real fuzzy numbers converge to real numbers. The second is convergence of bounding functions to a given underlying distribution.

Theorem 1. (Uncertainty reduction convergence) *Given a sequence of random variables X_k with mean M_{X_k} and standard deviation σ_{X_k}, if $M_{X_k} \to x^*$ and $\sigma_{X_k} \to 0$, then X can be validated; i.e., $p_k(x) \to \mu^*_X(x)$ and $n_k(x) \to \mu^*_X(x)$, where $\mu^*_X(x) = \{1$ for $x \geq x^*$ and 0 otherwise$\}$, the cumulative distribution of a real number.*

The proof will be given.

Theorem 2. (Convergence of the enclosure to a given distribution) *Given an underlying distribution $\mu_X(x)$ there exists sequence of enclosures, $n_k(x)$ and $p_k(x)$, such that $p_k(x) \to \mu_X(x)$ and $n_k(x) \to \mu_X(x)$ $(n_k(x) \leq \mu_X(x) \leq p_k(x))$.*

This second approach to validation assumes that there is one underlying measure for each real fuzzy number or random variable in the expression. The idea is to partition the domain to obtain tighter bounds for the expression. We will illustrate the use of consistent possibility and necessity measures to obtain tight bounds on the estimated expected value via an example.

4 Application

We apply the ideas developed above to optimization under uncertainty.

5 Conclusions

We have extended the idea of validation to validation in the context of fuzzy set theory. Moreover, two approaches to fuzzy validation were developed and numerical examples presented.

References

[1] D. Dubois, H. T. Nguyen, and H. Prade, "Possibility theory, probability and fuzzy sets: Misunderstandings, bridges and gaps," In: D. Dubois and H. Prade (eds.), *The Handbook of Fuzzy Sets*, Kluwer, 2000, pp. 343–438.

[2] D. Dubois, E. Kerre, R. Mesiar, and H. Prade, "Fuzzy interval analysis," In: D. Dubois and H. Prade (eds.), *The Handbook of Fuzzy Sets*, Kluwer, 2000, pp. 483–581.

[3] K. D. Jamison and W. A. Lodwick, "The construction of consistent possibility and necessity measures," *Fuzzy Sets and Systems*, 2002 (to appear).

[4] W. A. Lodwick, K. D. Jamison, and F. D. Newman, "Extension of Interval Validation Methods to Fuzzy Set Theory," In: L. Hall (ed.), *Proceedings of NAFIPS'2001*, July 2001.

[5] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM Publications, Philadelphia, 1979.

[6] R. E. Moore, "Risk Analysis without Monte Carlo methods", *Freiburger Intervall-Berichte*, 1984, Vol. 1, pp. 1–48.

[7] L. A. Zadeh, "Fuzzy sets as a basis of a theory of possibility," *Fuzzy Sets and Systems*, 1978, Vol. 1, pp. 3–28.

The Early Days of Interval Global Optimization

Kaj Madsen[1] and Stig Skelboe[2]

[1]Informatics and Mathematical Modelling
Technical University of Denmark
[2]Computer Science, University of Copenhagen

The use of interval arithmetic for global optimization over an n-dimensional interval X was first described by Ramon Moore in a technical report from Stanford University in 1962. The knowledge of the technique became widespread in 1966 through the classical book of Moore, "Interval Analysis" [1].

Section 6.4, "Determination and use of extreme values of rational functions", has all the essentials needed for global optimization: reduction by monotonicity if possible, stationary point reduction by interval Newton if possible, and the use of a splitting strategy if none of the above is applicable.

The splitting strategy is described in Chapter 4 in [1]. It is expressed as a general formula for the computation of an arbitrarily good approximation of the united extension,

$$\bar{f}(X_1, X_2, \ldots, X_n) = \{f(x_1, x_2, \ldots, x_n) | x_i \in X_i,\ i = 1, 2, \ldots, n\}$$

Define

$$X_{i,j}^{(N)} = a_i + [j-1, j](b_i - a_i)/N,\ j = 1, 2, \ldots, N \quad \text{where } X_i = [a_i, b_i],\ a_i \leq b_i$$

Using the natural interval extension of f denoted $F(X_1, X_2, \ldots, X_n)$, Theorem 4.4 in [1] states,

$$F^{(N)}(X_1, X_2, \ldots, X_n) =$$

$$\bigcup_{j_1=1}^{N} \bigcup_{j_2=1}^{N} \cdots \bigcup_{j_n=1}^{N} F(X_{1,j_1}^{(N)}, X_{2,j_2}^{(N)}, \ldots, X_{n,j_n}^{(N)}) = \bar{f}(X_1, X_2, \ldots, X_n) + E_N(1)$$

where the width of E_N fulfils $w(E_N) \leq (K/N) \max_i(w(X_i))$ for some $K > 0$ and $0 \in E_N$.

The computational cost of (1) is N^n interval function evaluations. If the interval E_N is too wide, it may be reduced by a factor of 2 by computing (1) using $2N$ for N. The cost is now $(2N)^n = 2^n N^n$. A straightforward computation of (1) is therefore mainly of theoretical interest.

In his 1974 paper "Computation of rational interval functions" [2], Skelboe suggested an efficient algorithm for the evaluation of (1). Consider a sequence of computations, $F^{(1)}(X_1, X_2, \ldots, X_n)$, $F^{(2)}(X_1, X_2, \ldots, X_n)$, $F^{(4)}(X_1, X_2, \ldots, X_n)$, ... In the computing of $F^{(2N)}$ from $F^{(N)}$, the aim is only to compute those values $F(X_{1,j_1}^{(2N)}, X_{2,j_2}^{(2N)}, \ldots, X_{n,j_n}^{(2N)})$ that affect the interval end values of $F^{(2N)}$. It was also noted that $w(E_N) \leq (K/N^2)\max_i(w(X_i))$ when (1) is based on the centered form in stead of the natural interval extension.

The algorithm is closely related to a branch-and-bound algorithm and it is described in [2] both informally and as an Algol program. Today it is often referred to as the Moore-Skelboe algorithm.

The straightforward computation of (1) for $F^{(1)}$, $F^{(2)}$, $F^{(4)}$, ... can be structured as a tree, and for most functions it is obvious that the branch-and-bound algorithm only uses the same interval computations for the minimum and maximum interval values during the first few levels of computation of the tree. Therefore the algorithm in [2] first searched for the the minimum and then for the maximum. Some interval function computations are performed twice resulting in a minor waste. Originally the main purpose was to save space, but the strategy also leads to a simpler algorithm.

Moore found interest in this efficient algorithm, and we had an exciting visit of him in Copenhagen in 1975 where we discussed these matters. In his 1976 paper "On computing the range of a rational function of n variables over a bounded region" [3], Moore combined Skelboe's strategy with monotonicity tests, the centered form, and Krawzyk's version of Newton's method into a method for global optimization.

A key observation in [3] is that during the computation of the minimum (or maximum), just one interval function value $F(X_{1,j_1}^{(N)}, X_{2,j_2}^{(N)}, \ldots, X_{n,j_n}^{(N)})$ defines the minimum (or maximum). Therefore it is of interest to compute this interval value as accurately as possible, and this can be done using the appropriate method or combination of methods among the above-mentioned.

The total computational cost in interval function evaluations for (1) using the centered form with a resulting accuracy of $w(E_N) \approx \varepsilon$ is given in [3] as: $(K_1/\varepsilon)^{n/2}$. The branch-and-bound algorithm in [2] requires $K_2 2^n \log_2(K_1/\varepsilon)$ interval function evaluations assuming only a finite number of isolated extrema.

The phrase "Global Optimization" was not used in the early papers. The first time we have seen it is in Eldon Hansen's paper "Global optimization using interval analysis: The one-dimensional case" [4] and in the paper "True worst-case analysis of linear electrical circuits by interval arithmetic" [5] by Skelboe, both published in 1979.

In [5] the algorithm from [2] – augmented with check for monotonicity – is used for the solution of a more realistic problem, namely worst-case analysis of the frequency response of two simple filters with 3 and 4 interval variables, respectively. The interval extension is computed using the mean-value form. This gives the same convergence properties as the centered form, but the mean-

value form has two advantages over the centered form: it is straightforward to derive and with the partial derivatives readily available, monotone intervals are identified for free.

In the talk we describe these early attempts, seen from our side. Highlights were visits of Ramon Moore as well as Eldon Hansen, and some years later – in 1980 – a full year's visit of Louis Rall. They all gave us new insight and inspiration through our many discussions and their excellent lectures.

References

[1] R. E. Moore, *Interval Analysis*, Prentice-Hall, 1966.

[2] S. Skelboe, "Computation of rational interval functions", *BIT*, 1974, Vol. 14, pp. 87–95.

[3] R. E. Moore, "On computing the range of a rational function of n variables over a bounded region", *Computing*, 1976, Vol. 15, pp. 1–15.

[4] E. R. Hansen, "Global optimization using interval analysis: The one-dimensional case", *Journal of Optimization Theory and Applications*, 1979, Vol. 29, pp. 331–344.

[5] S. Skelboe, "True worst-case analysis of linear electrical circuits by interval arithmetic", *IEEE Trans. Circuits and Systems*, 1979, Vol. CAS-26, pp. 874–879.

Taylor Model Based Verified Integration for the Volterra Equations and the Lorenz System

Kyoko Makino[1] and Martin Berz[2]

[1]Department of Physics,
University of Illinois at Urbana-Champaign,
Urbane, IL 61801-3080, USA, makino@uiuc.edu
[2]Department of Physics and Astronomy, and
National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing, MI 48824, USA, berz@msu.edu

One of the important fields in validated methods is solving initial value problems in ordinary differential equations (ODEs). The typical problem of overestimation in interval arithmetic is mostly caused by the lack of information of functional dependency. The dependency problem that is related to cancellation is one issue to be overcome. Besides, in the case of solving a system of multidimensional ODEs, there arises the so-called wrapping effect, which is caused by the inflation of the size of the geometric set enclosing the validated solution set at each time step. The wrapping effect is a particular form of the dependency problem based on the connection of current dynamical values on initial conditions, which often is more dramatic than the other sources of overestimation. The history of development of new schemes for verified integration of ODEs illustrates the struggle with those two challenging questions [13, 6, 14, 7, 15].

The Taylor model method [10, 9] combines interval method for validation and high order automatic differentiation for local functional behavior. The method models a function f in the domain $\vec{D}$ by a high order multivariate Taylor polynomial P and the remainder error interval I :

$$\forall \vec{x} \in \vec{D}, \quad f(\vec{x}) \in P(\vec{x} - \vec{x}_0) + I, \tag{1}$$

where $\vec{x}_0$ is the reference point of the Taylor expansion. The nth order Taylor polynomial P is expressed with floating point number coefficients, and it captures the bulk of functional dependency, hence the major source of interval overestimation is removed. The Taylor remainder and any numerical errors arisen in the domain $\vec{D}$ are kept in an interval, namely the remainder error interval I, and the size of I is proportional to $|\vec{D} - \vec{x}_0|^{n+1}$, which can be very small in practice by choosing the size of $\vec{D}$ sufficiently small. The standard binary

operations and intrinsic functions on Taylor models were implemented in the code COSY Infinity [9, 2]. It is of particular significance that an antiderivation operation ∂^{-1} is treated as an intrinsic function in the Taylor model structure [9], and this formally removes the difference between the solution of ODEs and merely algebraic equations based on fixed point methods.

We applied the Taylor models to verified integrations of ODEs,

$$\frac{d\vec{x}\,(t;\vec{x}_{ini})}{dt} = \vec{f}\,(\vec{x}\,(t;\vec{x}_{ini})\,,t) \quad \text{with} \quad \vec{x}(t_{ini};\vec{x}_{ini}) = \vec{x}_{ini},$$

and the basic algorithm is discussed in [3, 9]. The Taylor approach is applied to expand not only in the independent variable t, but also in the initial value $\vec{x}_{ini}$, which is possible with our Taylor model implementation with high order multivariate Taylor polynomials, and several advantages have been observed.

- The direct availability of the antiderivation on Taylor models allows us to treat the Picard operator like any other function, avoiding the need to explicitly bound error terms of integration formulas.
- The inclusion requirement asserting existence of a solution reduces to a mere inclusion of the remainder intervals.
- The explicit dependency on initial variables can be carried through the whole integration process. This controls the dependency problem optimally, and, most importantly, there is no need to re-package the momentary solution set at each time step, and hence there is no wrapping effect. Thus, it allows for a much larger domain of initial condition and longer integration times.

We have shown how the Taylor models control the dependency problems efficiently for non "single use expression" (SUI) problems [11, 12], and the same efficiency applies to complicated ODEs like the near earth asteroid problem [4, 5]. When the ODEs have SUI expressions, i.e. the right hand side of the equations do not have a source of overestimation of arithmetic nature, the overestimation mostly comes from the pure wrapping effect. Such ODEs are suitable to study the difficulties unique to ODE initial value problems. In this paper, we study SUI ODEs, and by doing that, we want to show the essence of why and how the Taylor model based verified integrator is successful for the near earth asteroid problem.

We use the Volterra equations

$$\frac{dx_1}{dt} = 2x_1(1-x_2), \quad \frac{dx_2}{dt} = -x_2(1-x_1)$$

to illustrate how the method works. The Volterra equations have been historically used as a test case of validated initial value problems [1, 14]. For the purpose of illustration, we take a large interval box for the initial condition

$$x_{1ini} \in 1 + [-0.05, 0.05], \quad x_{2ini} \in 3 + [-0.05, 0.05].$$

We used our Taylor model based integrator VI coded in COSY Infinity [2] and AWA by Lohner [8, 7] to study the performance. AWA represents the conventional methods, since it is one of the most successful codes based on conventional methods and it is widely spread. Despite of the large size of the initial condition, the total error is easily kept around 10^{-10} for the whole one cycle with COSY-VI, while AWA cannot complete the cycle. Both codes take about the same CPU time. The extensive study on the problem addresses why the conventional approach [13, 6, 14, 7, 15] could not handle the problem.

The Lorenz system is another good example to illustrate how the Taylor model based verified integrator works.

$$\frac{dx_1}{dt} = 10(x_2 - x_1), \quad \frac{dx_2}{dt} = x_1(28 - x_3) - x_2, \quad \frac{dx_3}{dt} = x_1 x_2 - \frac{8}{3}x_3.$$

Similar to the Volterra equations, the right hand side is SUI. Since the system exhibits a chaotic motion, it is particularly challenging to validating methods. Even for a large initial condition box

$$x_{1ini} \in 15 + [-0.01, 0.01], x_{2ini} \in 15 + [-0.01, 0.01], x_{3ini} \in 36 + [-0.01, 0.01],$$

the Taylor model approach can integrate beyond the time 5 easily, while AWA breaks down around the time 1.5, indicating that the Taylor model method can be used for validation of various ODE initial value problems for a larger domain and longer times.

References

[1] W. F. Ames and E. Adams, "Monotonically convergent numerical two-sided bounds for a differential birth and death process", In: K. Nickel (ed.), *Interval Mathematics*, Vol. 29 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin–New York, 1975, pp. 135–140,

[2] M. Berz and J. Hoefkens. *COSY INFINITY Version 8.1 – programming manual*, Technical Report MSUCL-1196, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 2001; see also http://cosy.pa.msu.edu.

[3] M. Berz and K. Makino, "Verified integration of ODEs and flows with differential algebraic methods on Taylor models", *Reliable Computing*, 1998, Vol. 4, pp. 361–369.

[4] M. Berz, K. Makino, and J. Hoefkens, "Verified integration of dynamics in the solar system", *Nonlinear Analysis*, 2000.

[5] J. Hoefkens, *Rigorous Numerical Analysis with High-Order Taylor Models*, Ph.D. thesis, Michigan State University, East Lansing, Michigan, USA, 2001.

[6] F. Krückeberg, "Ordinary differential equations", In E. Hansen (ed.), *Topics in Interval Analysis*, Clarendon Press, Oxford, 1969, pp. 91–97.

[7] R. Lohner, *Einschliessung der Loesung gewoehnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Ph.D. thesis, Universitaet Karlsruhe, Karlsruhe, Germany, 1988.

[8] R. Lohner, *AWA, the progam AWA (AnfangsWertAufgabe)*; see also http://www.uni-karlsruhe.de/ Rudolf.Lohner/english/index.html.

[9] K. Makino, *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*, PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998; also available as Technical Report MSUCL-1093, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824.

[10] K. Makino and M. Berz, "Remainder differential algebras and their applications", In: M. Berz, C. Bischof, G. Corliss, and A. Griewank (eds.), *Computational Differentiation: Techniques, Applications, and Tools*, SIAM, Philadelphia, 1996, pp. 63–74.

[11] K. Makino and M. Berz, "Efficient control of the dependency problem based on Taylor model methods", *Reliable Computing*, 1999, Vol. 5, pp. 3–12.

[12] K. Makino and M. Berz, "Higher order verified inclusions of multidimensional systems by Taylor models" *Nonlinear Analysis*, 2000.

[13] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1966.

[14] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, 1979.

[15] N. Nedialkov, *Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation*, Ph.D. thesis, University of Toronto, Toronto, Canada, 1999.

A Numerical Study on a New Heuristical Decision Index for Interval Global Optimization

Mihály Csaba Markót[1], Tibor Csendes[1],
José Fernández Hernández[2], and Leocadio González Casado[3]

[1]Institute of Informatics
University of Szeged, Hungary
[2]Dpt. Statistics and Operations Research
University of Murcia, Spain
[3]Dpt. Computer Architecture and Electronics
University of Almería, Spain

The talk gives an overview on the numerical test results of solving inequality constrained global optimization test problems with interval Branch-and-Bound methods.

In [1, 3] a new heuristic decision index was discussed for *unconstrained* problems and investigated in detail. This index has the form of $p\hat{f}(\boldsymbol{X}) := (\hat{f} - \underline{f}(\boldsymbol{X}))/w(\boldsymbol{f}(\boldsymbol{X}))$, where $\boldsymbol{X}$ is an interval vector, $\hat{f}$ is an approximation of the global minimum value and $\boldsymbol{f}$ denotes the interval inclusion function of the objective function. This index measures the relative position of the minimum within the inclusion $\boldsymbol{f}(\boldsymbol{X})$ and it is suitable to be applied as a subinterval selection criterion and as a part of the subdivision rule as a decision factor.

J. F. Hernández proposed the idea of extending this index for constrained problems by taking the effect of the constraints into account in a similar way:

$$pu_{\boldsymbol{g}_j}(\boldsymbol{X}) := \min\left\{\frac{-\underline{\boldsymbol{g}_j}(\boldsymbol{X})}{w(\boldsymbol{g}_j(\boldsymbol{X}))}, 1\right\}, \quad pu(\boldsymbol{X}) := \prod_{j=1}^{r} pu_{\boldsymbol{g}_j}(\boldsymbol{X}).$$

(where $\boldsymbol{g}_j$ is the interval inclusion function of the jth constraint). The *pu* quantity measures the relative position of 0 within the inclusions of the constraint functions, i.e. the feasibility of the box $\boldsymbol{X}$. Finally, the heuristical decision index for constrained problems is formalized by $pup(\hat{f}, \boldsymbol{X}) := pu(\boldsymbol{X}) \cdot p(\hat{f}, \boldsymbol{X})$. We can conclude that if the *pup* value for a given box is high, then the box should be preferred for an early selection (interval selection step), or it is advisable to split it into a higher number of subboxes (subdivision step).

In the numerical tests we were dealing with two different types of problems: the first was the problem class of the obnoxious facility location model [4].

For such a problem our goal is to place an unpleasing object into a region by considering the disappointment of the inhabitants (described by an exponential objective to be minimized) and the geographical possibilities (modelled by linear and quadratic constraint functions).

The second part of the test problems came basically from unconstrained global optimization; we have selected some harder problems, e.g. Hartman-6, Goldstein-Price, Levy-3, Ratz-4 and EX2 (for the definitions see [2, 6, 7]) and completed them with sets of randomly generated lineár and quadratic constraints.

The main consequence of our investigations is that the new type of interval selection criterion significantly improves the efficiency in terms of both the running time and the memory complexity. In addition, the largest improvements were achieved on the hardest problem instances. We found that it is worth to make further investigations of our methods on other type of hard constrained problems.

Acknowledgements

This work was supported by the Grants FKFP 0449/99, OTKA T 32118 and T 34350.

References

[1] L. G. Casado, J. A. Martínez, and I. García, "Experimenting with a new selection criterion in a fast interval optimization algorithm", *J. Global Optimization*, 2001, Vol. 19, pp. 247–264.

[2] T. Csendes and D. Ratz, "Subdivision direction selection in interval methods for global optimization", *SIAM J. Num. Anal.*, 1997, Vol. 34, pp 922–938.

[3] T. Csendes, "New subinterval selection criteria for interval global optimization", *J. Global Optimization*, 2001, Vol. 19, pp. 307–327.

[4] J. Fernández, P. Fernández, and B. Pelegrín, "A continuous location model for siting a non-noxious undesirable facility within a geographical region", *European Journal of Operational Research*, 2000, Vol. 121, pp. 259–274.

[5] M. Cs. Markót, J. Fernández, L. G. Casado, and T. Csendes, *New interval methods for constrained global optimization*, In preparation. Available at `http://www.inf.u-szeged.hu/~markot/`.

[6] D. Ratz and T. Csendes, "On the Selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization", *J. Global Optimization*, 1995, Vol. 7, pp. 183–207.

[7] A. Törn and A. Žilinkas, *Global Optimization*, Springer-Verlag, Berlin, 1989.

Solving Electrical Power Load Flow Problems Using Intervals

Eustaquio A. Martínez[1], Benjamín Barán[2],
and Tiaraju Asmuz Diverio[3]

[1]Facultad Politécnica, Universidad Nacional del Este,
Ciudad del Este, Paraguay, amartinez@politec.une.edu.py
[2]Centro Nacional de Computación
Universidad Nacional de Asunción
San Lorenzo, Paraguay, bbaran@cnc.una.py
[3]Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brasil, diverio@inf.ufrgs.br

Due to its importance to the planning and maintenance of large electrical power distribution systems, the electrical power load flow problem has been exhaustively studied by electrical engineers. Taken in this context, the idea of an interval approach is particularly interesting, if one considers the possibility of finding all the solutions within a domain. This advantage is specially important given that non-linearity of the problem give rise to at least one operating point (feasible solution) for which traditional controls drive the system to a collapse, producing an undesirable blackout as the ones produced in Brazil and the United States of America. It also allows the representation of the problem's parameters as intervals, based on the physical values and their respective tolerance intervals, aiming to the application of the method in the sensitivity analysis of an electrical power system.

This work first presents the use of interval arithmetic to solve the electrical power flow problem in a sequential environment. However, the sequential approach requires expensive computational resources (as supercomputers) not always available to solve the problem in workable time. Consequently, a new parallel asynchronous approach running on a local area network of personal computers is presented to speed up the process with resources already available in most organizatións. This novel approach is based on the interval Newton/Generalized Bisection algorithm, that is parallelize using an asynchronous communication techniques that let each processor of an heterogeneous network of computers to work at its own speed, sharing results and workload with other processors of the network, to accomplish the calculation goal in workable time.

This way, the method may be scaled with a number of available processors, to solve problems of greater magnitude.

The electrical power load flow problem can be formulated as a quasi-linear system of equations

$$Yx = I(x) \tag{1}$$

where Y is the admittance matrix, $Y = \{y_{ki}\} \in C^{nxn}$, with $y_{ki} = G_{ki} + B_{ki} \in C$; $x \in C^n$ represents the (usually unknown) voltage vector (therefore, n is the problem size), and $I(x)$ is the current vector, $I \in C^n$. To facilitate control of operational restrictions (usually on the voltage magnitude), the problem is mostly solved in polar coordinates as:

$$P_k = V_k \sum_{i \in K} V_i (G_{ki} \cos \Theta_{ki} - B_{ki} \sin \Theta_{ki}) \tag{2}$$

$$Q_k = V_k \sum_{i \in K} V_i (G_{ki} \cos \Theta_{ki} - B_{ki} \sin \Theta_{ki}) \tag{3}$$

where $\Theta_{ki} = \Theta_k - \Theta_i \forall k \in \{1, \ldots, n\}$, K is the group of the bus bars adjacent to k and k itself.

The use of an interval approach for the solution of a non-linear system as the one above, brings along some interesting advantages, such as high accuracy and self-validation, as well as proof of root existence and uniqueness of solutions. The interval approach allows us to find the solutions by estimating an interval (or union of intervals) which is expected to contain one or more solution. The method then will indicate if such solutions exist or not. Observe that point methods used at present, such as the *Newton-Raphson*, do not posses these important features.

In this context, a non-linear system can be written:

$$F(X) = (f_1(X), \ldots, f_2(X))^T = 0 \tag{4}$$

where $F : R^n \to R^n$, $X = (x_1, x_2, \ldots, x_n)^T \in R^n$ and $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $1 \leq i \leq n$, $\underline{x}_i$ y $\overline{x}_i$ are the lower and upper bounds of x_i . The interval Newton method for non-linear equation systems has quadratic convergence can be used to solve the problem. The system (4) can be written as a linear interval system:

$$\mathbf{F}'(\mathbf{X}^k)(\tilde{\mathbf{X}}^k - X^k) = -F(X^k) \tag{5}$$

where $\mathbf{X}^k \in IR^n$ is the interval vector where the solution $X^\star \in R^n$ is expected to be found; $X^k \in R^n$ is an inner vector of $\mathbf{X}^k$, i.e. $X^k \in \mathbf{X}^k$ (usually the midpoint of $\mathbf{X}^k$) is the unknown interval vector which is expected to contain the solution $X^\star$; $\mathbf{F}'(\mathbf{X}^k) \in IR^{nxn}$ is the interval extension of the Jacobian matrix of $Fin\mathbf{X}^k$. $\tilde{\mathbf{X}}^k$ can be calculated by solving equation (5). The iterative formula for a system with n variables results in:

$$\mathbf{X}^{k+1} = \mathbf{X}^k \cap \tilde{\mathbf{X}}^k \tag{6}$$

If $\mathbf{X}^{k+1} = \phi$ (empty interval) then the non-existence of a solution in $\mathbf{X}^k$ is proved. To compute $\tilde{\mathbf{X}}^k$ solving (5), any known method, such as Gauss elimination Method or Gauss-Seidel interval Method can be used. In this work it is used the latter.

In order to solve (5) using the interval Newton Method, the applicable interval system may be written as:

$$\begin{bmatrix} \mathbf{H}^k & \mathbf{N}^k \\ \mathbf{J}^k & \mathbf{L}^k \end{bmatrix} \left(\begin{bmatrix} \tilde{\Theta}^k \\ \tilde{\mathbf{V}}^k \end{bmatrix} - \begin{bmatrix} \Theta^k \\ V^k \end{bmatrix} \right) = \begin{bmatrix} \Delta\mathbf{P}^k \\ \Delta\mathbf{Q}^k \end{bmatrix} \tag{7}$$

where $\begin{bmatrix} \mathbf{H}^k & \mathbf{N}^k \\ \mathbf{J}^k & \mathbf{L}^k \end{bmatrix} = \mathbf{F}'(\mathbf{X}^k); \begin{bmatrix} \tilde{\Theta}^k \\ \tilde{\mathbf{V}}^k \end{bmatrix} = \tilde{\mathbf{X}}^k; \begin{bmatrix} \Delta\mathbf{P}^k \\ \Delta\mathbf{Q}^k \end{bmatrix} = F(X^k)$, and $\mathbf{H}$, $\mathbf{N}$, $\mathbf{J}$, $\mathbf{L}$ are the interval sub- matrices that depend of the problem, $\mathbf{V}$ and Θ are intervals vectors.

It is well established in the field that the search region for the load flow problem is:

$$\Theta = [-\Theta_{max}, \Theta_{max}] \tag{8a}$$

$$\mathbf{V} = [-\zeta + 1, \zeta + 1] \tag{8b}$$

where $\Theta_{max} \cong 10°$ and $\zeta < 1$, according to heuristic recommendation.

System of equations (7) is first solved sequentially by using Interval Newton/Generalized Bisection. However, electrical systems are non-linear system of large dimensions and requires computational resources not always available. This fact motivated our studies of parallel techniques and algorithms in an asynchronous environment of personal computers to reduce processing times and to optimize the use of available computer networks. To solve the problem in a parallel asynchronous environment as a network of computers, the original problem should be partitioned in several smaller sub-problems, in such a way that each processor of the network, can work on its own sub-problems without much intervention of other processors.

A simple approach to partition the low flow problem in sub-problems when using the proposed interval method is by dividing the search domain in disjoint sub-domains. That way, each processor makes calculations on its assigned sub-domain without interfering with other processors. Of course, a master process is needed to manage the work of each processor. In this way, each processor carries out its search in a particular region, which is smaller that the global domain. At each processor, the algorithm detects whether a solution exists (or not) within its sub-domain and it communicates its finding to the master, which may assign a new sub-domain for each new available processor, managing load balancing, until the problem is completely solved. Of course, hard sub- problems may be further sub-divided using Generalized Bisection, reducing the total processing time.

In order to determine the advantages obtained by the proposed parallel method, a Sp (Speed-Up) measure of acceleration is defined as the relation between the sequential processing time and the parallel processing time.

In order to verify the proposition's validity, algorithms (both traditional point method and the proposed interval approach) were implemented in C language and several well known test problems were solved, as the IEEE 5 and 14 busbur paradigm, the Monticelli 30-busbar system and a 88-busbar electrical system. The authors decided to use an existing asynchronous communication facilities, already built in MPI (Message Passing Interface) software without any existing interval software, considering the difficulty of implementing a reliable parallel asynchronism in the latter. At the moment is in study the implementation of algorithms in c-xsc for Linux.

The computation environment was based on a 10 Mbps local area network of 5 personal computers with Pentium II processors of 400MHz and 32 Mb RAM, running a Linux Red Hat operating system. In the parallel implementation, one acts as the master, NFS and NIS server and has MPI installed. The others four work as slaves.

Experimental results show that the parallel approach is not only faster, but it also founds better results with smaller solution diameter and power mismatch (difference between actual and calculated power), thus offering an additional advantage.

In summary, a parallel asynchronous interval approach to the load flow problem seems beneficial in time reduction and quality of solutions and a conjecture of scalability of this advantages to larger problems with an even larger number of inexpensive computers in a local area network may be stated.

On the Shape of the Limit of the Total Step Method in Interval Analysis

Günter Mayer and Ingo Warnke
Fachbereich Mathematik, Universität Rostock
D–18051 Rostock, Germany
emails guenter.mayer@mathematik.uni-rostock.de and
ingo@sun4.math.uni-rostock.de

Given an $n \times n$ interval matrix [C] and an interval vector $[b]$ with n components the solution set S of the interval linear system $[C]x = [b]$ is defined by

$$S := \{\, x \in \mathbb{R}^n \mid Cx = b,\ C \in [C],\ b \in [b] \,\} .$$

Since this set normally cannot easily be described (cf. [7], e.g.) one looks for enclosures of S by an interval vector. One of the simplest iterative methods to obtain such an enclosure is based on the Richardson splitting $[C] = I - [A]$, i.e., $[A] := I - [C] = ([\underline{a}_{ij}, \overline{a}_{ij}])$, and reads

$$[x]^{(k+1)} = [A][x]^{(k)} + [b], \quad k = 0, 1, \ldots . \tag{1}$$

It can be found in [1] and can be regarded as the starting point of many other iterative algorithms for enclosing S, among them such well–known iterations like

$$[x]^{(k+1)} = (I - R[C])[x]^{(k)} + R[b], \quad k = 0, 1, \ldots , \tag{2}$$

and

$$[x]_\Delta^{(k+1)} = (I - R[C])[x]_\Delta^{(k)} + R([b] - [C]\tilde{x}), \quad k = 0, 1, \ldots , \tag{3}$$

(cf. [6], [8], e.g.) where $R \in \mathbb{R}^{n\times n}$ denotes any nonsingular preconditioning matrix, $\tilde{x}$ is an approximation of an arbitrary element of S and $[x]_\Delta^{(k)}$ is used to approximate and/or to enclose the errors $x - \tilde{x}$ for $x \in S$. Replacing $[A]$ in (1) by $I - R[C]$, $[b]$ by $R[b]$, and $R([b] - [C]\tilde{x})$, respectively, ends up with (2) and (3), respectively. Thus these latter iterations are particular cases of (1).

Assuming infinite precision the method (1) was extensively studied in the late sixties and the early seventies where the question of convergence was completely answered by a well–known theorem of Otto Mayer [5]. It turned out

that $\rho(|[A]|) < 1$ is necessary and sufficient for the convergence of (1) where $|[A]| := (\max\{\,|\underline{a}_{ij}|, |\overline{a}_{ij}|\,\}) \in \mathbb{R}^{n\times n}$ denotes the absolute value of $[A]$ and $\rho(|[A]|) := \max\{\,|\lambda| \mid \lambda \text{ eigenvalue of } |[A]|\,\}$ is the spectral radius of the real matrix $|[A]|$. It is easily seen that in this case the limit $[x]^*$ of (1) satisfies $S \subseteq [x]^*$. Moreover, $S \subseteq [x]^* \subseteq [x]^{(k+1)} \subseteq x^{(k)}$, $k = 0, 1, \ldots$ can be shown if $[x]^{(1)} \subseteq [x]^{(0)}$ is true. In the case $\rho(|[A]|) < 1$ this holds, e.g., if one starts with $[x]^{(0)} = (I - |[A]|)^{-1}|[b]|[-1, 1]$ or with $[x]^{(0)} = (1 - \| \, |[A]| \, \|_\infty)^{-1}|[b]|[-1, 1]$ where the absolute value $|[b]|$ of $[b]$ is defined analogously to $|[A]|$ and where $\|\cdot\|_\infty$ denotes the row sum norm.

Two questions on (1) remained open up to now: How does the limit $[x]^*$ look like and how close with respect to the well–known Hausdorff distance does it approach the interval hull $\square(S)$ of S. If, for instance, $[A] \equiv A \in \mathbb{R}^{n\times n}$, $[b] \equiv b \in \mathbb{R}^n$ one gets $[x]^* \equiv x^* = (I - A)^{-1}b$, hence $[x]^*$ is known and equals $\square(S) = S$. If, however, $[A] = \begin{pmatrix} 0 & [-1,1] \\ [-\frac{1}{2}, \frac{1}{2}] & 0 \end{pmatrix}$, $[b] \equiv b = (-1, 1)^T$ the limit $[x]^* = ([-4, 2], [-1, 3])^T$ overestimates the interval hull $\square(S) = ([-4, 0], [\frac{1}{3}, 3])^T$. Although we could not answer our two questions completely we gained some insight into the structure of $[x]^*$. We represented $[x]^*$ by means of the solution of two coupled systems of linear equations; cf. [3]. Knowing a simple element of $[x]^*$ (which one gets, e.g., when solving a real linear system $x = Ax + b$ with fixed $A \in [A]$, $b \in [b]$) one often is able to determine one half of the – a priori unknown – matrix entries of these two systems. For particular classes of matrices $[A]$ and vectors $[b]$ we even were able to represent $[x]^*$ precisely by means of midpoint and radius of $[A]$ and $[b]$; cf. [2].

As a limit case of (1) with the convergence condition $\rho(|[A]|) < 1$ we consider algebraic solutions $[x]^*$ of the equation

$$[x] = [A][x] + [b]$$

with $\rho(|[A]|) = 1$ and – as a generalization thereof - with $\rho(|[A]|) > 1$. Assuming $|[A]|$ to be irreducible (cf. [9], e.g.) both cases could completely be handled in [4] concerning existence, uniqueness and shape of $[x]^*$. In the first case 'the' Perron vector of $|[A]|$, i.e. the (up to a positive factor) unique positive eigenvector of $|[A]|$ associated with the eigenvalue $\rho(|[A]|)$ plays a crucial role while in the second case $[x]^*$ necessarily is degenerate. Maybe that in the case $\rho(|[A]|) < 1$ the final structure of $[x]^*$ can be derived by means of some ideas used in the case $\rho(|[A]|) = 1$.

Matrices $[A]$ with $\rho(|[A]|) = 1$ and reducible absolute value $|[A]|$ normally involve the case $\rho(|[\tilde{A}]|) < 1$ by some of their diagonal blocs $[\tilde{A}]$ when looking at their reducible normal form as defined in [9]. If the shape of $[x]^*$ is known for $\rho(|[\tilde{A}]|) < 1$, it can be shown that due to our results this shape is also known in the case $\rho(|[A]|) \geq 1$.

References

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, New York, 1983.

[2] G. Mayer and I. Warnke, "On the Shape of the Fixed Points of $[f]([x]) = [A][x] + [b]$", In: G. Alefeld, J. Rohn, S. M. Rump, and T. Yamamoto (eds.), *Symbolic Algebraic Methods and Verification Methods – Theory and Applications*, Springer, Wien, 2001, pp. 153 – 162.

[3] G. Mayer and I. Warnke, "On the Limit of the Total Step Method in Interval Analysis", In: U. Kulisch, R. Lohner, and A. Facius (eds.), *Perspectives on Enclosure Methods*, Springer, Wien, 2001, pp. 157–172.

[4] G. Mayer and I. Warnke, "On the Fixed Points of the Interval Function $[f]([x]) = [A][x] + [b]$", *Linear Algebra Appl.* (to appear).

[5] O. Mayer, *Über die in der Intervallrechnung auftretenden Räume und einige Anwendungen*, Ph.D. Thesis, Universität Karlsruhe, Karlsruhe, 1968.

[6] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[7] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.

[8] S. M. Rump, "Solving Algebraic Problems with High Accuracy", In: U. W. Kulisch and W. L. Miranker (eds.), *A New Approach to Scientific Computation*, Academic Press, New York, 1983, pp. 51–120.

[9] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1962.

Twin Estimates for Slopes

Humberto Muñoz and R. B. Kearfott
Department of Mathematics
University of Louisiana at Lafayette
Box 4-1010, Lafayette, LA 70504-1010 USA
hnm2294@louisiana.edu
rbk@louisiana.edu

1 Introduction

Interval slopes are useful in the rigorous treatment of non-smooth optimization problems, as we have outlined in [3, Ch. 6]. There, we developed formulas for outer estimates for the ranges of slopes of non-smooth and discontinuous functions. Slopes of non-smooth functions provide a simple alternative, computable with automatic differentiation procedures, to concepts such as the generalized gradient [1], and semigradient [5].

For various reasons, it may also be useful to compute inner estimates to the range of slopes for non-smooth or for discontinuous functions. For example, we can develop theory relating slopes to generalized gradients; depending on that theory, inner slope estimates would then be guaranteed to be elements of the generalized gradient. We could then develop general, automatic algorithms based on previous algorithms that utilized generalized gradients.

An alternate reason for developing inner estimates is to obtain bounds on the overestimation in the outer slopes.

Formulas for inner estimates for slopes are somewhat trickier than formulas for outer estimates. In [6], such formulas for inner slopes for various elementary functions, such as max and $|\cdot|$ are presented. The development there is analogous to that of [3, Ch. 6].

Incorporation of the formulas for inner slopes into expressions for objective functions, etc. requires an arithmetic based on inner estimations, rather than standard interval arithmetic. We have used *twin arithmetic* as Kreinovich and Nesterov [4, 7] have proposed. This arithmetic is operationally equivalent to Kaucher arithmetic (ibid.).

2 A Few Details

We term our procedure automatic twin slope computation (ATSC). Inner and outer bounds of the actual slope set are given simultaneously for nonsmooth

functions such as $|f(x)|$, $\max\{f(x), g(x)\}$, $f, g : \mathbb{R}^n \to \mathbb{R}$, and expressions defined by `if-then-else` branches.

Definition 1 *(Twin arithmetic [4] and [7]). A twin is a pair of intervals* $\boldsymbol{t} = (\boldsymbol{x}_{inn}, \boldsymbol{x})$*, with associated relations* $\subseteq$ *and* $\sqsubseteq$*, such that* $\boldsymbol{x}_{inn} \in \mathbb{IR} \cup \{\emptyset\}$, $\boldsymbol{x} \in \mathbb{IR}$*, and for* $\boldsymbol{y} \in \mathbb{IR}$*,* $\boldsymbol{y} \sqsubseteq \boldsymbol{t}$ *denotes* $\boldsymbol{x}_{inn} \subseteq \boldsymbol{y} \subseteq \boldsymbol{x}$*.* $(\boldsymbol{x}, \boldsymbol{x})$ *is a degenerate twin, and* $\boldsymbol{y} \sqsubseteq (\emptyset, \boldsymbol{x})$ *means that there is only an outer estimation of* $\boldsymbol{y}$*, which is* $\boldsymbol{x}$*.*

Basically, a twin estimation of some function $f(x_1, \ldots, x_n)$ consists of a pair of intervals, the inner interval estimation, $f_{inn}(\boldsymbol{t})$ and the outer interval estimation $f(\boldsymbol{x})$. An inner interval estimation must only contain values that are in the actual range of f. We denote the twin estimation of f by

$$f_{twin}(\boldsymbol{t}) = (f_{inn}(\boldsymbol{t}), f(\boldsymbol{x})).$$

The basic arithmetic operations with twins given in [7] are identical with those given in Kaucher arithmetic [2] for the set of proper intervals, i.e., $[a, b]$, where $a < b$.

ATSC evaluates functions specified by algorithms or formulas in such a way that all operations are executed according to the rules of a *twin slope arithmetic* to guarantee inner and outer estimations for the function and slope values. Throughout, $\check{\boldsymbol{x}} = (\check{\boldsymbol{x}}_1, \check{\boldsymbol{x}}_2)$ and $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ will represent twins such that $\check{\boldsymbol{x}}_2 \subseteq \boldsymbol{x}_2$.

Definition 2 *Let* $\boldsymbol{x}$ *and* $\check{\boldsymbol{x}}$ *be real twins and let* $u : \boldsymbol{x}_2 \to \mathbb{R}$ *be a real function. A twin slope for* u *over* $\boldsymbol{x}$ *and centered at* $\check{\boldsymbol{x}}$ *is defined as the twin*

$$\mathbf{S_{twin}}(u, \boldsymbol{x}, \check{\boldsymbol{x}}) = (\mathbf{S_{inn}}(u, \boldsymbol{x}, \check{\boldsymbol{x}}), \mathbf{S}(u, \boldsymbol{x}_2, \check{\boldsymbol{x}}_2)),$$

where $\mathbf{S_{inn}}(u, \boldsymbol{x}, \check{\boldsymbol{x}})$ *and* $\mathbf{S}(u, \boldsymbol{x}_2, \check{\boldsymbol{x}}_2)$*, the inner and outer slope estimations, are obtained according to the rules of a twin slope arithmetic.*

Twin slope arithmetic is based on defining operations and standard functions on automatic twin ordered triplets of the form $\langle\langle \check{\boldsymbol{u}}, \boldsymbol{u}, \boldsymbol{u}^{(s)} \rangle\rangle$, where $\check{\boldsymbol{u}}$, $\boldsymbol{u}$, and $\boldsymbol{u}^{(s)}$ are real twins. $\check{\boldsymbol{u}}$ is the twin evaluation of $u(x)$ over $\check{\boldsymbol{x}}$, $\boldsymbol{u}$ is the twin evaluation of $u(x)$ over $\boldsymbol{x}$ and $\boldsymbol{u}^{(s)}$ is the twin slope $\mathbf{S_{twin}}(u, \boldsymbol{x}, \check{\boldsymbol{x}})$. Inner estimates for slopes are expressed in terms of bounds of intervals, considering concavity conditions of the functions, and executing all intermediate operations with inward rounding. Outer estimates for slopes are obtained with the formulas given in [3] with outward rounding. The following example illustrates the application of ATSC.

Example 1 *Let* $f(x) = x^2 - 4x + 2$*. Considering the interval* $[1, 7]$ *and its midpoint 4, the actual slope is* $\mathbf{S}^\sharp(f, [1, 7], 4) = [1, 7]$*, and the actual range is* $f^u([1, 7]) = [-2, 23]$*. Let* $\boldsymbol{x} = ([1, 7], [1, 7])$ *and* $\check{\boldsymbol{x}} = ([4 - \epsilon, 4 + \epsilon], [4 - \epsilon, 4 + \epsilon])$*, where* ϵ *is large enough so repeated inward rounding does not result in the empty set. The next table presents intermediate evaluations using twin arithmetic and twin slope arithmetic with forward substitution. In this table,* $\boldsymbol{xr}$*,* $\check{\boldsymbol{x}}$*, and* $\boldsymbol{xs}$ *denote the range, center and twin slope evaluations for the intermediate variables respectively (rounded out or in as appropriate to three digits). Also,* **op** *indicates which intermediate operation is performed to compute the displayed result.*

op
$x_1 = x$
$x_2 = x_1^2$
$x_3 = 4x_1$
$x_4 = x_2 - x_3$
$x_5 = x_4 + 2$

op	xr	$\check{x}$	xs
x_1	$([1,7],[1,7])$	$([4,4],[4,4])$	$([1,1],[1,1])$
x_2	$([1,49],[1,49])$	$([16,16],[15.9,16.1])$	$([5,11],[4.99,11.1])$
x_3	$([4,28],[4,28])$	$([16,16],[15.9,16.1])$	$([4,4],[3.99,4.01])$
x_4	$([-3,21],[-27,45])$	$([0,0],[-0.01,0.01])$	$([1.01,6.99],[.999,7.01])$
x_5	$([-1,23],[-25,47])$	$([2,2],[1.99,2.01])$	$([1.01,6.99],[.999,7.01])$

Finally the twin slope, and the twin enclosures evaluation with 15 digits in the computation, are

$$\mathbf{S_{twin}}(f,x,\check{x}) = \begin{array}{l}([1.00000000000001, 6.99999999999997], \\ [0.999999999999982, 7.00000000000004])\,,\end{array}$$

$$f_{twin}(x) = \begin{array}{l}([-.99999999999997, 22.999999999997], \\ [-25.0000000000001, 47.0000000000004])\,.\end{array}$$

References

[1] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990.

[2] E. W. Kaucher, "Interval analysis in the extended interval space $\mathbb{IR}$", *Computing (Suppl.)*, 1980, Vol. 2, pp. 33–49.

[3] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.

[4] V. Kreinovich, V. M. Nesterov, and N. A. Zheludeva, "Interval methods that are guaranteed to underestimate (and the resulting new justification of Kaucher arithmetic)", *Reliable Computing*, 1996, Vol. 2, No. 2, pp. 119–124.

[5] L. Moreau and D. Aeyels, "Optimization of discontinuous functions: A generalized theory of differentiation", *SIAM Journal of Optimization*, 2000, Vol. 11, No. 1, pp. 53–69.

[6] H. Muñoz, *Theory and Practice in Verified Nonsmooth Optimization*, PhD thesis, University of Louisiana at Lafayette, 2001.

[7] V. M. Nesterov, "Interval and twin arithmetics", *Reliable Computing*, 1997, Vol. 3, pp. 369–380.

Two Topics in Computer Assisted Proofs for the Problems in Fluid Dynamics

Mitsuhiro T. Nakao

Faculty of Mathematics
Kyushu University 33
Fukuoka 812-8581, Japan
e-mail: mtnakao@math.kyushu-u.ac.jp

We have been devoted for years to studying the numerical verifications of solutions to elliptic partial differential equations. Our approach is based on the combination of fixed point theorems in functional spaces and the constractive error estimations of finite element (or spectral) method. In our verification process, the interval method for finite-dimensional linear equations plays an essential role.

In this talk, we first briefly describe the basic idea of our verification method for nonlinear elliptic problems. Next, we apply the method to two important problems appeared in fluid dynamics, i.e., Rayleigh-Bénard and Kolmogorov problems. In both cases, the existence of exact solutions is verified and the usefulness of our approach have been shown.

1 The Basic Idea ([4, 5])

Suppose that the concerned elliptic problem is reformulated as the following fixed point problem of a nonlinear compact operator F in some appropriate infinite-dimensional function space X:

$$u = F(u). \tag{1}$$

Suppose also that we find a nonempty, bounded, convex, and closed subset $U \subset X$, which is referred to as a *candidate set* of solutions, satisfying

$$F(U) = \{F(u) | u \in U\} \subset U. \tag{2}$$

Then by the Schauder fixed point theorem, an infinite-dimensional version of Brouwer's theorem, there exists an element $u \in F(U)$ such that $u = F(u)$.

Let S_h be a finite-dimensional subspace of X dependent on h $(0 < h < 1)$. Let $P_h : X \longrightarrow S_h$ be the orthogonal projection operator, where the parameter

h corresponds to the degree of approximation. For example, it means the mesh size in the finite element methods or the reciprocal of the term number in the spectral approximations. We usually choose a candidate set U of the form $U = U_h \oplus U_\perp$, where $U_h \subset S_h$ and $U_\perp \subset S_h^\perp$. Here, $S_h^\perp$ stands for the orthogonal complement subspace of S_h in X. Then, the verification condition (2) can be decomposed into the two parts as follows:

$$\begin{cases} P_h F(U) & \subset & U_h \\ \\ (I - P_h)F(U) & \subset & U_\perp. \end{cases} \tag{3}$$

Since the first inclusion is in the finite-dimensional space S_h, it may be verified on computer using interval arithmetic. The second inclusion is in the infinite-dimensional space $S_h^\perp$, and will be verified by constructive error analysis of the numerical method in use. Combining verifications of both inclusions in (3) we may conclude the inclusion (2) is verified.

The set U_h consists of linear combinations of base functions in S_h with interval coefficients, and the set $U_\perp$ is constructed as a ball in $S_h^\perp$ with radius $\alpha \geq 0$. Namely, we represent U_h and $U_\perp$ by

$$U_h = \sum_{j=1}^{M} [\underline{A}_j, \overline{A}_j]\phi_j \quad \text{and} \quad U_\perp = \{\phi \in S_h^\perp \mid \|\phi\|_{H_0^1} \leq \alpha\},$$

respectively, where $\{\phi_j\}_{j=1}^M$ is a basis of S_h. Here, $\sum_{j=1}^{M} [\underline{A}_j, \overline{A}_j]\phi_j$ is interepreted as the set of functions in which each element is a linear combinaion of $\{\phi_j\}_{j=1}^M$ whose coefficient of ϕ_j belongs to the corresponding interval $[\underline{A}_j, \overline{A}_j]$ for each $1 \leq j \leq M$.

Then, it can be easily seen that $P_h F(U)$ is directly computed or enclosed of the form

$$P_h F(U) \subset \sum_{j=1}^{M} [\underline{B}_j, \overline{B}_j]\phi_j$$

by solving a linear system of equations with interval right-hand side which is determined from U_h and $U_\perp$ using interval computations. Thus, the first condition in (3) is validated as the inclusion relations of corresponding coefficient intervals, that is, $[\underline{B}_j, \overline{B}_j] \subset [\underline{A}_j, \overline{A}_j]$. On the other hand, $(I - P_h)F(U)$ is not directly computable but can be numerically evaluated by the effective use of constructive *a priori* error estimates of the projection P_h. Hence, the second condition can be verified by a simple comparison of two nonnegative real numbers which correspond to the radii of balls. In the actual computation, we use some iterative methods for both part of $P_h F(U)$ and $(I - P_h)F(U)$.

In order to apply the verification method to more general problems, we usually utilize a version of Newton-like method (see e.g., [5], [6] for details) which is also considered as an extension of the interval Newton method (e.g., [1]) to

the infinite-dimensional cases. We also note that, in our verification, we estimate rigorously not only the rounding error of floating point computations, but also the truncation error due to the approximation of the infinite-dimensional operator. Therefore, our method can also be applied to the guaranteed *a posteriori* error analysis for the various kinds of approximation methods for elliptic problems.

2 Heat Convection Problems Governed by the Navier-Stokes Equation

The two-dimensional (x-z) Oberbeck-Boussinesque approximations for the Rayleigh-Bénard convection are described as follows [7]:

$$\begin{cases} u_t + uu_x + wu_z &= p_x + \mathcal{P}\Delta u, \\ w_t + uw_x + ww_z &= p_z - \mathcal{P}\mathcal{R}\theta + \mathcal{P}\Delta w, \\ u_x + w_z &= 0, \\ \theta_t + w + u\theta_x + w\theta_z &= \Delta\theta, \end{cases} \tag{4}$$

where (u, w), p and θ denote the velocity field, pressure and temperature from a linear profile while $\mathcal{P}$ and $\mathcal{R}$ denote Prandtl and Rayleigh numbers, respectively. We consider the steady-state solution branches of (4). By using the stream function Ψ for the velocity and setting $\Theta \equiv \sqrt{\mathcal{P}\mathcal{R}}\theta$, we have the following system of equations on the domain $\{-\infty < x < \infty, 0 < z < \pi\}$.

$$\begin{cases} \mathcal{P}\Delta^2\Psi &= \sqrt{\mathcal{P}\mathcal{R}}\Theta_x - \Psi_z\Delta\Psi_x + \Psi_x\Delta\Psi_z \\ -\Delta\Theta &= -\sqrt{\mathcal{P}\mathcal{R}}\Psi_x + \Psi_z\Theta_x - \Psi_x\Theta_z \\ \Psi = 0 \quad , & \quad \Psi_{zz} = 0, \quad \Theta = 0 \qquad on \;\; z = 0,\; \pi \end{cases} \tag{5}$$

We suppose the periodic boundary condition in x and the stress free boundary condition on $z = 0$ and $z = \pi$. We have numerically verified several solution branches from the trivial solution of (5) by using the spectral approximations and the constructive error estimates. Several new results which would be difficult to derive by theoretical approaches are obtained.

3 Kolmogorov's Problem of Viscous Incompressible Fluid

This is a non-selfadjoint eigenvalue problem of the linearlized stationary Navier-Stokes equation in two dimension of the following form [3]:

Find a stream function ϕ, periodic in x and y, and a number $\sigma \in R^1$ such

that

$$\begin{cases} \dfrac{1}{R}\Delta^2\phi - \sin y(\Delta + I)\dfrac{\partial\phi}{\partial x} &= \sigma\Delta\phi, \qquad (x,y) \in T_\alpha \\ \displaystyle\int_{T_\alpha} \phi^2 dxdy &= 1, \end{cases} \tag{6}$$

where R is the Reynolds number, $T_\alpha \equiv [-\pi/\alpha, \pi/\alpha] \times [-\pi, \pi]$ (α: aspect ratio).

The final purpose of the computer assisted proof is the validation of a stability condition of the flow. This can be carried out by showing that a certain inequality holds for the numerically verified eigenpair (σ, ϕ). Using the Fourier-Galerkin method with explicit error estimates as in the previous problem, we have actually succeeded to verify stability results related to the aspect ratio α. Proving this result would also be very difficult by any kind of theoretical analysis up to now.

In the presentation, we will show some numerical examples of the above topics. In both examples we use the spectral method. Note that it is also possible to use the finite element approximation with constructive error estimates in stead of the spectral method.

References

[1] G. Alefeld, "On the convergence of some interval-arithmetic modifications of Newton's method", *SIAM J. Numer. Anal.*, 1984, Vol. 21, pp. 363–372.

[2] V. Girault and P. A. Raviart, *Finite element methods for Navier-Stokes equations*, Springer-Verlag, Berlin, Heidelberg, 1986.

[3] K. Nagatou, *A computer assisted proof on the stability in Kolmogorov's problem of viscous incompressible fluid on 2-D flat tori*, preprint.

[4] M. T. Nakao, "A numerical approach to the proof of existence of solutions for elliptic problems", *Japan Journal of Applied Mathematics*, 1988, Vol. 5, pp. 313–332.

[5] M. T. Nakao and N. Yamamoto, "Numerical verification of solutions for nonlinear elliptic problems using L^∞ residual method", Journal of Mathematical Analysis and Applications 1998, Vol. 217, pp. 246–262.

[6] M. T. Nakao, "Numerical verification methods for solutions of ordinary and partial differential equations", *Numerical Functional Analysis and Optimization*, 2001, Vol. 22, No. 3/4, pp. 321–356.

[7] Y. Watanabe, M. T. Nakao, N. Yamamoto, and T. Nishida, "A numerical verification of nontrivial solutions for heat convection problems", *Journal of Mathematical Fluid Mechanichs*, 2002 (to appear).

Generation of Bode and Nyquist Plots for Nonrational Transfer Functions to Prescribed Accuracy

P. S. V. Nataraj[1] and J. J. Barve[2]

[1]Associate Professor and [2]Research Scholar
Systems and Control Engineering
Indian Institute of Technology
Bombay, Mumbai 400 076, India
{nataraj,jjbarve}@ee.iitb.ac.in

Abstract

We present interval analysis based procedures for construction of the well-known Bode and Nyquist frequency response plots for nonrational transfer functions. The proposed procedures can be used to construct the plots reliably and to a prescribed accuracy over a user-specified frequency range. The procedures overcome the limitations of the only available method for nonrational transfer functions that is based on arbitrary gridding of the given frequency range. Several important examples drawn from various branches of engineering are used to demonstrate the merits of the proposed procedures.

Keywords: Frequency Response, Interval Analysis, Bode plots, Nyquist plots, Nonrational Transfer Functions.

1 Introduction

For over five decades, the Bode and Nyquist frequency response plots have been of great use in frequency domain analysis and synthesis of linear systems, see, for instance, [1, 6, 11]. For transfer functions (TFs) having a rational form, an automatic frequency grid selection procedure is available in the MATLAB toolbox [4] to generate the frequency response plots. However, this procedure has several limitations:

1. it does not guarantee that the generated plots are of a user-specified accuracy,

2. it uses an unreliable phase unwrapping procedure that can be fooled if a suitably fine frequency grid is not chosen in the frequency region having sharp phase changes, and

3. it is not applicable to the large and important class of nonrational transfer functions.

The class of nonrational transfer functions is of great practical importance, especially in chemical process control where virtually every process has significant time-delays (a time delay is modeled as a $e^{-\tau_d s}$ term, where τ_d is the amount of time delay, and this leads to a nonrational transfer function). Some of the application areas where nonrational transfer functions can be found are:

1. pressure fluctuations in a long flexible hose-tube connecting servo-valve to actuator in hydraulic servo system [3],

2. feedback system with measurement time delays [10],

3. heating of a one dimension metal rod along its length by a steam chest [12],

4. heat-exchanger systems [12],

5. multi-modal reactor systems in nuclear reactors [2], and

6. flexible or smart structures.

At the present time, the *only* method for generating the Bode and Nyquist frequency response plots for such nonrational transfer functions is through arbitrary rastering or gridding of the frequency range of interest. However, as is well-known, this so-called gridding method has significant limitations: (a) the number of grid points required to obtain a specified accuracy is unknown, and (b) for a given frequency response plot, the amount of error present is unknown, i.e., no error estimates are available. These limitations show up particularly severely when the frequency responses exhibit single or multiple sharp peaks or dips (this happens for the application systems we mentioned above). Despite the severe limitations of the gridding method, surprisingly little effort has been made in the literature to overcome them.

In this work, we propose a procedure each to generate the Bode and Nyquist frequency response plots for nonrational transfer functions. Since our procedures are based on a Vector - Adaptive subdivision and evaluation strategy, we call them as *VA procedures*. *VA procedures are guaranteed to automatically* generate the plots *reliably* and to a *prescribed* accuracy, throughout a given frequency range. The VA procedures are applicable to a very general class of transfer functions in the continuous as well as in the discrete-time domains. Transfer functions involving a composition of time-delay and transcendental terms can be handled equally easily in the VA procedures, without the need for any approximations. Moreover, error estimates are readily available from all plots that have been generated by the VA procedures.

2 A Procedure for Bode Plot Generation

We present the proposed VA procedure for Bode plot generation. A similar procedure can be given for the Nyquist plot generation.

The Vector-Adaptive Procedure (VA) for Bode Plot Construction

- Inputs : An expression for the transfer function $g(s)$, the frequency interval Ω of interest, and the specified maximum width ε of each magnitude and phase rectangle in the generated Bode plot. In general, ε can be different for magnitude and phase plots.
- Output: A collection of magnitude and phase rectangles, each of width at most ε, and enclosing the actual Bode magnitude and phase plot.

BEGIN Procedure

1. From the transfer function expression $g(s)$, obtain the magnitude and phase expressions $f_{mag}(\omega)$ and $f_{phase}(\omega)$, where ω is the frequency variable.
2. Construct natural interval extensions $F_{mag}(\Omega), F_{phase}(\Omega)$ for $f_{mag}(\omega)$, $f_{phase}(\omega)$, respectively.
3. Set current frequency subinterval as Ω and set the solution list L^{sol} as empty.
4. (Adaptive subdivision and vectorized evaluation)
   (a) Subdivide all current frequency subintervals, and discard the original subintervals.
   (b) Using vectorized operations, perform vectorized *evaluation* of $F_{mag}(\Omega)$ over the frequency subintervals obtained in above substep.
   (c) Deposit all magnitude rectangles whose widths are less than ε in the solution list L^{sol}, and discard the corresponding frequency subintervals from further processing [1]. Keep the remaining frequency subintervals in the current frequency list for further processing.
   (d) If there are no more frequency subintervals left for processing, go to the following step. Else, go back to the beginning of this step (of adaptive subdivision and vectorized evaluation), and repeat.
5. Output the generated Bode magnitude plot as the collection of all magnitude rectangles' present in the solution list L^{sol}.
6. Repeat the above three steps but for $F_{phase}(\Omega)$. Output the generated Bode phase plot as the collection of all phase rectangles present in the solution list L^{sol}.

END Procedure.

[1] The corresponding frequency subintervals are no longer needed as these have produced small enough magnitude rectangles which have been just stored.

3 Results and Discussion

We test the performance of the proposed VA procedures on several real-life nonrational transfer function examples. We also test them on some challenging rational transfer function examples. The examples are chosen from the application problems listed above.

We program the VA procedures using the interval analysis toolbox INTLAB [14] in the MATLAB environment. We carry out all computations on a PC/Pentium-III 550 MHz machine. In all the examples, we set the prescribed accuracy as $\varepsilon_{\text{mag}} = 1$ decibel (dB) and $\varepsilon_{\text{phase}} = 1$ deg. This means that magnitude (resp. phase) side of each box in the plot is to have a width at most of 1 dB (resp. 1 deg.).

We compare the frequency response plots generated using the VA procedures with those obtained using conventional rastering or gridding of the frequency interval, for three different grids of $10^2, 10^3$, and 10^4 grid points. Further, we benchmark all results against the plot obtained using a very dense grid of 5×10^5 grid points.

The results of the examples show that the gridding method yields large errors in the plots, if the frequency grid size is not carefully chosen. For instance, in our examples, it was found that grids of 10^4 grid points were often required, and in some extreme cases, even grids of 10^5 grid points were inadequate to obtain the same accuracy. Further, the accuracy of the obtained frequency response plots is unknown unless and until these are benchmarked against the "exact" plots (hopefully obtained using very dense grids). Without a good estimate of the grid points to be used and of the error present in the generated plots, there is every *danger* that one may be lead to erroneous analysis and synthesis results. The proposed procedures relieve the user of the difficulties associated with grid point selection and lack of error estimates.

(The Table containing the comparative analysis of errors, and the plots of frequency responses, are not given here due to space constraints but will be given in the full paper).

References

[1] H. W. Bode, *Network Analysis and Feedback Design*, Van Nostrand, New York, 1945.

[2] Y. L. Chen and K. W. Han, "Stability analysis of a nonlinear reactor control system", *IEEE Transactions on Nuclear Science*, 1971, Vol. NS-18, pp. 18–25.

[3] E. O. Doebelin, *Control system principles and design*, John Wiley and sons, New York, 1985.

[4] A. Grace, A. J. Laub, J. N. Little, and C. M. Thompson, *Control system toolbox for use with MATLAB: User guide*, The MathWorks Inc., MA, 1992.

[5] E. Hansen, *Global optimization using interval analysis*, Marcel Dekker, 1992.

[6] I. J. Nagarath and M. Gopal, *Control system engineering*, Wiley Eastern, New Delhi, 1995.

[7] R. B. Kearfott, *Rigorous global search: continuous problems*, Kluwer Academic Publishers, Dordrecht, 1996.

[8] R. E. Moore, *Interval analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[9] R. E. Moore, *Methods and applications of interval analysis*, SIAM, Philadelphia, 1979.

[10] M. Malek-Zavarei and M. Jamshidi, *Time delay systems: Analysis, optimization and application*, North-Holland, 1987.

[11] K. Ogata, *Modern control engineering*, Prentice Hall of India, New Delhi, 1997.

[12] B. A. Ogunnike and W. H. Ray, *Process Dynamics, Modelling and Control*, Oxford University Press, New York, 1994.

[13] H. Ratschek and J. Rokne, *Computer methods for the range of functions*, Ellis Horwood Ltd, Chichester, 1984.

[14] S. M. Rump, "INTLAB – interval laboratory", In: T. Csendes (ed.), *Developments in reliable computing*, Kluwer Academic Publishers, 1999.

A New Super-Convergent Inclusion Function Form and its Use in Global Optimization

P. S. V. Nataraj[1] and K. Kotecha[2]

[1]Systems and Control Engineering Group
EE Department, IIT Bombay 400076, India
Email: nataraj@ee.iitb.ernet.in
[2]Department of IT
GH Patel College of Engineering and Technology
Vallabh Vidyanagar, 388120, Gujarat, India
Email: kk@gcet.ac.in

Abstract

Recently, Lin and Rokne [10] introduced the so-called Taylor-Bernstein form as an inclusion function form for multidimensional functions. This form was theoretically shown to have the super-convergence property. Here, we present an improvement of Lin and Rokne's Taylor-Bernstein form to make it more effective in practice. We test and compare the super-convergence behavior of the proposed form with that of Lin and Rokne's Taylor-Bernstein form and also with that of the Taylor model of Berz *et al.* [3]. We obtain super-convergence of orders up to 9 with the proposed form. Moreover, with the proposed form we quite easily obtain such high orders of super-convergence for up to $5-$dim problems.

We also investigate the use of higher order inclusion functions in the Moore-Skelboe (MS) algorithm of interval analysis (IA) for unconstrained global optimization. We use the improved TB form as an inclusion function in a prototype MS algorithm and also modify the cut-off test and termination condition in the algorithm. We test and compare on several examples the performances of the proposed algorithm, the MS algorithm, and the MS algorithm with the Taylor model of Berz *et al.* [3] as inclusion function. The results of these (preliminary) tests indicate that the proposed algorithm with the improved TB form as inclusion function is quite effective for low to medium dimension problems studied.

1 Introduction

An important problem in interval analysis is the construction of inclusion functions having the property of so-called *super-convergence* (i.e., having a convergence order that is greater than quadratic) for multidimensional functions.

Such inclusion functions have applications in the solutions of equations, optimization, quadrature, and others. The first paper in the literature concerning super-convergence is that of Herzberger [6], who shows that super-convergence can be obtained for a certain class of intervals. However, his requirement on the function is unrealistically strong. Cornelius and Lohner [4] propose the interpolation and remainder forms for multidimensional functions that enable any convergence order to be obtained in theory. However, in practice, convergence order of at most 4 or 5 is recommended even for unidimensional functions, see [4] and [15, pg. 9]. The same holds for the improved version of these forms for unidimensional functions, as proposed by Neumaier in [14, sec. 2.4]. Alefeld and Lohner [1] propose centered forms with super-convergence for *unidimensional* functions. However, because of the strong condition on the functional representation, these higher order centered forms have limited practical value [1, pg. 8]. Lin and Rokne [10] propose super-convergent forms that combine Taylor and Bernstein (or B-spline) forms for multidimensional functions. However, for small domains these forms become computationally very demanding, even for unidimensional functions, see [10, pg. 108]. Berz *et al.* [3, 12] propose the so-called Taylor models for multidimensional functions. Although the accuracy of the so-called remainder interval part of the Taylor model increases in a super-convergent fashion, the Taylor model itself is known to exhibit only quadratic convergence see Kearfott and Arazyan [9].

We propose in this work a new inclusion function form having the super-convergence property for multidimensional functions. The proposed inclusion function form uses Bernstein polynomials for bounding the range of the polynomial obtained from the Taylor form of the function f. The Bernstein algorithm is combined with the Taylor form to obtain the resulting so-called Taylor-Bernstein form as an inclusion function form of f. The proposed Taylor-Bernstein form has some important differences (in the practical way it is constructed) from the Taylor-Bernstein form of Lin and Rokne [10].

We numerically investigate the super-convergence property of the above inclusion function forms on some benchmark examples. The selected examples are of low to medium dimensions. For all our computations, we use a PC/Pentium III 800 MHz 256 MB RAM machine with a FORTRAN 90 compiler, and version 8.1 of the COSY-INFINITY package of Berz *et al.* [2, 7]. We also investigate the performance of the Taylor model as an inclusion function form in these examples. With the proposed form, we quite easily obtain super-convergence (of orders up to 9) for low to medium dimensional problems. To our knowledge, it is perhaps for the first time that super-convergence of such high orders has actually been demonstrated on multidimensional problems. Moreover, the new super-convergent form can be constructed on a computer with the fully automated algorithm presented, without any need for hand computations.

We next use the new super-convergent form to solve the following optimization problem. Let $\Re$ be the set of reals, $\mathbf{X} \subseteq \Re^l$ be a right parallelepiped parallel to the axes (also called as a box), and $f : \mathbf{X} \to \Re$ be a $m+1$ times differentiable function for some positive integer m. Let $\bar{f}(\mathbf{X})$ denote the set of all values of f on $\mathbf{X}$. We seek global optimization algorithms that are able to efficiently

determine arbitrarily good lower bounds for the minimum of $\bar{f}(\mathbf{X})$.

Many algorithms based on interval analysis (IA) are available to solve this global optimization problem, see for example, [5], [8], [16] and the references cited therein. A basic branch and bound algorithm of IA is the so-called Moore-Skelboe (MS) algorithm [16]. Although the MS algorithm is reliable, it is somewhat slow to converge in 'difficult' problems, when inclusion functions of first and sometimes even second orders are used. Faster convergence could possibly be obtained with higher order inclusion functions, and it is of interest in this work to investigate their effectiveness in some such 'difficult' problems.

Our proposed algorithm for global optimization uses the new super-convergent form having high order convergence, and we therefore expect to obtain faster convergence with this form. The new form also allows us to make the cut-off test and termination condition more effective, and we incorporate these modifications in the proposed algorithm. Since this global optimization algorithm involves using the new **T**aylor - **B**ernstein form in **M**oore-**S**kelboe algorithm, we call it as Algorithm TBMS.

We can also have the Taylor model of Berz *et al.* as an inclusion function form in the MS algorithm as done, for instance, in the preliminary work in [9]. We call such an algorithm as Algorithm TMS.

We test and compare the performance of the proposed algorithm with that of Algorithms TMS and MS on six 'difficult' examples. These preliminary tests indicate that Algorithms TMS and TBMS are quite effective compared to Algorithm MS, for lower accuracy problems. For higher accuracy problems, Algorithm TBMS is the most effective one. The best overall choice, in terms of the number of iterations, space-complexity, and speed seems to be Algorithm TBMS with a medium Taylor order $m = 4$.

2 Numerical Results for super convergence

We numerically investigate the super-convergence property of the above inclusion function forms on some benchmark examples.

In each example, we compute the intervals

$F_{TM}(\mathbf{X})$ – using Taylor model of Berz *et al.* [11], computed with the COSY-INFINITY package.

$F_{LR}(\mathbf{X})$ – using Taylor-Bernstein form of Lin and Rokne, computed with Algorithm LR.

$F_{TB}(\mathbf{X})$ – using the proposed Taylor-Bernstein form, computed with Algorithm TB.

$F_{inner}(\mathbf{X})$ – using *inner* estimates of the range computed with the well-known Moore-Skelboe optimization algorithm of interval analysis (see, for instance, [16]).

Let $\mathbf{X} = [a, b], \mathbf{Y} = [c, d]$ be any two intervals. Then, following [4], as a measure of the overestimation we use the Hausdorff metric

$$\mathcal{H}(\mathbf{X}, \mathbf{Y}) = |[a, b], [c, d]| = \max\{|a - c|, |b - d|\}$$

Consider a sequence of nested intervals $\{\mathbf{X}^{(i)}\}$. We wish to find and compare for each form, the reduction in overestimation with decrease in the domain interval widths. Consider first the form F_{TM}. Let

$$\mathcal{H}_{TM}\left(\mathbf{X}^{(i-1)}\right) := \mathcal{H}\left(\bar{f}(\mathbf{X}^{(i-1)}), F_{TM}\left(\mathbf{X}^{(i-1)}\right)\right) \tag{1}$$

As a measure of the reduction in overestimation obtained with form F_{TM} over successive nested intervals $\mathbf{X}^{(i-1)}$ and $\mathbf{X}^{(i)}$, we use the ratio

$$\mathcal{R}_{TM}\left(\mathbf{X}^{(i-1)}, \mathbf{X}^{(i)}\right) := \frac{\mathcal{H}_{TM}\left(\mathbf{X}^{(i-1)}\right)}{\mathcal{H}_{TM}\left(\mathbf{X}^{(i)}\right)} = \frac{\mathcal{H}\left(\bar{f}(\mathbf{X}^{(i-1)}), F_{TM}\left(\mathbf{X}^{(i-1)}\right)\right)}{\mathcal{H}\left(\bar{f}(\mathbf{X}^{(i)}), F_{TM}\left(\mathbf{X}^{(i)}\right)\right)}$$

Define

$$\mathcal{R}^*\left(\mathbf{X}^{(i-1)}, \mathbf{X}^{(i)}\right) := \left(\frac{w\left(\mathbf{X}^{(i-1)}\right)}{w\left(\mathbf{X}^{(i)}\right)}\right)^{m+1}$$

If F_{TM} is an inclusion function form having convergence order $m+1$, then

$$\mathcal{R}_{TM}\left(\mathbf{X}^{(i-1)}, \mathbf{X}^{(i)}\right) \to \mathcal{R}^*\left(\mathbf{X}^{(i-1)}, \mathbf{X}^{(i)}\right) \tag{2}$$

(where the tending is from above) for "small" enough $w\left(\mathbf{X}^{(i-1)}\right), w\left(\mathbf{X}^{(i)}\right)$.

In practice, the exact range $\bar{f}$ is generally difficult to compute, so the overestimation can be generally found relative only to some *inner* estimate F_{inner} of the range. However, we can easily show that if the $(m+1)$- th convergence order property holds relative to F_{inner}, then it implies that the same holds relative to the exact range $\bar{f}$. That is, it is sufficient if we can show the $(m+1)$- th convergence order property with F_{inner} used in place of $\bar{f}$ in above definitions. To avoid introducing more notation, in the sequel we use the quantities given in (1) through (2), with F_{inner} replacing $\bar{f}$ throughout.

Similarly, we can define $\mathcal{H}_{LR}, \mathcal{H}_{TB}, \mathcal{R}_{LR}, \mathcal{R}_{TB}$ for the forms F_{LR} and F_{TB}. For brevity of notation, we drop the arguments $\mathbf{X}^{(i-1)}, \mathbf{X}^{(i)}$ of all $\mathcal{H}$ and $\mathcal{R}$.

Example 1. Trigonometric function [13, problem 26]. The $4-$dim function is

$$f(x) = \sum_{i=1}^{4} f_i(x)^2, f_i(x) = 4 - \sum_{j=1}^{4} \cos x_j + i(1 - \cos x_i) - \sin x_i$$

The domain is $\mathbf{X}^{(i)} = \left[1.75 + 2^{-i}[-1,1]\right]^4$.

For Taylor order $m = 2$:

i	0	1	2	3
$w\left(\mathbf{X}^{(i)}\right)$	$2 * 2^{-0}$	$2 * 2^{-1}$	$2 * 2^{-2}$	$2 * 2^{-3}$
$\mathcal{H}_{TM}$	$4E+2$	$9E+1$	$2E+1$	$5E+0$
$\mathcal{H}_{LR}$	$3E+2$	$3E+1$	$3E+0$	$*$
$\mathcal{H}_{TB}$	$3E+2$	$3E+1$	$3E+0$	$3E-1$
$\mathcal{R}^*$	$-$	8	8	8
$\mathcal{R}_{TM}$	$-$	4.9	4.5	4.2
$\mathcal{R}_{LR}$	$-$	10.5	9.5	$-$
$\mathcal{R}_{TB}$	$-$	10.5	9.5	8.8

i	4	5	6	7
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-4}$	$2*2^{-5}$	$2*2^{-6}$	$2*2^{-7}$
$\mathcal{H}_{TM}$	$1E+0$	$3E-1$	$7E-2$	$2E-2$
$\mathcal{H}_{LR}$	$*$	$*$	$*$	$*$
$\mathcal{H}_{TB}$	$3E-2$	$4E-3$	$5E-4$	$7E-5$
$\mathcal{R}^*$	8	8	8	8
$\mathcal{R}_{TM}$	4.1	4.1	4.0	4.0
$\mathcal{R}_{LR}$	–	–	–	–
$\mathcal{R}_{TB}$	8.4	8.2	8.1	8.1

For Taylor order $m = 4$:

i	0	1	2	3
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-0}$	$2*2^{-1}$	$2*2^{-2}$	$2*2^{-3}$
$\mathcal{H}_{TM}$	$4E+2$	$9E+1$	$2E+1$	$5E+0$
$\mathcal{H}_{LR}$	$1E+1$	$2E-1$	$*$	$*$
$\mathcal{H}_{TB}$	$1E+1$	$2E-1$	$5E-3$	$1E-4$
$\mathcal{R}^*$	–	32	32	32
$\mathcal{R}_{TM}$	–	4.9	4.4	4.2
$\mathcal{R}_{LR}$	–	56.3	–	–
$\mathcal{R}_{TB}$	–	56.3	50.4	44.5

i	4	5	6	7
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-4}$	$2*2^{-5}$	$2*2^{-6}$	$2*2^{-7}$
$\mathcal{H}_{TM}$	$1E+0$	$3E-1$	$7E-2$	$2E-2$
$\mathcal{H}_{LR}$	$*$	$*$	$*$	$*$
$\mathcal{H}_{TB}$	$3E-6$	$8E-8$	$2E-9$	$7E-11$
$\mathcal{R}^*$	32	32	32	32
$\mathcal{R}_{TM}$	4.1	4.1	4.0	4.0
$\mathcal{R}_{LR}$	–	–	–	–
$\mathcal{R}_{TB}$	39.8	36.5	34.3	30.6

For Taylor order $m = 6$:

i	0	1	2	3
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-0}$	$2*2^{-1}$	$2*2^{-2}$	$2*2^{-3}$
$\mathcal{H}_{TM}$	$4E+2$	$9E+1$	$2E+1$	$5E+0$
$\mathcal{H}_{LR}$	$2E+0$	$9E-3$	$*$	$*$
$\mathcal{H}_{TB}$	$2E+0$	$9E-3$	$6E-5$	$4E-7$
$\mathcal{R}^*$	–	128	128	128
$\mathcal{R}_{TM}$	–	4.9	4.4	4.2
$\mathcal{R}_{LR}$	–	189.0	–	–
$\mathcal{R}_{TB}$	–	189.0	167.6	151.3

i	4	5	6	7
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-4}$	$2*2^{-5}$	$2*2^{-6}$	$2*2^{-7}$
$\mathcal{H}_{TM}$	$1E+0$	$3E-1$	$7E-2$	$2E-2$
$\mathcal{H}_{LR}$	$*$	$*$	$*$	$*$
$\mathcal{H}_{TB}$	$3E-9$	$3E-11$	$7E-12$	$7E-12$
$\mathcal{R}^*$	128	128	128	128
$\mathcal{R}_{TM}$	4.1	4.1	4.0	4.0
$\mathcal{R}_{LR}$	–	–	–	–
$\mathcal{R}_{TB}$	140.5	99.2	3.6	1.0

For Taylor order $m = 8$:

i	0	1	2	3
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-0}$	$2*2^{-1}$	$2*2^{-2}$	$2*2^{-3}$
$\mathcal{H}_{TM}$	$4E+2$	$9E+1$	$2E+1$	$5E+0$
$\mathcal{H}_{LR}$	$5E-1$	$6E-5$	$*$	$*$
$\mathcal{H}_{TB}$	$5E-1$	$6E-5$	$8E-8$	$1E-10$
$\mathcal{R}^*$	–	512	512	512
$\mathcal{R}_{TM}$	–	4.9	4.4	4.2
$\mathcal{R}_{LR}$	–	828.6	–	–
$\mathcal{R}_{TB}$	–	828.6	734.6	623.2

i	4	5	6	7
$w\left(\mathbf{X}^{(i)}\right)$	$2*2^{-4}$	$2*2^{-5}$	$2*2^{-6}$	$2*2^{-7}$
$\mathcal{H}_{TM}$	$1E+0$	$3E-1$	$7E-2$	$2E-2$
$\mathcal{H}_{LR}$	$*$	$*$	$*$	$*$
$\mathcal{H}_{TB}$	$8E-12$	$7E-12$	$7E-12$	$7E-12$
$\mathcal{R}^*$	512	512	512	512
$\mathcal{R}_{TM}$	4.1	4.1	4.0	4.0
$\mathcal{R}_{LR}$	–	–	–	–
$\mathcal{R}_{TB}$	17.2	1.1	1.0	0.9

3 Numerical Tests for Global Optimization

We test and compare the performances of Algorithms TBMS, TMS, and MS on various examples. Here we present one 3-dim example.

Example 2. Bard function [13, problem 8]. The three dimensional function is

$$f(x) = \sum_{i=1}^{15} f_i(x)^2, \quad f_i(x) = y_i - \left(x_1 + \frac{u_i}{v_i x_2 + w_i x_3}\right),$$

$$u_i = i, v_i = 16 - i, w_i = \min(u_i, v_i)$$

where,

i	1	2	3	4	5	6	7	8
y_i	0.14	0.18	0.22	0.25	0.29	0.32	0.35	0.39
i	9	10	11	12	13	14	15	
y_i	0.37	0.58	0.73	0.96	1.34	2.10	4.39	

We take the initial domain as $([-0.25, 0.25], [0.01, 2.5], [0.01, 2.5])$ .The performances of the various Algorithms are as under.

		TBMS			
Order, m	Accuracy	Iterations	Time, s	Max. LL	Final LL
2	10^{-3}	406	16.64	74	45
	10^{-5}	520	32.13	74	7
4	10^{-3}	191	35.00	38	7
	10^{-5}	202	60.65	38	1
6	10^{-3}	162	67.80	38	2
	10^{-5}	165	90.22	38	1
8	10^{-3}	157	79.90	38	2
	10^{-5}	159	92.03	38	1

		TMS			
Order, m	Accuracy	Iterations	Time, s	Max. LL	Final LL
2	10^{-3}	3145	76.13	822	772
	10^{-5}	*	$>$ 3600	*	*
4	10^{-3}	3124	86.13	818	772
	10^{-05}	*	$>$ 3600	*	*
6	10^{-3}	3123	122.81	818	772
	10^{-05}	*	$>$ 3600		
8	10^{-3}	3122	181.05	818	772
	10^{-5}	*	$>$ 3600	*	*

MS				
Accuracy	Iterations	Time, s	Max. LL	Final LL
10^{-03}	6122	466.56	1643	1622
10^{-05}	*	$>$ 3600	*	*

The global minima found using each of the algorithms is $8.21487....E-03$.

4 Summary

We proposed a new super-convergent inclusion form for multidimensional functions form and quite easily obtained super-convergence (of order up to 9) for low to medium dimensional problems. We also tested and found the new form to be quite effective in all six global optimization problems that were selected for the tested, in terms of number of iterations, space-complexity and speed.

References

[1] G. Alefeld and R. Lohner, "On higher order centered forms", *Computing*, 1985, Vol. 35, pp. 177–184.

[2] M. Berz and J. Hoefkens, *COSY INFINITY Version 8.1 Programming Manual*, Technical Report MSUCL-1196, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 2001.

[3] M. Berz and G. Hoffstatter, "Computation and application of Taylor polynomials with interval remainder bounds", *Reliable Computing*, 1998, Vol. 4, pp. 83–97.

[4] H. Cornelius and R. Lohner, "Computing the range of values of real functions with accuracy higher than second order", *Computing*, 1984, Vol. 33, pp. 331–347.

[5] E. Hansen, *Global optimization using interval analysis*, Marcel Dekker, 1992.

[6] J. Herzberger, "Zur approximation des wertebereiches reeller funktionen durch intervallausdrucke", *Computing, Supplement*, 1977, Vol. 1, pp. 57–64.

[7] J. Hoefkens, *Rigorous Numerical Analysis with High-Order Taylor Models*, Ph.D. thesis, Michigan State University, East Lansing, Michigan, USA, 2001; also avaliable as Technical Report MSUCL-1217.

[8] R. B. Kearfott, *Rigorous global search: continuous problems*, Kluwer Academic Publishers, Dordrecht, 1996.

[9] R. B. Kearfott and A. Arazyan, "Taylor series models in deterministic global optimization", In: *Proc. 3rd Int. Conf. and Workshop on Automatic Differentiation*, June 2000.

[10] Q. Lin and J. G. Rokne, "Interval approximation of higher order to the ranges of functions", *Computers Math. Applic.*, 1996, Vol. 31, No. 7, pp. 101–109.

[11] K. Makino and M. Berz, "Remainder differential algebras and their applications", In: M. Berz, C. Bischof, G. Corliss, and A. Griewank (eds.), *Computational differentiation: techniques, applications, and tools*, SIAM, 1996, pp. 63–75.

[12] K. Makino and M. Berz, "Efficient control of the dependency problem based on Taylor model methods", *Reliable Computing*, 1999, Vol. 5, pp. 3–12.

[13] J. J. More, B. S. Garbow, and K. E. Hillstrom, "Testing unconstrained optimization software", *ACM Trans. Mathematical Software*, 1981, Vol. 7, No. 1, pp. 17–41.

[14] A. Neumaier, *Interval methods for systems of equations*, Cambridge University Press, Cambridge, England, 1990.

[15] H. Ratschek and J. Rokne, *Computer methods for the range of functions.* Ellis Horwood Ltd, Chichester, 1984.

[16] H. Ratschek and J. Rokne, *New computer methods for global optimization*, Wiley, New York, 1988.

Verified Estimation of Taylor Coefficients and Taylor Remainder Series of Analytic Functions

Markus Neher
Institut für Angewandte Mathematik, Universität Karlsruhe
D-76128 Karlsruhe, Germany
markus.neher@math.uni-karlsruhe.de

1 Introduction

Let $y = \sum_{j=0}^{\infty} b_j z^j$ be the analytic solution of a problem $F(y) = 0$ that depends on analytic functions $f_1, \ldots, f_n$ and that can be solved by recurrent computation of the Taylor coefficients b_j of y. Only a finite number of the b_j can be calculated in practical computations. Then a finite sum $\sum_{j=0}^{p} b_j z^j$ only yields an approximation to $y(z)$.

In some cases, however, it is possible to determine a geometric series (or some derivative of a geometric series) that serves as a bound for the remainder series of y, provided that the remainder series of the functions $f_1, \ldots, f_n$ can also be estimated by geometric series (or derivatives of geometric series). In [4], such an error analysis was used for the validated solution of linear ODEs.

A prerequisite of this general method is the computation of bounds for the Taylor coefficients of arbitrary order of a given analytic function f. The subject of this talk is the validated solution of the latter problem.

2 Estimates for Taylor Coefficients

In the following, let $f(z) = \sum_{j=0}^{\infty} a_j z^j$ be analytic in B and bounded on C, where B is the complex disc $\{z : |z| < r\}$ and C the circle $\{z : |z| = r\}$, for some $r > 0$. A well known bound for the Taylor coefficients of f is Cauchy's estimate $M(r)$:

$$|a_j| \leq \frac{M(r)}{r^j}, \quad M(r) := \max_{|z|=r} |f(z)|, \quad j \in \mathbb{N}_0.$$

Unfortunately, Cauchy's estimate is sometimes very pessimistic. To obtain better bounds, two modifications of Cauchy's estimate were proposed in [5]. The first uses a Taylor polynomial to approximate f:

Theorem 1 *Let f be analytic in B and bounded on C. Furthermore, let $T_l(z)$ denote the Taylor polynomial of order l to f. Then*

$$|a_j| \;\le\; \frac{N(r,l)}{r^j} \quad \textit{for } j > l, \quad \textit{where} \quad N(r,l) := \max_{|z|=r} |f(z) - T_l(z)| .$$

Cauchy's estimate can also be improved using the derivatives of f:

Theorem 2 *Let f be analytic in B and let $f^{(m)}$ (the m–th derivative of f) be bounded on C. Furthermore, let $P(j,m) := (j+1)\cdots(j+m)$, $P(j,0) := 1$ for $m \in \mathbb{N}$, $j \in \mathbb{N}_0$. Then*

$$|a_j| \;\le\; \frac{U(r,m)r^m}{P(j-m,m)r^j} \quad \textit{for } j \ge m, \quad \textit{where} \quad U(r,m) := \max_{|z|=r} |f^{(m)}(z)| .$$

3 Estimates for Taylor Remainder Series

The estimation of the remainder series $R_p := \sum_{j=p+1}^{\infty} a_j z^j$ is obtained by addition of the above estimates of the Taylor coefficients. Using Theorem 1, at some point z with $|z| = \omega r$, $\omega \in (0,1)$, for arbitrary $p \ge l$ we have

$$|R_p(z)| \;\le\; \sum_{j=p+1}^{\infty} N(r,l)\,\omega^j \;=\; N(r,l)\,\frac{\omega^{p+1}}{1-\omega},$$

whereas for $p \ge m-1$ we have

$$|R_p(z)| \;\le\; \sum_{j=p+1}^{\infty} \frac{U(r,m)\,r^m}{P(j-m,m)}\omega^j$$

by Theorem 2. For $p = m-1$ the latter sum is derived by repeated integration of $\frac{1}{1-\omega}$. For $p \gg m$ the resulting formula suffers from severe cancellation. In this case, the estimate

$$R_p \;\le\; \frac{U(r,m)\,r^m}{P(p+1-m,m)}\,\frac{\omega^{p+1}}{1-\omega}$$

is better suited for practical calculations.

In the situation that was mentioned in the introduction, it is usually not known in advance which order p of the Taylor polynomial T_p is required for sufficient accuracy of the approximation of the unknown solution y. Because it is very expensive to recalculate the bounds N or U for different values of l or m, respectively, it is better to guess sufficiently large values for l or m a priori and to calculate N or U only once. Only if $p = l$ or $p = m$ are not sufficient for a good approximation of y, then p is augmented iteratively, but R_p is still calculated for the same values of l or m (note that $R_p \to 0$ for $p \to \infty$ independently of l or m).

4 Implementation

Using interval arithmetic [1] on the computer, the validated computation of the above estimates is possible for analytic compositions of rational functions and of those complex standard functions that are available on the computer (like e^z, $\sin z$, $\mathrm{Log}\, z$, ...). The real and imaginary parts of many standard functions can be expressed as compositions of real standard functions. Such compositions have been utilized in [2] for the construction of complex interval standard functions that enclose the respective ranges over complex intervals. These inclusion functions and well known methods for rigorous global optimization [3] are used in the practical calculation of the estimates $M(r)$, $N(r,l)$, or $U(r,m)$.

5 ACETAF Software

The above algorithms for the computation of guaranteed upper bounds for the Taylor coefficients and remainder series of analytic functions have been implemented in a C–XSC program called ACETAF. The program also contains routines for the validated automatic computation of derivatives of complex analytic functions and for the check of analyticity of user–defined functions in circles in the complex plane.

ACETAF is distributed under the terms of the GNU General Public License and is available at the following site:

`http://www.uni-karlsruhe.de/~Markus.Neher/acetaf.html`

References

[1] G. Alefeld and J. Herzberger, *Introduction to interval computations*, Academic Press, New York, 1983.

[2] K. Braune and W. Krämer, "High-accuracy standard functions for real and complex intervals", In: E. Kaucher, U. Kulisch, and Ch. Ullrich (eds), *Computerarithmetic: Scientific computation and programming languages*, Teubner, Stuttgart, 1987, pp. 81–114.

[3] R. B. Kearfott, *Rigorous global search: Continuous problems*, Kluwer, Dordrecht, 1996.

[4] M. Neher, "Geometric series bounds for the local errors of Taylor methods for linear n-th order ODEs", In: G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto (eds), *Symbolic Algebraic Methods and Verification Methods*, Springer, Wien, 2001, pp. 183–193.

[5] M. Neher, "Validated bounds for Taylor coefficients of analytic functions", *Reliable Computing*, 2001, Vol. 7, pp. 307–319.

Interval Methods for Accelerated Global Search in the Microsoft Excel Solver

Ivo P. Nenov and Daniel H. Fylstra
Frontline Systems Inc.

This paper describes advanced interval methods for finding a verified global optimum and finding all solutions of a system of nonlinear equations, as implemented in the Premium Solver Platform, an extension of the Solver bundled with Microsoft Excel. It also describes the underlying tools that allow Excel spreadsheets to be evaluated over reals and intervals, with fast computation of real gradients and interval gradients. The advanced interval methods described include mean value (MV) and generalized interval (GI) representations for functions, constraint propagation for both the MV and GI forms, and a linear programming test for the GI form, in the context of an overall interval branch and bound algorithm. Numerical results for a set of sample problems demonstrate a significant speed advantage for the GI techniques, compared to alternative methods.

1 Introduction

The Solver bundled with Microsoft Excel, developed by Frontline Systems for Microsoft, is among the most widely used tools for optimization and equation solving. It is capable of solving small-scale linear programming (LP), smooth nonlinear programming (NLP), and mixed integer programming (MIP) problems. Included in nearly 100 million copies of Microsoft Excel, it offers Excel spreadsheet users an easy introduction to classical methods of optimization. An upgraded Premium Solver for Education, now bundled with more than a dozen textbooks, is used in a wide range of MBA and engineering courses.

The Premium Solver Platform is a compatible upgrade that extends the functionality, capacity and speed of the Microsoft Excel Solver to handle industrial-scale problems, including LP problems of over 100,000 variables and constraints; NLP problems with tens of thousands of variables and constraints; challenging mixed-integer problems; global optimization problems using multi-start or clustering methods; and non-smooth problems using methods based on genetic and evolutionary algorithms and tabu search.

In the past two years, we have sought to greatly extend the capabilities of

the Premium Solver Platform for verified global optimization and solution of systems of equations, using interval methods. Since Microsoft Excel is designed to evaluate spreadsheet models only over real numbers, we built a new parser and interpreter for Excel formulas that can evaluate models over several domains including intervals, as outlined below. We then implemented and tested a variety of interval-based techniques for global optimization and solution of systems of equations, including techniques implemented in other interval solvers and some new techniques implemented for the first time, to our knowledge, in the Premium Solver Platform. Numerical results suggest that these new techniques offer significant speed advantages over previously described methods.

2 Parser/Interpreter and Automatic Differentiation

Our parser and interpreter for Excel formulas take the place of the standard facility in Microsoft Excel for function evaluation or "recalculation." These tools can be used to evaluate Excel formulas over real and interval numbers; real gradients and interval gradients, using the techniques of automatic differentiation; and a special "diagnostic number type" that identifies sparsity in models to save memory, and classifies models, functions, and individual variable occurrences as linear, smooth nonlinear, or non-smooth.

Since Microsoft Excel supports a wide range of arithmetic (and non-arithmetic) operators and several hundred built-in functions, and we wished to evaluate all of these functions over several domains, we broke down the task by (i) defining a small number of basic functions, and implementing all other functions in terms of these, and (ii) making extensive use of operator overloading in the C++ programming language. Operator overloading allows us to define composite functions that can be evaluated over reals and intervals, real and interval gradients, G-intervals, and the "diagnostic number type."

We have implemented both forward and reverse mode automatic differentiation for both real gradients and interval gradients, using certain memory efficient techniques.

3 Interval Methods for Global Optimization and Equation Solving

We describe a framework for finding the global optimum of a constrained optimization problem, and finding all solutions of a system of equations. We begin with the basic interval branch and bound algorithm, which processes a list of "boxes" and seeks to reduce their size, splitting them as necessary, until a list of sufficiently small boxes enclosing the solution(s) is obtained.

We *represent the function* to be minimized and the equality and inequality constraints using the natural interval extension form, the mean value (MV) form (a quadratic outer approximation), and the generalized interval (GI) or linear enclosure form. The MV and GI forms are effectively first-order methods that require more computation than the natural interval extension, but they permit more rapid reduction of box sizes and elimination of boxes in the overall branch and bound algorithm.

We then describe *constraint propagation techniques* as a way of speeding up the process of box size reduction. These techniques can be applied to single equations (or equality constraints) at a time, and hence require less memory and computation than the general MV and GI box reduction techniques.

Finally, we describe a *linear programming test* that can be applied to the GI form to rapidly eliminate a box where the (dual) Simplex method finds no feasible solutions for the linear enclosures of the equality constraints.

4 Numerical Results and Conclusions

We present numerical results for a few test problems, including an electronic circuit analysis problem and a problem used by van Hentenryck to illustrate performance of the Numerica system. We report CPU time and the number of function evaluations, gradient evaluations, and box splits for three interval method implementations: using MV form techniques, using GI form techniques, and adding a LP test to the second method. The GI form techniques including the LP test dramatically outperform the natural interval extension and the MV form techniques in all reported cases, with several-fold reductions in CPU time, function evaluations, and gradient evaluations.

We conclude that the interval methods presented for global optimization and equation solving show promise for practical problems, especially where a verified rather than an approximate or merely "good enough" solution is desired, and that the availability of these methods in an easy-to-use, highly accessible form in Microsoft Excel should result in more widespread use and appreciation for the power of these methods.

Solving Real-Life Robotics Problems with Interval Techniques

Arnold Neumaier and Jean-Pierre Merlet

The design, validation, and real-time use of robots poses a number of challenging global constraint satisfaction and/or optimization problems in dimensions ranging from a few to several hundreds, with quadratic, polynomial, or transcendental constraints. The talk will discuss background, formulation, and solution for some of these problems.

Detecting and Locating Curved Cracks in Thin Plates by Lamb Wave Reflection: Validated Geometric Approach

Roberto Osegueda and Vladik Kreinovich
FAST Center for Structural Integrity of Aerospace Structures
University of Texas, El Paso, TX 79968, vladik@cs.utep.edu

The practical problem. In many practical situations, it is extremely important to be able to detect hidden cracks (and other possible faults) in aerospace and other structures. One way of testing the structural integrity of such structures is by using ultrasonic waves. For thin plates, Lamb waves (which direct the energy along the plate) are especially useful. To use these waves, we set up a transmitter T and (one or several) receivers R, and try to extract, from the signals measured by the receivers, the information about the possible cracks and other faults.

This extraction is not easy.

Lamb waves are difficult to control. In many practical problems – e.g., in radar detection – we detect objects by sending waves and processing the signals measured by receivers. In most such situations, waves travel by air, so we can easily control them: we can use an antenna to reflect the generated waves and thus, to focus the waves in the desired direction; similarly, we can use an antenna to gather waves coming from different directions into a single point and thus, amplify the received signal.

In contrast, Lamb waves travel *inside* the plate. There is no easy way to reflect them without placing reflectors inside the plate – i.e., without introducing additional faults.

Reconstructing the fault location and shape from the Lamb wave measurements is a computationally difficult problem. Propagation of Lamb waves is described by known PDEs. In general, there exist efficient computational techniques for solving PDEs. However, these techniques cannot be easily applied to the problem of fault location: Indeed, these techniques usually assume that we know the boundary conditions, but in the fault location problem, the location of the boundary is exactly the problem.

As a result, reconstructing the location and the shape of a crack from the measurement results is a very difficult task. Even for simple faults like edge

cracks near the rivet holes, the corresponding methods have been developed only recently [1]. The existing methods enable us, at best, to predict, for a given size and orientation of the crack, what the signal will be. Since the corresponding formulas are very complex, the only way to detect the location and size from the measurement results is to compare these results with theoretical predictions corresponding to different crack sizes and locations. This is not practical. We need a method which will transform the known signal into the information about the location and size of the crack, and the complexity of the existing formulas prevents us from doing it.

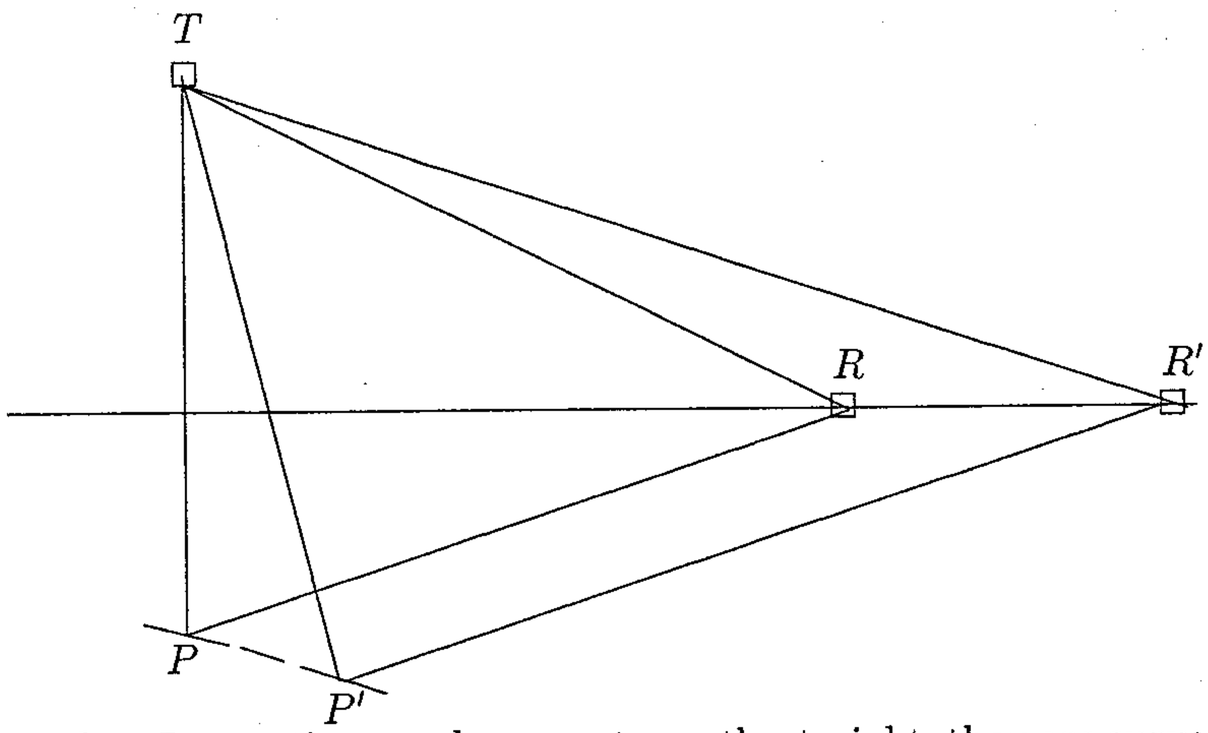
Geometric approach. One way to avoid this complexity problem is to use a *geometric* approach, when instead of trying to fully capture the complex physics of Lamb waves, we only use general geometry-related properties of wave propagation.

What was known before: detection of linear cracks. In [2], the geometric approach has been successfully used to detect linear cracks. Detection of such cracks is based on the following idea. In an ideal (faultless) plate, a Lamb wave goes directly from the transmitter to the receiver. The signal emitted by the transmitter at time t reaches the receiver at time $t + t_0$, where $t_0 = d_0/v$, $d_0 = \overline{TR}$ is the distance between the transmitter T and the receiver R, and v is the wave speed. The signal detected by the receiver has the same shape as the signal sent by the transmitter, but shifted in time by t_0. In particular, if the transmitter emitted a short pulse train, the signal registered by the receiver will consist of the same short pulse train – occurring at a later moment of time.

What happens when the plate contains a crack? A crack reflects the wave. As a result, if a transmitter sends a pulse train at a certain moment of time t, this pulse train first reaches the receiver directly, at the moment $t + t_0$, and then another copy of this pulse train reaches the receiver indirectly, after first hitting the crack and then being reflected by the crack. So, if the plate contains a crack, the receiver will receive the signal consisting of *two* pulse trains: the earlier pulse train at time $t + t_0$ which comes directly from the transmitter, and a later pulse train coming at a time $t + t_1 = t + d_1/v$, where $d_1 = \overline{TP} + \overline{PR}$ is the total path of the reflected signal (P is the point on the crack where the signal was reflected). By measuring the time t_1 between the emitted pulse train and the second detected pulse train, we can thus determine the total path D of the reflected signal as $d_1 = v \cdot t_1$.

This information can be used to locate the crack. Indeed, for the (unknown) reflection point P, we know the sum $\overline{TP} + \overline{PR}$ of the distances from two known points: T and R. It is a known geometrical fact that for any given two points T and R, the set of all points P with a given sum $\overline{TP} + \overline{PR}$ is an ellipse. Due to Snell's law describing wave reflection, the angle between the incoming wave and the crack must be the same as between the crack and the outcoming wave. Due to the properties of an ellipse, we can conclude that the crack is tangent to this ellipse at the reflection point P. The paper [2] shows how we can use this

fact to compute the coordinates of the straight line crack.



Our main results. In practice, cracks are not exactly straight, they are curved. There is no known way to describe a shape of a realistic (curved) crack by a finite-parametric formula :-(We therefore need techniques for detecting and locating generic cracks. Such geometric techniques are proposed in this paper.

We also provide guaranteed bounds for the crack location. The inaccuracy of the fault location is caused by the inaccuracy of measuring the arrival time. The main component of the time measurement error is the discretization error. For this error, we know its upper bound Δt. Thus, when the measured time value is $\widetilde{t} = k \cdot \Delta t$, the only information about the actual time t is that t must belong to the interval $[\widetilde{t} - \Delta t, \widetilde{t} + \Delta t]$.

The exact values of t would lead to the exact location of the point in a fault. Since the input data has interval uncertainty, we cannot get the exact location; instead, we use interval computations to describe the area that is guaranteed to contain a point from the fault.

Preliminary results show that this method indeed works well on actual data.

Acknowledgments. This work was supported by AFOSR grant F49620-00-1-0365. We are thankful to Ajit Mal and to the anonymous referees for useful discussions.

References.

[1] Z. Chang and A. Mal, "Scattering of Lamb waves from a rivet hole with edge cracks", *Mechanics of Materials*, 1999, Vol. 31, pp. 197–204.

[2] F. De Villa, E. Roldan, R. Mares, S. Nazarian, and R. Osegueda, "Defect detection in this plates using S_0 Lamb wave scanning", *Proc. SPIE Conf. on NDE for Health Monitoring & Diagnostics*, Newport Beach, CA, March 2001.

Adaptive Numerical Methods for Sensitivity Analysis of Differential-Algebraic Equations and Partial Differential Equations

Linda Petzold
petzold@engineering.ucsb.edu

Sensitivity analysis of differential-algebraic equation (DAE) systems generates essential information for design optimization, parameter estimation, optimal control, model reduction, process sensitivity and experimental design. Recent work on methods and software for sensitivity analysis of DAE systems has demonstrated that forward sensitivities can be computed reliably and efficiently. However, for problems which require the sensitivities with respect to a large number of parameters, the forward sensitivity approach is intractable and the adjoint (reverse) method is advantageous. In this talk we give the adjoint system for general DAEs and investigate some of its fundamental analytical and numerical properties. We describe our new adjoint DAE software and outline some issues which are critical to the implementation.

Defining the adjoint sensitivity system and writing the appropriate software to describe it can be a very challenging problem for large-scale engineering systems, particularly when it comes to finding appropriate boundary conditions for the adjoint partial differential equation (PDE) system. Therefore our goal for both DAE and PDE systems has been the development of methods and software in which generation and solution of the sensitivity system are transparent to the user. This has been largely achieved for DAE systems. We will propose a solution to this problem for PDE systems solved with adaptive mesh refinement.

Evaluation of Functions, Gradients and Jacobians

Louis B. Rall
Department of Mathematics
University of Wisconsin-Madison
Madison, Wisconsin 53706, U.S.A.
rall@math.wisc.edu

A function $f : D \subset \mathbf{R}^d \to \mathbf{R}$ is ordinarily evaluated by constructing a sequence $\{x_1, x_2, \ldots, x_n\}$, where each x_k is obtained by evaluating an expression of the form $x_k = f_k(x_1, \ldots, x_{k-1})$, and $x_n = f(x_1, \ldots, x_d)$. In realization of such algorithms by computer programs, the expressions f_k can be limited to assignments, arithmetic operations, and functions already programmed or built into the hardware, such as those belonging to a standard library of mathematical functions. The number of steps n of the algorithm to evaluate the function may depend on the given values of $x_1, \ldots, x_d$, but this dependence will be suppressed for simplicity of notation.

Instead of considering the evaluation process as a mapping from a point in $\mathbf{R}^d$ to a point in $\mathbf{R}$, it will be viewed as a transformation in $\mathbf{R}^n$ of the form $\mathbf{x} = \mathbf{F}(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, that is, as a fixed point problem in $\mathbf{R}^n$ or the equivalent equation $\mathbf{G}(\mathbf{x}) \equiv \mathbf{x} - \mathbf{F}(\mathbf{x}) = 0$. These formulations give the possibility of improvement of accuracy of the function evaluation and validation of the computed result.

In particular, if the expressions f_k are differentiable, then the Jacobian $J = \mathbf{F}'(\mathbf{x}) = (\partial x_i / \partial x_j)$ exists and is strictly lower triangular. One has $\mathbf{G}'(\mathbf{x}) = I - J$ and $\mathbf{G}'(\mathbf{x})^{-1} = (I - J)^{-1} = I + J + \cdots + J^m$ for some $m \leq n$. Thus, Newton's method can be applied to $\mathbf{G}(\mathbf{x}) = 0$ for improvement of computed values and their validation by interval inclusion.

Another use of the matrix J is the computation of the gradient ∇f, a process commonly referred to as automatic differentiation. The forward mode consists of computing the right eigenvectors of J by the power method, the reverse mode yields a left eigenvector, also by the power method. In either case, accurate matrix-vector multiplication with the aid of a long accumulator and interval validation of results are applicable.

The above results apply immediately to evaluation of functions $f : D \subset \mathbf{R}^p \to \mathbf{R}^q$ to obtain the corresponding Jacobian of the transformation. In the case $p = q$, the function being computed may be an inverse function, that is, one uses the computer to solve the equation $f(x) = y$ for $x = f^{-1}(y) = g(y)$. Since the program will contain a subroutine for $f(x)$, it is not necessary of find the Jacobian of $g(y)$ by differentiation of the entire routine. Once a satisfactory

value of x has been computed, one has $g'(y) = [f'(x)]^{-1}$, so only the subroutine for $f(x)$ has to be differentiated. In this sense, differentiation and inversion commute because of linearity.

Motivations for an Arbitrary Precision Interval Arithmetic and the MPFI Library

N. Revol[1] and F. Rouillier[2]

[1]Lab. ANO, University of Lille and
and CNRS/ENSL/INRIA Project Arenaire
LIP, École Normale Supérieure de Lyon, France
[2]Project Spaces, LORIA/INRIA/LIP 6, France

1 Motivations for Changing Arithmetic

Nowadays, computations involve more and more operations and consequently errors. The limits of applicability of some numerical algorithms are now reached: for instance the theoretical stability of a dense matrix factorization (LU or QR) is ensured under the assumption that $n^3u < 1$, where n is the dimension of the matrix and $u = 1^+ - 1$, with 1^+ the smallest floating-point larger than 1; this means that n must be less than 200,000, which is almost reached by modern simulations. The numerical quality of solvers is now an issue, and not only their mathematical quality. Let us cite studies performed by the CEA (French Nuclear Agency) on the simulation of nuclear plant accidents and also softwares controlling and possibly correcting numerical programs, such as Cadna [10] or Cena [20].

Another approach consists in computing with certified enclosures, namely interval arithmetic [21, 2, 18]. The fundamental principle of this arithmetic consists in replacing every number by an interval enclosing it. For instance, π cannot be exactly represented using a binary or decimal arithmetic, but it is certified that π belongs to $[3.14159, 3.14160]$. The advantages of interval arithmetic are numerous. On the one hand, it exhibits the property of *validated* or *certified computing*. On the other hand, computer implementations are based on outward roundings and thus computed results take into account rounding errors and constitute a way to estimate these errors. A last and very important advantage, even if it is often less known, is that this arithmetic provides global information: for instance, it provides the range of a function over a whole set S, which is crucial for global optimization; furthermore, if this range is a (strict) subset of S, then Brouwer's theorem states that this function has a (unique) fixed-point and this can be used by Newton's algorithm for instance. Such properties cannot be reached without set computing, and interval arithmetic computes with sets and is easily available.

However, in spite of the improvements in interval analysis, the problem of overestimation, *i.e.* of enclosures which are far too large and thus inaccurate, seems to be the destiny of interval computation when it is implemented using fixed-precision floating-point arithmetic. Using a multiple precision postpones the occurrence of numerical problems, however the number of correct figures remains unknown. Computing with intervals provides guaranteed results, but the bounds can be far apart even when the input data are provided with the machine precision; a remedy for this phenomenon consists in computing with a higher precision. This proposal is the core of the MPFI library (*Multiple Precision Floating-point Interval arithmetic library*), a library implementing arbitrary precision interval arithmetic which is described in this paper.

This quest for extra accuracy can be found in other works such as those by [9] where polynomial expressions are symbolically rewritten before being evaluated, so as to reduce the overestimation due to dependency, or by [5] where high-order Taylor expansions are used. In this latter work, the time overhead is about 1500 for a single evaluation, however it is compensated by the reduction in the number of steps performed by the algorithm. Real-world applications where extra accuracy is required are to be found in automatics (we have been asked to integrate linear systems with high accuracy) or chemistry: determining a molecular conformation entails the minimization of an energy function and requires accurate evaluations of this energy function.

Several multiple precision interval packages are available. Let us quote for instance intpak [11] and intpakX [13] for Maple or a similar package for Mathematica [17]. Due to unverified assumptions on the roundings of elementary functions (0.6 ulp for intpak in Maple, 1 ulp for Mathematica), to bugged roundings (for instance, with 3 decimal digits, the rounding towards $-\infty$ of $1-9.10^{-5}$ gives 1 instead of 0.999 in Maple v6 and v7), and to several undue assumptions, these packages cannot be considered as reliable. Earlier works include the "range arithmetic" [1], a multiple precision library which aims at indicating the number of correct digits rather than at performing interval arithmetic, and IntLab [27] which primarily implements efficiently interval algorithms using MatLab, and, besides, mainly provides a type for arbitrary precision computations but implements few related functionalities. Such an arithmetic was also mentioned as an easy-to-implement extension to Brent's multiple precision package MP as early as 1981 [7]. Anyway, none of the aforementioned packages implements a complete and really reliable arbitrary precision interval arithmetic and this led us to implement our own library.

2 Theoretical Background

The theoretical result underlying this idea can be found in [23]: let us denote by X an interval, by f a function and by F an interval extension of f, where F is given by a Lipschitz expression, let ε correspond to the current computing precision p: $\varepsilon = 2^{-p}$, then $F(X)$ overestimates $f(X)$ and the overestimation is

bounded by

$$q(f(X), F(X)) \leq c_1 w(X) + c_2 \varepsilon \tag{1}$$

where q is the Hausdorff distance, $w(X)$ is the width of X and the constants c_1 and c_2 depend on F. This means that the computing precision can become a limiting factor and that being able to increase it can be an issue.

Furthermore, a classical procedure in interval analysis is the bisection one: if the output width is too large, then the inputs are split in two (or more) parts and the computation is repeated on each part. Bisection is a way to escape the wrapping effect by providing a paving of the sought set, and also the dependency problem, even if, in that respect, nothing supersedes the use of a good formulation. Bisection is often the last resort to get more accuracy, by reducing in formula (1) the quantity $w(X)$. In cases where $w(X) = u$ (which happened in our experiments on global optimization), only an increase in the computing accuracy, by "adding new floating-point numbers" between the endpoints of X, would have yielded a solution.

It can be noticed that the rule of thumb "*to get more digits, one has to increase the computing precision by roughly the same number of digits*" can fail, for instance when computing a square root or more generally a $1/n$-th power close to 0. However, the rule of thumb becomes in such cases "*to get α more digits, one has to increase the computing precision by roughly $n\alpha$ digits*". In other words, in most cases an increase in the computing precision yields an improved accuracy on the results.

This is also the starting point of Müller's work on an effective simulation of a Real RAM [22], following the theoretical results by Brattka and Hertling [6] on the feasibility of a Real RAM. In Müller's work, a computation is performed and, if the final accuracy is not sufficient, then the whole computation is restarted with an increased precision; this is reiterated until the outputs are accurate enough.

3 The MPFI Library

In order to implement an arbitrary precision interval arithmetic, a multiple precision library was needed. By multiple precision, it is meant that the computing precision is not limited to the single or double precision of machine floating-point numbers; on the contrary, arbitrary precisions should be available. Furthermore, this computing precision must be dynamically adjustable to fulfill the accuracy needs. A more precise requirement for interval arithmetic is the outward rounding facility: this ensures that for each operation, the interval computed using floating-point arithmetic contains the interval obtained if exact real arithmetic were used. Even more desirable is *exact* directed rounding to avoid losing accuracy, *i.e.* the interval computed using floating-point arithmetic is the smallest one (for inclusion); however, it is rarely fulfilled for elementary functions. To sum up, compliance with the IEEE 754 standard for floating-point arithmetic, extended to elementary functions, is welcome.

The Arithmos project of the CANT team, U. Antwerpen, Belgium [8], or the MPFR library (*Multiple Precision Floating-point Reliable library*), developed by the Spaces team, INRIA Lorraine, France [12], are such libraries. For portability and efficiency reasons (MPFR is based on GMP and efficiency is a motto for its developers) and also because of the availability of the source code, we chose MPFR. The corresponding library, named MPFI [25], is a portable library written in C for arbitrary precision interval arithmetic. It is based on the GNU MP library and on the MPFR library and is part of the latter. The largest achievable computing precision is provided by MPFR and depends in practice on the computer memory on which it runs. The only theoretical limitation, which may be removed soon, is that the exponent must fit in an integer. Let us just say that it is possible to compute with numbers of several millions of binary digits if needed.

Intervals are implemented using their endpoints, which are MPFR reliable floating-point numbers: this is not visible for the user but ensures that the swelling of intervals' widths is less important than with the midpoint-radius representation such as implemented by Rump in IntLab [27, 28]. Indeed, switching the rounding modes incurs no penalty with multiple precision arithmetic and the motivation for this choice in IntLab does not hold for MPFI: every multiple precision operation is a software one. The arithmetic operations are implemented and the elementary functions available up to now are exp, log, sine and cosine; all functions provided by MPFR will shortly be included as well (trigonometric and hyperbolic trigonometric functions and their reciprocals).

The planned functionalities, that will be added in a near future, include a C++ interface à la Profil/BIAS [19] for ease of use, basic tools for linear algebra (vector and matrix data types, additions and multiplications) and automatic differentiation (forward differentiation by overloading operators and functions).

4 Applications

The MPFI library is already in use. Rouillier and Zimmermann [26] have developed a hybrid algorithm (symbolic/interval) for isolating real roots of polynomials, Revol [24] has implemented interval Newton algorithm [15] adapted to multiple precision computations. A main advantage of using MPFI is that one is no more limited by the computing precision: for instance one can impose arbitrary accuracy on both the root and the residual in Newton's algorithm [4]. Furthermore, the aforementioned implementations manage to adapt dynamically the precision to the computing needs without restarting the whole program. This desirable feature will be sought after for future implementations of other algorithms.

5 Conclusion

MPFI is a library for multiple precision interval arithmetic. It is written in C and built upon MPFR and GMP and can be freely downloaded. It is still under development: new facilities such as automatic differentiation and linear algebra will be added in the near future. It still has enabled us to implement and test some algorithms and this will be pursued with a careful study of the solution of linear systems and of global optimization of continuous functions [14, 3]. Applications such as parameter estimation in automatics [16] will offer the opportunity to gain further insight in the development of new algorithms.

References

[1] O. Aberth and M. J. Schaefer, "Precise computation using range arithmetic, via C++", *ACM TOMS*, 1992, Vol. 18, No. 4, pp. 481–491.

[2] G. Alefeld and J. Herzberger, *Introduction to interval analysis*, Academic Press, 1983.

[3] R. Baker Kearfott, *Rigorous global search: continuous problems*, Kluwer, 1996.

[4] R. B. Kearfott and G. W. Walster, "On stopping criteria in verified nonlinear systems or optimization algorithms", *ACM TOMS*, 2000, Vol. 26, No. 3, pp. 373–389.

[5] M. Berz and J. Hoefkens, "Verified high-order inversion of functional dependencies and interval Newton methods", *Reliable Computing*, 2001, Vol. 7, pp. 1–20.

[6] V. Brattka and P. Hertling, "Feasible real random access machines", *J. of Complexity*, 1998, Vol. 14, No. 4, pp. 490–526.

[7] R. P. Brent, "A Fortran multiple-precision arithmetic package", *ACM TOMS*, March 1978, Vol. 4, pp. 57–70.

[8] CANT Research Group, *Arithmos: a reliable integrated computational environment*, University of Antwerpen, Belgium, `http://win-www.uia.ac.be/u/cant/arithmos`, 2001.

[9] M. Ceberio and L. Granvilliers, "Solving Nonlinear Systems by Constraint Inversion and Interval Arithmetic", In *Int. Conf. on Artificial Intelligence and Symbolic Computation (AISC'2000)*, LNAI 1930.

[10] J.-M. Chesneaux, S. Guilain, and J. Vignes, *La bibliothèque CADNA: présentation et utilisation.* `http://www-anp.lip6.fr/cadna`, 1996.

[11] A. E. Connell and R. M. Corless, "An experimental interval arithmetic package in Maple", In *Num. Analysis with Automatic Result Verification*, 1993.

[12] D. Daney, G. Hanrot, V. Lefèvre, F. Rouillier, and P. Zimmermann, *The MPFR library*, `http://www.mpfr.org`, 2001.

[13] I. Geulig and W. Krämer, *Intervallrechnung in Maple - Die Erweiterung intpakX zum Paket intpak der Share-Library*, Technical Report 99/2, Universität Karlsruhe, 1999.

[14] E. Hansen, *Global optimization using interval analysis*, Marcel Dekker, 1992.

[15] E. Hansen and R.I. Greenberg, "An interval Newton method", *J. of Applied Math. and Computing*, 1983, Vol. 12, pp. 89–98.

[16] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis*, Springer Verlag, 2001.

[17] J. Keiper, "Interval arithmetic in Mathematica", *Interval Computations*, 1993, No. 3.

[18] R. Klatte, U. Kulisch, C. Lawo, M. Rauch, and A. Wiethoff, *C-XSC: A C++ class library for extended scientific computing*, Springer Verlag, 1993.

[19] O. Knueppel, "PROFIL/BIAS - a fast interval library", *Computing*, 1994, Vol. 53, No. 3-4, pp. 277–287.

[20] P. Langlois, "Automatic linear correction of rounding errors", *BIT Numerical Algorithms*, 2001, Vol. 41, No. 3, pp. 515–539.

[21] R.E. Moore, *Interval analysis*, Prentice Hall, 1966.

[22] N. Müller, "The iRRAM: Exact Arithmetic in C++", In *Proc. Workshop on Constructivity and Complexity in Analysis*, Swansea, 2000.

[23] A. Neumaier, *Interval methods for systems of equations*, Cambridge University Press, 1990.

[24] N. Revol, *Newton's algorithm using multiple precision interval arithmetic*, Research Report 4334, INRIA, 2001; submitted to *Numerical Algorithms*.

[25] N. Revol and F. Rouillier, *The MPFI library* `http://www.ens-lyon.fr/~nrevol`, 2001.

[26] F. Rouillier and P. Zimmermann, *Efficient isolation of polynomial real roots*, 2001.

[27] S. Rump, "INTLAB - Interval Laboratory", In: T. Csendes (ed.), *Developments in Reliable Computing*, Kluwer, 1999, pp. 77–104.

[28] S. Rump, "Fast and parallel interval arithmetic", *BIT*, 1999, Vol. 39, No. 3, pp. 534–554.

Linear Interval Equations: The Role of Preconditioning

Jiri Rohn
Charles University and Academy of Science
Prague, Czech Republic
rohn@cs.cas.cz

It is well known that the exact interval hull $[\underline{x}, \overline{x}]$ of the solution set of a system of linear interval equations

$$\mathbf{A}x = \mathbf{b} \tag{1}$$

is NP-hard to compute [1]. On the other hand, the interval hull $[\underline{\underline{x}}, \overline{\overline{x}}]$ for the preconditioned system

$$A_c^{-1}\mathbf{A}x = A_c^{-1}\mathbf{b}$$

(where A_c is the midpoint of $\mathbf{A}$) can be computed in polynomial time with only two matrices to be inverted [2, 3]. Since $[\underline{x}, \overline{x}] \subseteq [\underline{\underline{x}}, \overline{\overline{x}}]$, we obtain an enclosure $[\underline{\underline{x}}, \overline{\overline{x}}]$ of the solution set of (1) whenever the procedure is applicable (which is known to be the case if and only if $\mathbf{A}$ is strongly regular).

Nevertheless, the main question has remained unanswered so far: how well does $[\underline{\underline{x}}, \overline{\overline{x}}]$ approximate the exact hull $[\underline{x}, \overline{x}]$ of (1)? In this talk we shall present explicit formulae for nonnegative vectors $\underline{d}$ and $\overline{d}$ computable in polynomial time such that

$$\underline{\underline{x}} \leq \underline{x} \leq \underline{\underline{x}} + \underline{d},$$

$$\overline{\overline{x}} - \underline{d} \leq \overline{x} \leq \overline{\overline{x}}$$

hold. The formulae for $\underline{d}$ and $\overline{d}$, obtained in a rather sophisticated way, aside from their computational value also allow to draw several theoretical consequences.

References

[1] J. Rohn and V. Kreinovich, "Computing exact componentwise bounds on solutions of linear systems with interval data is NP-hard", *SIAM Journal on Matrix Analysis and Applications*, 1995, Vol. 16, pp. 415–420.

[2] E. R. Hansen, "Bounding the solution of interval linear equations", *SIAM Journal on Numerical Analysis*, 1992, Vol. 29, pp. 1493–1503.

[3] J. Rohn, "Cheap and tight bounds: The recent result by E. Hansen can be made more efficient", *Interval Computations*, 1993, No. 4, pp. 13–21.

Verification Methods for the Linear Complementarity Problem with Interval Data

Uwe Schäfer
Institut für Angewandte Mathematik
Universität Karlsruhe
D-76128 Karlsruhe, Germany

Let $M \in \mathbf{R}^{n\times n}$ and $q \in \mathbf{R}^n$. Then the linear complementarity problem is defined as follows: Determine (or conclude that there is no) $z \in \mathbf{R}^n$ with

$$q + Mz \geq 0, \quad z \geq 0, \quad (q + Mz)^T z = 0. \tag{1}$$

(Here, matrix inequalities are understood componentwise, [2].) Linear complementarity problems model many important mathematical problems. The article [3] gave an extensive documentation of complementarity problems in engineering and equilibrium modeling.

Meanwhile, verification methods have been found to give guaranteed bounds on the distance between the numerical solution and the exact solution of the linear complementarity problem (see e.g. [1]).

In this talk we extend this idea to the case where the input data itself are not known exactly but can only be enclosed in intervals. This situation arises for example from the following application.

Let $f : [0,\infty) \times \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ be a function and $y_0 > 0$. Then the free boundary problem is defined as follows:

$$\left.\begin{array}{rll} \text{Find } c \in \mathbf{R} & \text{and} & y(\cdot) : [0,\infty) \to \mathbf{R} \text{ with} \\ y''(x) = f(x, y(x), y'(x)) & & \text{if } x \in [0, c], \\ y(x) > 0 & & \text{if } x \in [0, c), \\ y(x) = 0 & & \text{if } x \in [c, \infty), \\ y'(c) = 0, & & y(0) = y_0. \end{array}\right\} \tag{2}$$

Theorem 1. ([4]) *The free boundary problem (2) is considered. It is assumed that (2) has a unique solution* $\left(\tilde{c}, \tilde{y}(\cdot)\right)$ *and it is assumed that an* $a \in \mathbf{R}$ *with* $\tilde{c} \leq a$ *is known.* $n + 2$ *points are determined by* $x_0 := 0$, $h := a/(n + 1)$, $x_{i+1} := x_i + h$, $i = 0, ..., n$. *Let* f *fulfill the following conditions:*

- $f(x, s, t) : [0,\infty) \times \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ *is continuously differentiable.*

- *There exists an interval* $[F] = [\underline{F}, \overline{F}]$ *with* $\{\mu \in \mathbf{R} : \mu = f(x, \tilde{y}(x), \tilde{y}'(x)),\ x \in [0, a]\} \subseteq [F]$ *and* $\underline{F} \geq 0$.
- *There exists* $D \in \mathbf{R}$ *with* $|f_x(x, \tilde{y}(x), \tilde{y}'(x)) + f_s(x, \tilde{y}(x), \tilde{y}'(x))\tilde{y}'(x) + f_t(x, \tilde{y}(x), \tilde{y}'(x))\tilde{y}''(x)| \leq D$, $x \in [0, \tilde{c}]$.

Then there exists a vector $q \in \mathbf{R}^n$ *contained in the interval vector*

$$[q] = \frac{1}{2}\begin{pmatrix} \left[\frac{1}{2}, 1\right] h^2[F] + \frac{1}{2}h^3[-D, D] - y_0 \\ \left[\frac{1}{2}, 1\right] h^2[F] + \frac{1}{2}h^3[-D, D] \\ \vdots \\ \left[\frac{1}{2}, 1\right] h^2[F] + \frac{1}{2}h^3[-D, D] \end{pmatrix}$$

and the vector

$$\tilde{y} := \begin{pmatrix} \tilde{y}(x_1) \\ \vdots \\ \tilde{y}(x_n) \end{pmatrix}$$

is the unique solution of the linear complementarity problem defined by that q *and*

$$M = \begin{pmatrix} 1 & -\frac{1}{2} & 0 & \cdots & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & \cdots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & \cdots & 0 & -\frac{1}{2} & 1 \end{pmatrix} \in \mathbf{R}^{n \times n}.$$

The proof uses Taylor's formula with remainder where one has to take into account that it is not necessary that $\tilde{y}(x) : [0, \infty) \to \mathbf{R}$ is twice differentiable at $x = \tilde{c}$. In addition, we want to emphasize that it is not possible to prove Theorem 1 with $M \in \mathbf{R}^{n \times n}$ and $[q] = q \in \mathbf{R}^n$ (even if $f \equiv 1$).

In order to verify that a given interval vector $[z]$ includes $\tilde{y}$ of Theorem 1 we have to consider a family of linear complementarity problems. The presented verification methods are based on the following equivalence

$$(1) \Leftrightarrow H(z) := \min(z, q + Mz) = 0$$

and the knowledge of a slope matrix $G(x, y)$ satisfying

$$H(x) - H(y) = G(x, y)(x - y) \quad \text{for all } x, y \in [z].$$

For the matrix M and the interval vector $[q]$ of Theorem 1 we define $H(z; [q])$ componentwise by

$$(H(z; [q]))_i = \left\{ \begin{array}{ccc} z_i & \text{if} & z_i < \underline{q_i} + (Mz)_i, \\ [q_i] + (Mz)_i & \text{if} & z_i > \overline{q_i} + (Mz)_i, \\ [\underline{q_i} + (Mz)_i, z_i] & \text{if} & z_i \in [q_i] + (Mz)_i, \end{array} \right\} i = 1, .., n,$$

and we present an algorithm that calculates an interval matrix $G(x,[z],[q])$ with arbitrary $x \in [z]$ satisfying

$$G(x,y) \in G(x,[z],[q]) \text{ for all } y \in [z] \text{ and for all } q \in [q].$$

Then we define the operator

$$N(x,[z],[q]) := x - IGA(G(x,[z],[q]), H(x;[q]))$$

(IGA means interval Gaussian algorithm) and the operator

$$L(x,A,[z],[q]) := x - A \cdot H(x;[q]) + \Big(I - A \cdot G(x,[z],[q])\Big)([z]-x)$$

with arbitrary $x \in [z]$ and an arbitrary nonsingular matrix A. Using the Brouwer fixed point theorem we include $\tilde{y}$ of Theorem 1. We present some examples.

References

[1] G. Alefeld, X. Chen, and F. Potra, "Numerical validation of solutions of linear complementarity problems", *Numer. Math.*, 1999, Vol. 83, pp. 1–23.

[2] R. W. Cottle, J. S. Pang, and R. E. Stone, *The linear complementarity problem*, Academic Press, 1992.

[3] M. C. Ferris and J. S. Pang, "Engineering and economic applications of complementarity problems", *SIAM Rev.*, 1997, Vol. 39, No. 4, pp. 669–713.

[4] U. Schäfer, "An enclosure method for free boundary problems based on a linear complementarity problem with interval data", *Numerical Functional Analysis and Optimization*, 2001, Vol. 22, No. 7-8, pp. 991–1011.

Bounding the Composite Value at Risk for Energy Service Company Operation with DEnv, an Interval-Based Algorithm

Gerald B. Sheblé and Daniel Berleant
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011

1 Summary

Deregulation in the power industry drives competition. It also increases the risk of doing business. Therefore it is important to manage and assess the risk. Value at risk (VaR) analysis has been used in financial institutions to evaluate portfolios of assets for some time, but the application of the approach in the power industry has not been established. The VaR of serving customer demand using the energy purchased on the auction market is our focus. In this paper, the risks of the energy service company (ESCO) are identified and the contract specifications and the VaR reviewed. In describing the difference in the business environments between the power and financial industries, the VaR analysis that has been used in the financial industry has been remodeled to best describe the assumed deregulated power environment. The pros and cons of the VaR levels are presented. As a consequence of the the interval-based computational core of DEnv (Distribution Envelope Determination), results are validated with respect to two sources of potential error.

1. Given the cumulative distributions of random variables, a derived random variable which is an arithmetic combination of the given random variables will have a single defined cumulative distribution only if the joint distribution of the given distributions is fully defined. If the joint distribution is not defined, a verified characterization of the result will be envelopes bounding the space of cumulative distribution curves that correspond to the members of the set of all the possible joint distributions. Distribution Envelope Determination (DEnv) [1] provides those envelopes, so that uncertainty in results due to uncertainty about dependencies among model variables is bounded.

2. The distributions of input random variables can be discretized in DEnv in in order to avoid the problem of finding envelopes for arbitrary input distri-

butions analytically. Discretization typically involves approximation, but DEnv can avoid this by bounding each input distribution with envelopes such that the discretized form of an input is a pair of envelopes enclosing it. (While the input distribution is likely to be a continuous curve, the envelopes are staircase-shaped.) This representation for the input curves propagates into wider envelopes around the space of possible result curves, because those envelopes bound the space of results not only with respect to different dependency relationships between the inputs (as described in the previous item), but also with respect to the space of curves consistent with the envelopes around an input.

Results are valuable because insufficient data are typically present to specify the relevant dependencies accurately.

2 Introduction

Calls for competition in the power industry, from the wholesale level to the retail level, have made deregulation an attractive option around the world. New market structures have been studied to search for a good one that can ultimately satisfy regulatory bodies, customers, and suppliers. One approach that has been tried is the brokerage system. To accomplish it, the vertically integrated utilities are converted into a horizontal structure. The framework of the energy market is shown in Sheblé [11]. Since the emphasis of this paper is on the value at risk (VaR) of serving customer demand, the energy service company (ESCO), which serves customers, is discussed while leaving the rest to reference [11].

The ESCO collects its revenue from the customers of the energy and ancillary services it provides. It can also act as a wholesaler, reselling electric energy to other ESCOs, generating companies, etc. To obtain the desired electric energy to serve its purposes, the ESCO may purchase it through the auction market, or utilize the reserves that it has accumulated through load management programs or ownership of generation units.

In the deregulated environment, customers are free to choose among ESCOs. In addition, energy purchased by ESCOs from the auction market bears the risk of market price fluctuation. These, from the demand factors to the supply factors, are risks that the ESCO has to take in the new market structure. Since deregulation will render governmental financial protection largely obsolete, risk management and assessment tools should be considered and applied.

Ng and Sheblé (2000 [10]) introduce the different risk management and assessment tools available to assist an ESCO. This paper emphasizes Value at Risk (VaR) analysis.

3 VaR Analysis Review

VaR is the maximum amount of money that may be lost on a portfolio over a given period of time, with a given level of confidence (Best 1998 [2]). Figure

1 shows the graphical representation of VaR. VaR calculations are important because exceeding an appropriately defined maximum loss would be a major or even irrecoverable blow to the company. Thus business decisions need to be made with the objective of keeping the probability of such a loss below a relatively low level of probability deemed acceptable. Consequently determining the probability of such a catastrophic loss should be done carefully and, for dependability, should be validated with respect to lack of knowledge about the dependencies among the variables factoring into the calculation.

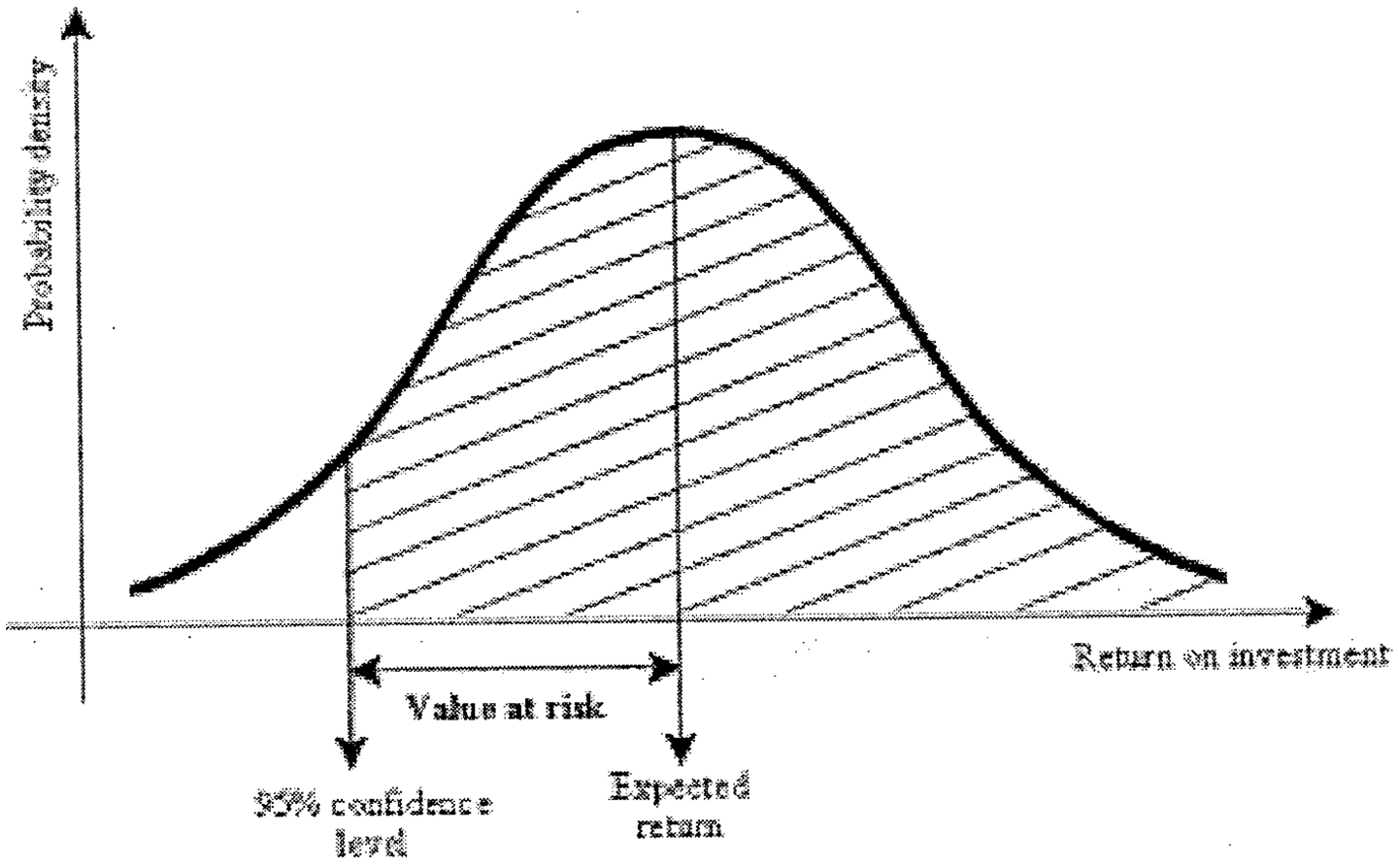


Figure 1: VaR at a given example confidence level.

There are currently three techniques that can be used to evaluate VaR of an ESCO. The first technique is historical simulation, which applies historical data to evaluate the VaR. The second technique is the covariance technique. To apply the covariance technique, the correlation matrix, C, of the uncertain factors is assumed available. The third technique is Monte Carlo simulation. Monte Carlo simulation involves artificially generating a very large set of events, from which VaR is derived [2].

The covariance technique is the easiest and fastest technique among the three. However, the technique assumes that the uncertain factors are normally distributed. Since normal distributions do not necessarily apply to all situations, the technique is consequently limited. Historical simulation and Monte Carlo simulation can supplement the covariance technique in such cases. Since historical simulation uses historical data to evaluate the VaR, there is no need to assume the form of the probabilistic distribution function of the uncertain factors. However, when historical data is limited, solving the VaR using the

historical simulation method can be problematic. The Monte Carlo simulation method requires assuming the probability distributions of the uncertain factors (often that they are normal, but uncertain factors that are not normally distributed can be handled). For instance, in determining the VaR of holding option contracts (whose prices are not normally distributed), the option sensitivities (normally distributed) are used for the Monte Carlo simulation. Thus, the resulting VaR is able to consider option contracts [2].

Best (1998 [2]) describes the VaR resulting from asset price changes, the diversity of the portfolio (the number of assets with correlated price changes), and the holding position of the portfolio (the amount of money invested in a particular asset). This evaluation process is sufficient in a financial institution where the risk is primarily a result of price changes. To an ESCO, however, evaluating the VaR of the price changes is not sufficient. In addition to the risk of price fluctuation, there are two additional risks not described by Best. First, the customer demand and the deliverability of energy are uncertain, as there is a risk associated with the ESCO not being able to serve the customer with sufficient energy. For example, energy delivery can be prevented by transmission system failure, generation failure, etc. Thus an ESCO suffers the risk of contract violation by its supplier. Figure 2 shows the three components of VaR affecting a particular decision (such as amount of load management energy, number of contracts, purchased ancillary services, etc.) for an ESCO.

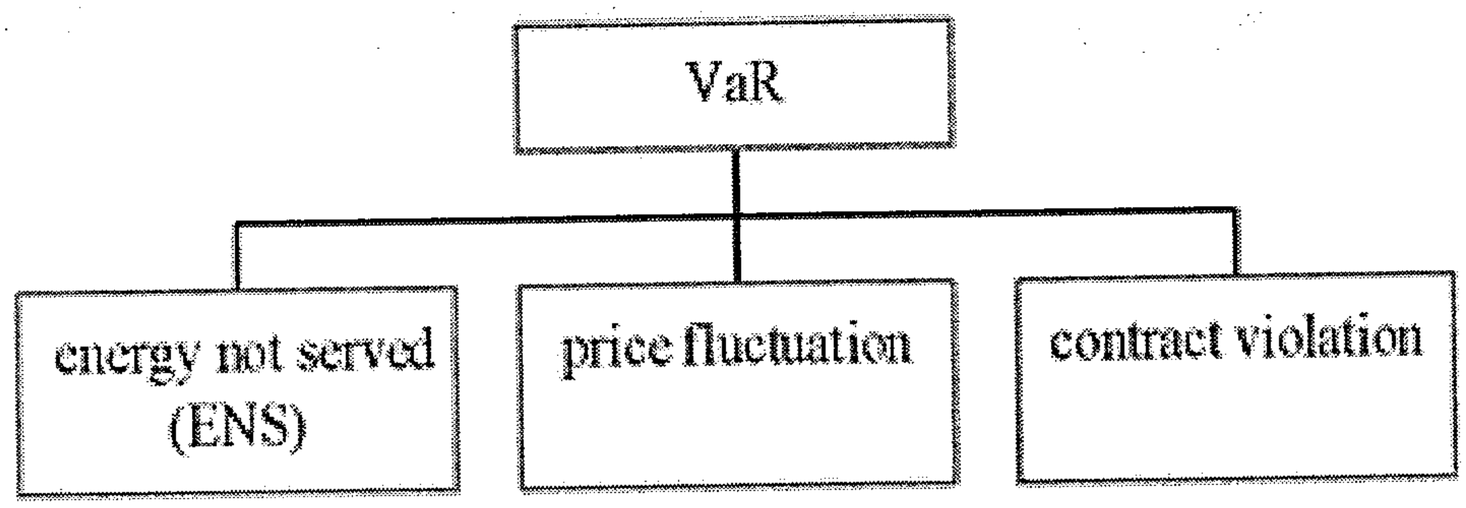


Figure 2: Factors in determining VaR of an ESCO.

4 Price Fluctuation VaR

To evaluate the VaR of market price fluctuation, the covariance matrix of the market price fluctuation is assumed to be available. Historical data may be used in determining the covariance matrix. Then, the VaR of market price fluctuation is evaluated using (1).

$$VaR = \lambda\sqrt{\boldsymbol{PCP}^t} \tag{1}$$

$\boldsymbol{P}$ is the proportion or position of the assets in monetary value. λ represents the degree of volatility and determines the confidence level. For instance, when $\lambda = 1$, the confidence level is 95% [2]. The covariance matrix, C, is determined using (2). Notation E in (2) denotes "expected value of":

$$C = \begin{vmatrix} C_{11} & \dots & C_{1n} \\ \dots & C_{ij} & \dots \\ C_{n1} & \dots & C_{nn} \end{vmatrix}, \tag{2}$$

where

$$C_{ii} = \frac{E[(P_i - EP_i)(P_i - EP_i)]}{\sigma^2_{cP_i}}; \quad C_{ij} = \frac{E[(P_i - EP_i)(P_j - EP_j)]}{\sigma_{P_i}\sigma_{P_j}} \text{ for } i \neq j.$$

Reference 11 shows the steps in evaluating VaR due to market price fluctuation.

In our presentation and full paper we will describe the remaining two major components of the ESCO value at risk, which are energy not served and contract violation. Background information on the the electric power industry will also be provided, emphasizing the ESCO as needed to support the discussion. Combining the three VaR components in a validated way that bounds uncertainty in results due to unspecified dependencies among the distributions describing the values of the three components, will be accomplished through Distribution Envelope Determination (DEnv) [1]. This is needed because the dependency relationships among the probability distributions for the three VaR components are not well understood. The output of DEnv will be used to bound the value at risk given a desired confidence level, or to bound the confidence level given a value at risk.

References

[1] D. Berleant and C. Goodman-Strauss, "Bounding the results of arithmetic operations on random variables of unknown dependency using intervals", *Reliable Computing*, 1998, Vol. 4, No. 2, pp. 147–165.

[2] P. Best, *Implementing Value at Risk*, Chichester, New York, 1998.

[3] S. Dekrajangpetch, *Auction development for the price-based electric power industry*, Ph.D. dissertation, Iowa State University, Ames, December 1999.

[4] J. C. Hull, *Options, Futures, and Other Derivatives*, Prentice Hall, Upper Saddle River, New Jersey, 1997.

[5] J. F. Marshall, *Futures and Option Contracting: Theory and Practice*, South-Western Publishing Co., Cincinnati, Illinois, 1989.

[6] K. H. Ng, *Reformulating Load Management under Deregulation*, Master's thesis, Iowa State University, Ames, May 1997.

[7] K. H. Ng, *Operational Planning for Energy Service Company*, Ph.D. Preliminary Report, Iowa State University, Ames, July 1999.

[8] K. H. Ng and G. B. Sheblé, "Direct Load Control - A Profit-Based Load Management using Linear Programming", *IEEE Trans. Power Systems*, 1998, Vol. 13, No. 2.

[9] K. H. Ng and G. B. Sheblé, *The Economics and Modeling of Scheduling Load Management*, 26th Annual Report, Electric Power Research Center/Power Affiliate Research Program, Iowa State University, Ames, April 1999.

[10] K. H. Ng and G. B. Sheblé, "Risk Management Tools for an ESCO Operation", *Proc. 6th International Conference on Probabilistic Methods Applied to Power Systems*, September 2000.

[11] G. B. Sheblé, *Computational Auction Mechanisms for Restructured Power Industry Operation*, Kluwer Academic Publishers, Boston, 1999.

[12] G. L. Thompson and S. Thore, *Computational Economics: Economic Modeling with Optimization Software*, The Scientific Press, San Francisco, California, 1992.

[13] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, John Wiley & Sons, New York, 1996.

[14] www.nymex.com.

Reliable Modeling Using Interval Analysis: Chemical Engineering Applications

Mark A. Stadtherr
Department of Chemical Engineering
University of Notre Dame
182 Fitzpatrick Hall
Notre Dame, IN 46556 USA

Chemical engineers often deal with nonlinear models of complex physical phenomena, on scales ranging from the macroscopic to the molecular. Frequently these nonlinear models occur in process engineering problems requiring nonlinear equation solving and/or optimization. The reliability with which these problems can be solved is often an important issue. For example, in process optimization a consistent issue concerning reliability is whether or not a global, as opposed to local, optimum has been achieved. In process modeling, especially with the use of highly nonlinear models, the issue of whether a solution is unique is of concern, and if no solution is found, of whether there actually exists a solution to the posed problem. For some problems, the model may have multiple solutions and all must be found, with no *a priori* knowledge of the number of solutions that exist. Methods based on interval analysis provide the power to solve these problems reliably, in fact with mathematical certainty.

In recent years, interval-Newton-based methods have begun to be used for the reliable solution of chemical engineering problems such as:

(1) phase stability analysis using excess Gibbs energy models (Stadtherr *et al.*, 1995; McKinnon *et al.*, 1996; Tessier *et al.*, 2000) and cubic equation of state models (Hua *et al.*, 1996, 1998),

(2) computation of homogeneous azeotropes (Maier *et al.*, 1998) and reactive azeotropes (Maier *et al.*, 2000),

(3) computation of mixture critical points (Stradi *et al.*, 2001),

(4) computation of solid-fluid equilibrium (Xu *et al.*, 2000) and

(5) parameter estimation in vapor-liquid equilibrium models (Gau *et al.*, 2000).

In each case, the interval methodology is used to deal rigorously with issues of multiple (or no) roots in nonlinear equation solving problems or issues of multiple local extrema in optimization problems.

In this presentation, the focus will be on some new applications of interval methods to solve nonlinear modeling and optimization problems in chemical engineering. Of particular interest are (1) some relatively large dimension problems (largest with 264 variables) arising from parameter estimation using the error-in-variables approach and (2) some molecular-scale problems arising in density functional theory and in statistical associating fluid theory. Improvements in the methodology used to solve these problems will also be discussed.

References

[1] C.-Y. Gau, J. F. Brennecke, and M. A. Stadtherr, "Reliable parameter estimation in VLE modeling", *Fluid Phase Equilib.*, 2000, Vol. 168, p. 1.

[2] J.-Z. Hua, J. F. Brennecke, and M. A. Stadtherr, "Reliable prediction of phase stability using an interval-Newton method", *Fluid Phase Equilib.*, 1996, Vol. 116, p. 52.

[3] J. Z. Hua, J. F. Brennecke, and M. A. Stadtherr, "Enhanced interval analysis for phase stability: cubic equation of state models", *Ind. Eng. Chem. Res.*, 1998, Vol. 37, p. 1519.

[4] R. W. Maier, J. F. Brennecke, and M. A. Stadtherr, "Reliable computation of homogeneous azeotropes", *AIChE J.*, 1988, Vol. 44, p. 1745.

[5] R. W. Maier, J. F. Brennecke, and M. A. Stadtherr, "Reliable computation of reactive azeotropes", *Comput. Chem. Eng.*, 2000, Vol. 24, 1851.

[6] K. I.-M. McKinnon, C. G. Millar, and M. Mongeau, "Global optimization for the chemical and phase equilibrium problem using interval analysis", In: C. A. Floudas and p. M. Pardalos (eds.), *State of the Art in Global Optimization: Computational Methods and Applications*, eds. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[7] M. A. Stadtherr, C. A. Schnepper, and J. F. Brennecke, "Robust phase stability analysis using interval methods", *AIChE Symp. Ser.*, 1995, Vol. 91, No. 304, p. 356.

[8] B. A. Stradi, J. F. Brennecke, J. p. Kohn, and M. A. Stadtherr, "Reliable computation of mixture critical points", *AIChE J*, 2001, Vol. 47, p. 212.

[9] S. R. Tessier, J. F. Brennecke, and M. A. Stadtherr, "Reliable phase stability analysis for excess Gibbs energy models", *Chem. Eng. Sci.*, 2000. Vol. 55, p. 1785.

[10] G. Xu, A. M. Scurto, M. Castier, J. F. Brennecke, and M. A. Stadtherr, "Reliable computation of high pressure solid-fluid equilibrium", *Ind. Eng. Chem. Res.*, 2000, Vol. 39, p. 1624.

Methods of Proving Chaos in Dynamical Systems via Transfer Maps

Ralf Toenjes, Jens Hoefkens, Martin Berz, and Kyoko Makino

The determination of regularity versus chaoticity of motion around fixed points is one of the important questions in the theory of dynamical systems. We define chaoticity as motion in which closeby points in phase space exhibit an exponential growth of separation over a sufficiently large time. The goal of this paper is to determine whether the region around a fixed point of the motion exhibits chaoticity or not, and to study at what values of parameters of the system a transition between the two phenomena develops and the system shows bifurcation. We develop methods that allow a rigorous determination of domains of chaoticity around a fixed point of the motion. Also, in cases where the fixed point is a stable attractor, we will be able to prove this stability. To this end, we begin by determining the flow of the motion and its dependence on parameters as a Taylor model using the verified integrator VI, a so-called transfer map. This Taylor model describes the dependence of final coordinates on initial coordinates and parameters via a Taylor polynomial, and provides a rigorous error of this approximation. We then use methods of verified inversion to determine the fixed point of the motion and its dependence on the parameter. Next, a coordinate shift to this parameter dependent fixed point is performed, and as a result, we have a Taylor model describing the dynamics as a function of parameters that is origin preserving.

In addition to the Taylor model of the flow, we also determine the Taylor model of the Jacobian of the flow as a function of both position and parameter by integrating the respective differential equations with the verified integrator. Then the Eigenvalues of the Jacobian are determined using Taylor model arithmetic, where the ability to suppress the dependency problem of Taylor models proves useful in the linear algebra involved in this step. Finally, the moduli of the Eigenvalues are bounded, and regions are determined in which they are bounded below and above by 1, corresponding to the sought regions of chaoticity and stability, respectively.

The method is illustrated with various examples, including the Henon map and the study of dynamics around Lagrangian points in the Earth-Sun system.

The Lorenz Attractor Exists – An Auto-Validated Proof

Warwick Tucker
Dept. of Math., Malott Hall
Cornell University
Ithaca, NY, 14853-4201, USA
warwick@math.cornell.edu

Abstract

We present an algorithm for computing rigorous solutions to a large class of ordinary differential equations. The main algorithm is based on a partitioning process and the use of interval arithmetic with directed rounding. As an application, we prove that the Lorenz equations support a strange attractor, as conjectured by Edward Lorenz in 1963. This conjecture was recently listed by Steven Smale as one of several challenging problems for the 21st century. We also prove that the attractor is robust, i.e., it persists under small perturbations of the coefficients in the underlying differential equations. Furthermore, the flow of the equations admits a unique SRB measure, whose support coincides with the attractor. The proof is based on a combination of normal form theory and rigorous computations.

1 Background to the Problem

The following non-linear system of differential equations,

$$\begin{aligned} \dot{x}_1 &= -\sigma x_1 + \sigma x_2 \\ \dot{x}_2 &= \varrho x_1 - x_2 - x_1 x_3 \\ \dot{x}_3 &= -\beta x_3 + x_1 x_2, \end{aligned} \tag{1}$$

was introduced in 1963 by Edward Lorenz, see [5]. As a crude model of atmospheric dynamics, these equations led Lorenz to the discovery of sensitive dependence of initial conditions - an essential factor of unpredictability in many systems. Numerical simulations for an open neighbourhood of the classical parameter values $\sigma = 10, \beta = 8/3$ and $\varrho = 28$ suggest that almost all points in phase space tend to a strange attractor - *the Lorenz attractor*.

For $\varrho > 1$, there are three fixed points: the origin and the two "twin points"

$$C^{\pm} = (\pm\sqrt{\beta(\varrho-1)}, \pm\sqrt{\beta(\varrho-1)}, \varrho-1).$$

Numerical experiments indicate the existence a forward invariant open set U containing the origin but bounded away from the fixed points $C^{\pm}$. If we let φ denote the flow of (1), we can form the maximal invariant set

$$\mathcal{A} = \bigcap_{t \geq 0} \varphi(U, t).$$

Due to the flow being dissipative, the attracting set $\mathcal{A}$ must have zero volume. It must also contain the unstable manifold of the origin $W^u(0)$, which seems to spiral around $C^{\pm}$ in a very complicated, non-periodic fashion, see Figure 1(a). In particular, $\mathcal{A}$ contains the origin itself, and therefore the flow on $\mathcal{A}$ can not have a hyperbolic structure. The reason is that fixed points of the vector field generate discontinuities for the return maps, and as a consequence, the hyperbolic splitting is not continuous. Apart from this, the attracting set appears to have a strong hyperbolic structure as described below.

As it was very difficult to extract rigorous information about the attracting set $\mathcal{A}$ from the differential equations themselves, a *geometric model* of the Lorenz flow was introduced by John Guckenheimer in the late sixties, see [2]. This model has been extensively studied, and it is well understood today, see e.g. [3], [14], [12], [8], [9], [10]. Oddly enough, the original equations introduced by Lorenz have remained a puzzle. A few computer-assisted proofs, however, have quite recently been announced, see [1], [4], and [6]. These articles deal with subsets of $\mathcal{A}$ which are not attracting, and therefore only concern a set of trajectories having measure zero. Despite this, it has always been widely believed that the flow of the Lorenz equations has the same qualitative behaviour as its geometric model. We prove that the geometric model does indeed give an accurate description of the dynamics of (1).

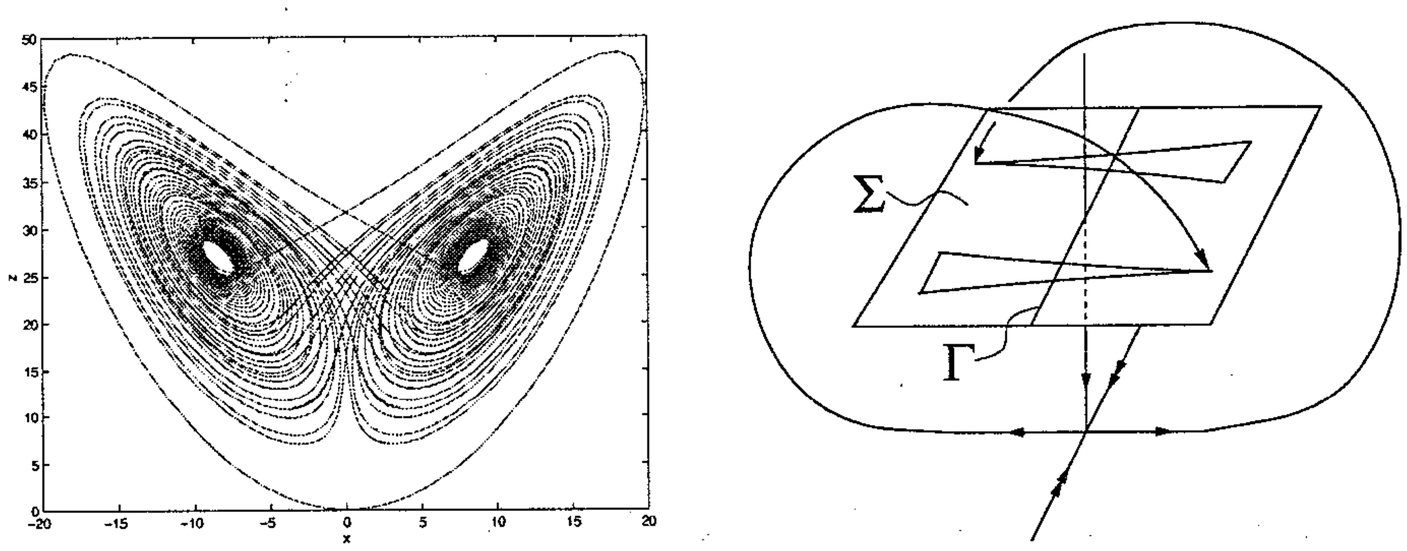


Figure 1: (a) A part of the unstable manifold of the origin. (b) The return map acting on Σ.

By the use of a Poincaré section, the flow of (1) can be reduced to a *first return map* R acting on the section $\Sigma \subset \{x_3 = \varrho - 1\}$, as schematically illustrated in Figure 1(b).

Note that R is not defined on the line $\Gamma = \Sigma \cap W^s(0)$: these points tend to the origin, and never return to Σ. Due to the fixed point at the origin, the return

times are not bounded. This constitutes a serious obstruction to any numerical approach. This is overcome by introducing a local change of coordinates, and we prove the following properties of the return map R:

- There exists a compact set $N \subset \Sigma$ such that $N \setminus \Gamma$ is *forward invariant* under R, i.e., $R(N \setminus \Gamma) \subset \text{int}(N)$. This ensures that the flow has an attracting set $\mathcal{A}$ with a large basin of attraction. We can then form a cross-section of the attracting set: $\mathcal{A} \cap \Sigma = \cap_{n=0}^{\infty} R^n(N) = \Lambda$.

- On N, there exists a cone field $\mathfrak{C}$ which is mapped strictly into itself by DR, i.e., for all $x \in N$, $DR(x) \cdot \mathfrak{C}(x) \subset \mathfrak{C}(R(x))$. The cones of $\mathfrak{C}$ are centered along two curves which approximate Λ, and each cone has an opening of at least $5°$.

- The tangent vectors in $\mathfrak{C}$ are eventually expanded under the action of DR: there exists $C > 0$ and $\lambda > 1$ such that for all $v \in \mathfrak{C}(x)$, $x \in N$, we have $|DR^n(x)v| \geq C\lambda^n|v|$, $n \geq 0$. In fact, the expansion is strong enough to ensure that R is topologically transitive on Λ.

The proof can be broken down into two main sections: one global part, which involves finding enclosures to solutions of ODEs, and one local part, which is based on normal form theory. Both parts require the use of interval arithmetic, as described in [7].

2 The Main Result

In a recent issue of *the Mathematical Intelligencer* the Fields medalist Steven Smale presented a list of challenging problems for the 21th century, see [11]. Problem number 14 reads as follows:

> *Is the dynamics of the ordinary differential equations of Lorenz that of the geometric Lorenz attractor of Williams, Guckenheimer, and Yorke?*

By proving the three abovementioned properties of R, we provide an affirmative answer to Smale's question:

Main Theorem *For the classical parameter values, the Lorenz equations support a robust strange attractor $\mathcal{A}$. Furthermore, the flow admits a unique SRB measure μ_φ with* $\text{supp}(\mu_\varphi) = \mathcal{A}$.

In fact, we prove that the attracting set is a singular hyperbolic attractor. Almost all nearby points separate exponentially fast until they end up on opposite sides of the attractor. This means that a tiny blob of initial values rapidly smears out over the entire attractor, as observed in numerical experiments. The complete proof has been published in [13].

References

[1] Z. Galias and P. Zgliczyński, "Computer assisted proof of chaos in the Lorenz equations", *Physica D*, 1998, Vol. 115, pp. 165–188.

[2] J. Guckenheimer, "A strange, strange attractor", In: Marsden and McCracken (eds.), *The Hopf Bifurcation and its Applications*, Springer-Verlag, New York, 1976.

[3] J. Guckenheimer and R. F. Williams, "Structural Stability of Lorenz Attractors", *Publ. Math. IHES*, 1979, Vol. 50, pp. 307–320.

[4] S. P. Hasting and W. C. Troy, "A Shooting Approach to the Lorenz Equations", *Bull. Amer. Math. Soc.*, 1992, Vol. 27, pp. 298–303.

[5] E. N. Lorenz, "Deterministic Non-periodic Flow", *J. Atmos. Sci.*, 1963, Vol. 20, pp. 130–141.

[6] K. Mischaikow and M. Mrozek, "Chaos in the Lorenz Equations: A Computer-Assisted Proof", *Bull. Amer. Math. Soc.* 1995, Vol. 32, pp. 66–72.

[7] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[8] D. Rand, "The topological classification of Lorenz attractors", *Math. Proc. Camb. Phil. Soc.*, 1978, Vol. 83, pp. 451–460.

[9] C. Robinson, "Homoclinic Bifurcation to a Transitive Attractor of Lorenz Type", *Nonlinearity*, 1989, Vol. 2, pp. 495–518.

[10] M. Rychlik, "Lorenz attractors through a Sil'nikov-type bifurcation. Part 1", *Ergod. Th. & Dynam. Sys.* 1989, Vol. 10, pp. 793–821.

[11] S. Smale, "Mathematical problems for the next century", *Math. Intelligencer*, 1998, Vol. 20, pp. 7–15.

[12] C. Sparrow, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*, Springer-Verlag, New York, 1982.

[13] Tucker, W., "A Rigorous ODE Solver and Smale's 14th Problem", *Foundations of Computational Mathematics*, 2002, Vol. 2, No. 1, pp. 53–117.

[14] R. F. Williams, "The Structure of Lorenz Attractors", *Publ. Math. IHES*, 1979, Vol. 50, pp. 321–347.

Kite: A New Inclusion Function for Optimization

Tamás Vinkó[1], Jean-Louis Lagouanelle[2], and Tibor Csendes[1]

[1]University of Szeged
Institute of Informatics, Szeged, Hungary
emails {tvinko,csendes}@inf.u-szeged.hu

[2]Lab. LIMA
Institut de Recherche en Informatique de Toulouse
France, email lagouane@irit.fr

Interval global optimization algorithms based on branch-and-bound methods provide guaranteed and reliable solutions for the problem

$$\min_{x \in X} f(x),$$

where the objective function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $X \subseteq D$ is the search box representing bound constraints for x. The aim of this work is to improve the efficiency by a tighter interval inclusion function, in particular we deal with *lower bounds* of f, because the guaranteed upper bound of the global minimum values can be obtained by a single function evaluation. The quality of an enclosure method is important in the implementation of the interval global optimization algorithms, because narrower enclosure of f may provide faster convergence.

In this paper the kite inclusion function is presented for branch-and-bound type interval global optimization using at least gradient information. In the one dimensional case [3] the basic idea comes from the simultaneous usage of the centered forms and the linear boundary value forms [2]. Figure 1 shows that the graph of f is within the convex inclusion cone determined by the points $(a, f(a)), S$ and $(b, f(b))$ and outside the concave exclusion cone MPN. In this figure the lines L and U mean the lower and the upper bound of the derivative of $f(x)$, while the lower bounds for the function $f(x)$ given by the centered form, the linear boundary value form and the kite are $\underline{F}_{CF}$, $\underline{F}_{LBVF}$ and $\min\{y_R, y_T\}$, respectively.

This leads to the assertion that the simultaneous usage provides a not worse enclosure of the objective function. The best choice for the center of the kite (the point (c, P) in Figure 1) correspond to the case $y_R = y_T$. The existence and uniqueness of this case can be shown in two ways: in a geometrical and in a constructive way. The isotonicity and at least quadratical convergence hold

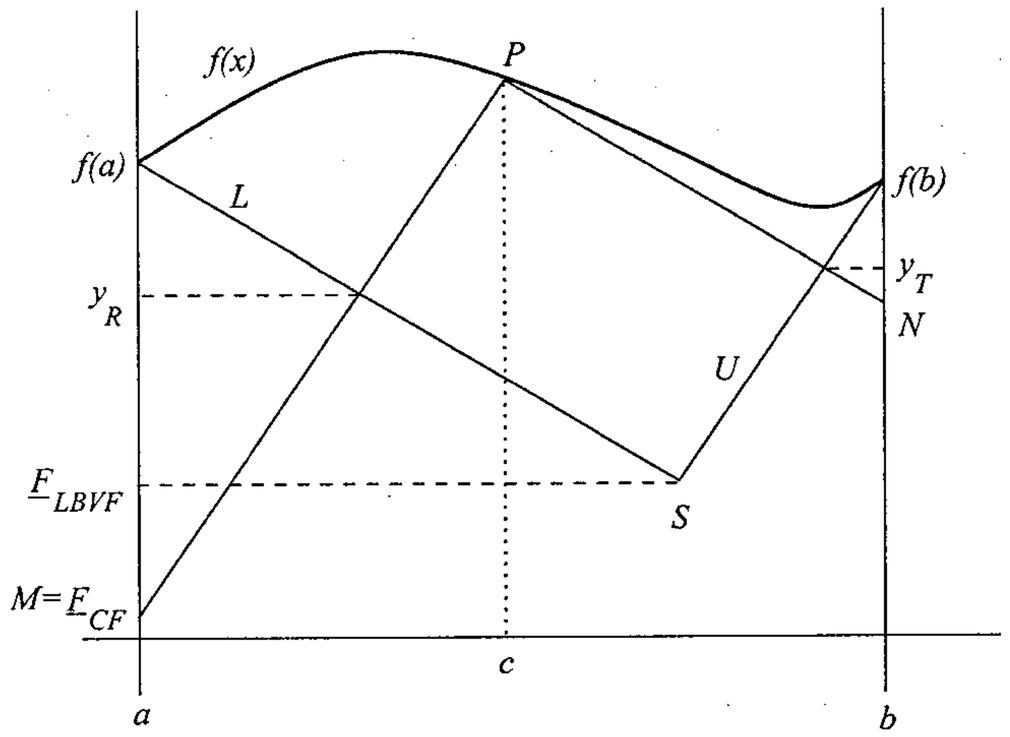


Figure 1: Simultaneous usage of the centered form (based on the middle point of the current interval) and the linear boundary value form.

and there is a pruning effect of the kite which is derived from the construction of the inclusion, thus more function evaluation is not needed to use it.

The new method can easily be implemented in a branch-and-bound type interval global optimization algorithm. For a single inclusion larger computation effort is needed by the kite algorithm, because we use the function values not only at the optimal center but at the extremal points of the examined interval.

Numerical investigations on 40 standard multiextremal test functions have been done to show the performance. For the one dimensional problems our results are summarized in Table 1, where the performance of the kite with and without its pruning effect (k+pr and k−pr, respectively) is compared to the centered form. The columns contain the number of function, derivative, and Hessian evaluation, number of bisection and the necessary list length. The percent values give the average values for the complete test set for the first and the second order algorithms.

F-eval.		D-eval.		H-eval.		bisection		list length	
k+pr.	k−pr	k+pr	k−pr	k+pr	k−pr	k+pr	k−pr	k+pr	k−pr
66%	74%	39%	57%	–	–	60%	87%	56%	63%
79%	84%	64%	69%	111%	120%	76%	83%	78%	78%

Table 1: Numerical results.

In the multi dimensional case the kite inclusion function can be based on the centered forms and the method of the supporting hyperplanes [1]. Another idea is the application of the componentwise approach, where the multidimensional

case can be lead back to the one dimensional case. These ideas should be investigated in the future.

Acknowledgments. This work have been supported by the grants OTKA T 034350 and OTKA T 32118.

References

[1] F. Messine and J.-L. Lagouanelle, "Enclosure Methods for Multivariate Differentiable Functions and Application to Global Optimization", *Journal of Universal Computer Science*, 1998, Vol. 4, No. 6, pp. 589–603.

[2] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.

[3] T. Vinkó, J.-L. Lagouanelle, and T. Csendes, *A New Inclusion Function for Optimization: Kite – The One Dimensional Case*, 2001; Submitted for publication; Available at: `http://www.inf.u-szeged.hu/~tvinko/kite.ps.gz`

Moore's Single-Use-Expression Theorem on Extended Real Intervals*

G. William Walster
Sun Microsystems

Moore [1] proved that conditions exist when a computed interval's value is the expression's exact range. The conditions are: the expression is valid (no division by zero), rational, and real (not extended real); and each interval variable occurs no more than once in the expression. Unfortunately, in the set of extended real numbers, denoted $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$, Moore's single-use-expression theorem is not always true. Division asymmetry in the $\mathbb{R}^*$ number system is the root cause of the problem.

This paper further extends the $\mathbb{R}^*$ number system to remove the asymmetry from extended real division so that Moore's single-use-expression result holds. The new system is denoted $\mathbb{R}^{**}$. The new system is also applied to the complex plane to show that closed complex interval systems can be based on sets in the $\mathbb{R}^{**} \otimes \mathbb{I}^{**}$ system. Interval implementations are easily developed within the IEEE 754 floating point standard.

References

[1] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1966.

Intervals: The "Glue" Between the Java™ Language, Computing Hardware and Grids*

G. William Walster
Sun Microsystems

Intervals can become the "glue" that connects the write-once-run-anywhere Java™ language paradigm to computing hardware and grids. The result will be Java™-language advantages *and* unsurpassed runtime performance. This paper describes how the properties of interval solvers will be used to logically connect Java™ applications to generic interval solver libraries that efficiently perform the most demanding floating-point computations.

Enclosures of Higher Order Derivative Tensors on the Basis of Univariate Taylor Expansions

Andrea Walther and Jan Riehme
Institute of Scientific Computing
Technical University Dresden, Germany

This contribution considers the problem of evaluating all pure and mixed partial derivatives of some vector function

$$\boldsymbol{y} \;=\; \boldsymbol{f}(\boldsymbol{x}) \quad \text{with } \boldsymbol{f} : \mathcal{D} \subset \mathbb{R}^{\bar{n}} \mapsto \mathbb{R}^m,$$

defined by a computer program. In order to provide the required derivative information Automatic Differentiation (AD) can be applied.

Even though the reverse mode of AD may be more efficient when the number of dependent variables is small compared to the number of independents, only the forward mode will be considered here. The mechanics of this direct application of the chain rule are completely independent of the number m of dependent variables so that it is possible to restrict the analysis to a scalar-valued function

$$y \;=\; f(\boldsymbol{x}) \quad \text{with} \quad f : \mathcal{D} \subset \mathbb{R}^{\bar{n}} \mapsto \mathbb{R}.$$

This greatly simplifies the notation, and the full tensors can then easily be obtained by an outer loop over the component index.

The natural approach to evaluate derivative tensors seems to be their recursive calculation using the usual forward mode of AD. This technique has been implemented by Berz [1], Neidinger [3], and others. The only complication using this multi-variant approach is the need to utilize the symmetry in the higher derivative tensors, which leads to fairly complex addressing schemes.

Much simpler data access patterns and similar or lower computational counts can be achieved through propagating a family of univariate Taylor series of an arbitrary degree. At the end, their values are used to compute the desired tensor coefficients [2].

Using exact arithmetic, both approaches yield the same derivative information. Obviously, the situation changes if the computations are performed in floating point or interval arithmetic. We analyze the effects of using interval arithmetic for both methods to evaluate derivatives. Furthermore the quality of the enclosures achieved are compared and discussed.

References

[1] M. Berz, "Algorithms for higher derivatives in many variables with applications to bear physics", In G. F. Corliss and A. Griewank (eds.), *Automatic Differentiation of Algorithms – Theory, Implementation, and Applications*, SIAM, 1991, pp. 147–156.

[2] A. Griewank, J. Utke, and A. Walther, "Evaluating Higher Derivative Tensors by Forward Propagation of Univariate Taylor Series", *Computation of Mathematics*, 2000, Vol. 69, pp. 1117–1130.

[3] R. Neidinger, "An efficient method for the numerical evaluation of partial derivatives of arbitrary order", *ACM Trans. Math. Software*, 1992, Vol. 18, pp. 159–173.

Author Index